DressCode: Autoregressively Sewing and Generating Garments from Text Guidance

KAI HE, ShanghaiTech University and Deemos Technology Co., Ltd., China
KAIXIN YAO, ShanghaiTech University and NeuDim Technology Co., Ltd., China
QIXUAN ZHANG, ShanghaiTech University and Deemos Technology Co., Ltd., China
JINGYI YU*, ShanghaiTech University, China
LINGJIE LIU*, University of Pennsylvania, USA
LAN XU*, ShanghaiTech University, China

Fig. 1. Our *DressCode* generates CG-friendly customized garments with sewing patterns and PBR textures under natural text guidance, enabling post-editing, animation, and high-quality rendering.

Apparel's significant role in human appearance underscores the importance of garment digitalization for digital human creation. Recent advances in 3D content creation are pivotal for digital human creation. Nonetheless, garment generation from text guidance is still nascent. We introduce a text-driven 3D garment generation framework, *DressCode*, which aims to democratize design for novices and offer immense potential in fashion design, virtual try-on, and digital human creation. We first introduce SewingGPT, a GPT-based architecture integrating cross-attention with text-conditioned embedding to generate sewing patterns with text guidance. We then tailor a pre-trained Stable Diffusion to generate tile-based Physically-based Rendering (PBR) textures for the garments. By leveraging a large language model, our framework generates CG-friendly garments through natural language interaction. It also facilitates pattern completion and texture editing, streamlining the design process through user-friendly interaction. This framework fosters innovation by allowing creators to freely experiment with designs and incorporate unique elements into their work. With comprehensive evaluations and comparisons with other state-of-the-art methods, our method showcases superior quality and alignment with input prompts. User studies further validate our high-quality rendering results, highlighting its practical utility and potential in production settings. Our project page is https://IHe-KaiI.github.io/DressCode/.

CCS Concepts: • **Computing methodologies** → **Computer graphics**.

Additional Key Words and Phrases: Garment Generation, Sewing Patterns, Autoregressive Model

*Corresponding author.

Authors' addresses: Kai He, hekai@shanghaitech.edu.cn, ShanghaiTech University and Deemos Technology Co., Ltd., Shanghai, China; Kaixin Yao, yaokx2023@shanghaitech.edu.cn, ShanghaiTech University and NeuDim Technology Co., Ltd., Shanghai, China; Qixuan Zhang, zhangqx1@shanghaitech.edu.cn, ShanghaiTech University and Deemos Technology Co., Ltd., Shanghai, China; Jingyi Yu, yujingyi@shanghaitech.edu.cn, ShanghaiTech University, Shanghai, China; Lingjie Liu, lingjie.liu@seas.upenn.edu, University of Pennsylvania, Philadelphia, USA; Lan Xu, xulan1@shanghaitech.edu.cn, ShanghaiTech University, Shanghai, China.

ACM 0730-0301/2024/7-ART72
https://doi.org/10.1145/3658147

ACM Reference Format:
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, Article 72 (July 2024), 13 pages. https://doi.org/10.1145/3658147

1 INTRODUCTION

Apparel substantially influences the appearance of us humans, and hence garment digitalization has emerged as a vital component of digital human creation. An effective digital garment creation tool should enable users to customize garments to depict the individuality and diversity that make up our physical world, with various garment traits like sewing patterns, styles, or materials. The creation process also needs to match specific themes and be convenient, as simple as chatting with AI agents like ChatGPT.

Recent years have witnessed tremendous progress in text-driven asset generation, triggered by the large-scale language models [Achiam et al. 2023; Radford et al. 2021]. It democratizes the accessible text-driven creation of diverse assets for novices, including images [Rombach et al. 2022], generic 3D objects [Liu et al. 2023b; Poole et al. 2022], human hair [Zhou et al. 2023], face or body [Liao et al. 2023; Zhang et al. 2023a]. What is still missing is the garment. Moreover, naively applying avatar or general generation [Liao et al. 2023; Poole et al. 2022] for the garment category is suboptimal since they turn to generate mesh or neural fields that are incompatible with digital garment production workflow.

In contrast, for our graphics community, the dominant representation of garments is sewing patterns, which facilitates both physical simulation and animation in a CG-friendly fashion [Autodesk, INC. 2019; Blender Foundation 2022]. For sewing pattern generation, early methods [Berthouzoz et al. 2013; Umetani et al. 2011] only use simple partial modules in the workflow like parsing or draping to 3D. With the consolidation of more advanced datasets [Korosteleva and Lee 2021], recent work enables sewing pattern generation from point clouds [Korosteleva and Lee 2022] or images [Liu et al. 2023d]. However, they largely overlook the generation through more natural language interactions, let alone handling the vivid generation with desired texture patterns or physically-based materials, which could significantly speed up the preliminary stage of garment design. Furthermore, through the natural interactions of text prompts, novices without professional skills in complex design software can directly describe and transform their ideas into creations. This significantly lowers the design barriers, allowing more newcomers to participate in the creative process. Most importantly, generative models introduce diversity to designed text prompts, generate varied types of garments conforming to the prompts, and hence stimulate designer creativity.

In this paper, we propose *DressCode*, a 3D garment generation framework that generates high-quality garments via natural language interaction. As illustrated in Figure 1, DressCode allows users to customize garments with preferred sewing patterns and physically-based texture details through text interaction. The resulting garments can be seamlessly integrated with CG pipelines, supporting post-editing and animation while ensuring high-quality rendering.

Notably, the garments' highly symmetric and structured nature, with uniform panels and stitching, leads to convenient conversion from sewing patterns to discrete "codes." To this end, inspired by powerful language generation apt for this nature, we introduce SewingGPT, a GPT-based architecture for sewing pattern generation. Specifically, we adopt a novel quantization process to translate the sewing patterns into token sequences and subsequently utilize a decoder-only Transformer with text-conditioned embedding for token prediction. We utilize the pre-trained CLIP [Radford et al. 2021] model to encode prompts as conditional embeddings, benefiting from CLIP's generalized capability in multimodal understanding. For effective training, we apply GPT-4V [Achiam et al. 2023] on the existing dataset [Korosteleva and Lee 2021] to detail diverse garment types and shapes with rich text prompts. Once trained, our SewingGPT autoregressively generates quantized sewing patterns with efficient text interactions, which have been unseen before.

To achieve high-quality garment rendering, we progressively tailor a pre-trained Stable Diffusion model [Rombach et al. 2022] to generate tile-based Physically-based Rendering (PBR) textures from text prompts. We first fine-tune the U-Net architecture [Ronneberger et al. 2015] of the diffusion model within latent space to generate the diffuse attribute, and then fine-tune the various VAE [Kingma and Welling 2013] decoders to generate normal and roughness maps separately. We showcase the capability of DressCode to generate CG-friendly garments with rich sewing patterns and PBR textures from text prompts. We also demonstrate the versatility of our approach, including a ChatGPT-like conversational agent for interactive garment generation, garment completion from partial inputs, and user-friendly texture editing. To summarize, our main contributions include:

- We propose a first text-driven garment generation pipeline with high-quality garment sewing patterns and physically-based textures.
- We introduce a novel generative paradigm for sewing patterns as a sequence of tokens, achieving high-quality autoregressive generation via text guidance.
- We tailor a diffusion model for vivid texture generation of garments from text prompts and showcase interaction-friendly applications for garment generation, completion, and editing.

2 RELATED WORK

Garment Sewing Pattern Modeling. Sewing pattern representation is vital for garment modeling. Recent studies have delved into sewing pattern reconstruction [Jeong et al. 2015; Pietroni et al. 2022; Sharp and Crane 2018; Su et al. 2022; Wang et al. 2018; Yang et al. 2018], generation [Shen et al. 2020], draping [Berthouzoz et al. 2013; De Luigi et al. 2023; Li et al. 2023b] and editing [Bartle et al. 2016; Qi et al. 2023; Umetani et al. 2011]. Early research [Chen et al. 2015] employs a search in a pre-defined database of 3D garment parts for garment reconstruction. Studies [Jeong et al. 2015; Su et al. 2020; Yang et al. 2018] use parametric sewing patterns, optimizing them for garment reconstruction from images. [Wang et al. 2018] advances this by applying deep learning to discern a shape space for sewing patterns and various input modalities. [Bang et al. 2021;

Sharp and Crane 2018] utilize surface flattening for sewing pattern reconstruction from 3D human models.

Recently, some work [Goto and Umetani 2021; Korosteleva and Lee 2022; Zhu et al. 2020] adopt data-driven approaches for reconstruction. [Goto and Umetani 2021] utilizes a deep network with surface flattening for sewing pattern reconstruction from 3D geometries. [Korosteleva and Lee 2021] generates a sewing pattern dataset covering a wide range of garment shapes and topologies. NeuralTailor [Korosteleva and Lee 2022] offers advanced sewing pattern reconstruction using a hybrid network to predict garment panels and stitching information from point cloud input. [Chen et al. 2022] introduces a CNN-based model capable of predicting garment panels from single images, using PCA to simplify the panel data structure. [Liu et al. 2023d] creates a comprehensive dataset featuring diverse garment styles and human poses and introduces a two-level Transformer network, achieving state-of-the-art sewing pattern reconstruction from single images. [Korosteleva and Sorkine-Hornung 2023] designs the first DSL for garment modeling, enabling users to do rich garment designs using interchangeable, parameterized components. [Li et al. 2023a] proposes a novel approach to recover garment materials and patterns with optimization using differentiable simulation. While these solutions yield notable results, a gap persists in user-friendliness and practicality compared to direct communication of outcomes through natural language. Furthermore, prior studies have largely overlooked generating garment color, texture, and material, essential elements for creating high-quality garments.

Text-to-3D Generation. Recent breakthroughs in the text-to-image domain [Ho et al. 2020; Rombach et al. 2022; Zhang et al. 2023b] have enhanced interest in text-guided 3D content generation. Early work [Jain et al. 2022] introduces a text-to-3D method guided by CLIP [Radford et al. 2021]. [Poole et al. 2022; Wang et al. 2023a] present the Score Distillation Sampling (SDS) algorithm, elevating pre-trained 2D diffusion models for the 3D generation. [Metzer et al. 2023] optimizes Neural Radiance Fields (NeRF) [Mildenhall et al. 2021] in the latent space. [Chen et al. 2023a; Lin et al. 2023] optimize efficient mesh representations [Munkberg et al. 2022; Shen et al. 2021] for higher quality generation. [Seo et al. 2023] integrates 3D awareness into 2D diffusion for improving text-to-3D generation consistency. Subsequent studies [Chen et al. 2023b; Lugmayr et al. 2022; Richardson et al. 2023] focus on texturing pre-existing meshes, balancing speed and quality. Despite their innovations, SDS-based methods faced challenges with over-saturation. [Tsalicoglou et al. 2023] proposes a novel method to refine mesh textures for more realistic generations. ProlificDreamer [Wang et al. 2023b] introduces the Variational Score Distillation (VSD) method to mitigate over-saturation effectively. Furthermore, several studies [Liu et al. 2023a; Shi et al. 2023b; Ye et al. 2023; Zhao et al. 2023] explored 3D generation using multi-view diffusion. Concurrently, some research [Huang et al. 2023; Liu et al. 2023c,b; Long et al. 2023; Melas-Kyriazi et al. 2023; Qian et al. 2023; Raj et al. 2023; Shi et al. 2023a; Tang et al. 2023; Wu et al. 2023; Xu et al. 2023] concentrates on reconstructing 3D content from a single image through distillation, achieving high-fidelity textured meshes from 2D diffusion priors. Additionally, some studies [Erkoç et al. 2023; Nash et al. 2020; Siddiqui et al. 2023; Yu et al. 2023] delve into shape generation. [Yu et al. 2023] generates high-quality 3D shapes with Unsigned Distance Field (UDF) through Diffusion models. [Nash et al. 2020; Siddiqui et al. 2023] employ autoregressive models for mesh structure generation. Although some general object generation methods [Mildenhall et al. 2021; Poole et al. 2022; Qiu et al. 2023; Wang et al. 2023a,b; Yu et al. 2023] can produce garments, their practicality in CG environments is limited. Very Recent work, Garment3DGen [Sarafianos et al. 2024] enables users to generate textured 3D garments from single images or text prompts based on a 3D base mesh. These 3D outputs, mostly mesh-based or derived from implicit fields, lack adaptability for fitting different bodies and layering multiple garments, common needs in garment design. Furthermore, the textures, typically produced via optimization or multi-view reconstruction, are often low-resolution and blurry, neglecting the structured UV mapping of garments, resulting in poor topology challenging for subsequent CG processing.

3 SEWING PATTERN GENERATION

Inspired by powerful language generative models, we introduce SewingGPT, a GPT-based autoregressive model for sewing pattern generation with text prompts. We first convert sewing pattern parameters into a sequence of quantized tokens and train a masked Transformer decoder, integrating cross-attention with text-conditioned embeddings. After training, our model can generate token sequences autoregressively based on user conditions. The generated sequences are then de-quantized to reconstruct the sewing patterns.

3.1 Sewing Pattern Quantization

Pattern representation. We utilize the sewing pattern templates from [Korosteleva and Lee 2022], which cover a wide variety of garment shapes. Each sewing pattern includes N_P panels $\{P_i\}_{i=1}^{N_P}$ and stitching information S. Each panel P_i forms a closed 2D polygon with N_i edges $\{E_{i,j}\}_{j=1}^{N_i}$. Each edge $E_{i,j}$ consists of four parameters (v_x, v_y, c_x, c_y), where (v_x, v_y) represents the edge's start point, and (c_x, c_y) represents the control point of the Bezier curve. Since the panels form closed polygons, we do not need to store the edges' endpoints. The 3D placement of each panel is indicated by rotation quaternion $R_i \in \mathrm{SO}(3)$ and translation vector $T_i \in \mathbb{R}^3$. For stitching information, we utilize per-edge stitch tags $\{S_{i,j}\}_{j=1}^{N_i}$ and stitch flags $\{U_{i,j}\}_{j=1}^{N_i}$ for each panel P_i, obtained from the stitching information S. Each stitch tag $S_{i,j} \in \mathbb{R}^3$ is based on the 3D placement of the corresponding edge, and each stitch flag is a binary flag with $U_{i,j} = \{0, 1\}$ indicating whether there is a stitch on this edge. We follow a similar approach to that used in [Korosteleva and Lee 2022], which utilizes Euclidean distance between tags as a similarity measure. To restore the stitching information from stitch tags and stitch flags, we filter out free and connected edges with stitch flags and then compare the stitch tags of all pairs of connected edges.

Quantization. For each panel, we first utilize a similar data preprocessing approach to that used in [Korosteleva and Lee 2022], which standardizes all edge vectors and control points to maintain the data within a standard normal distribution, and normalizes its 3D placement to ensure all values are between 0 and 1. Then, we quantize all

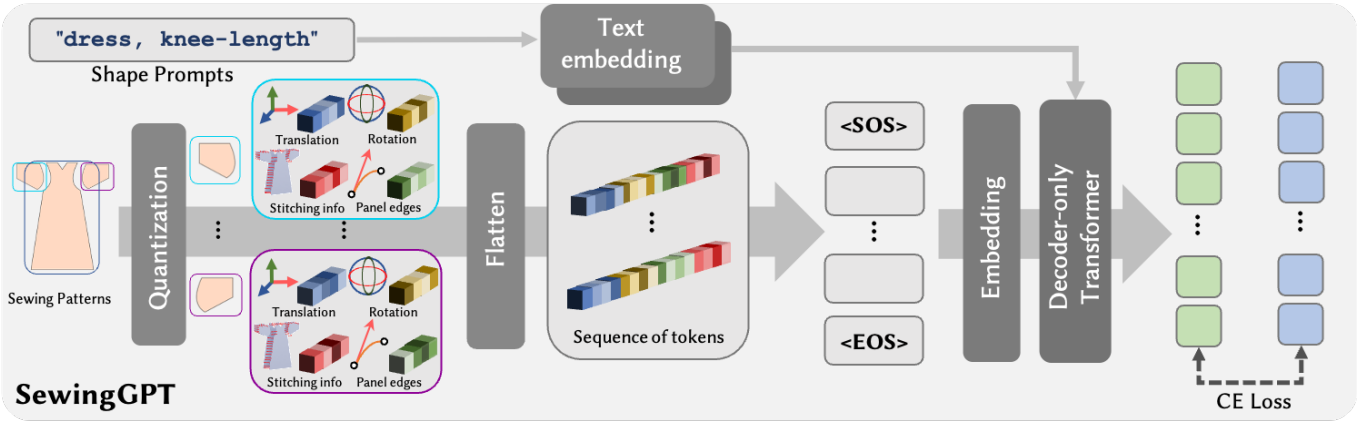

Fig. 2. **Overview of our SewingGPT pipeline.** We quantize sewing patterns to the sequence of tokens and adopt a GPT-based architecture to generate the tokens autoregressively. Our SewingGPT enables users to generate highly diverse and high-quality sewing patterns under text prompt guidance.

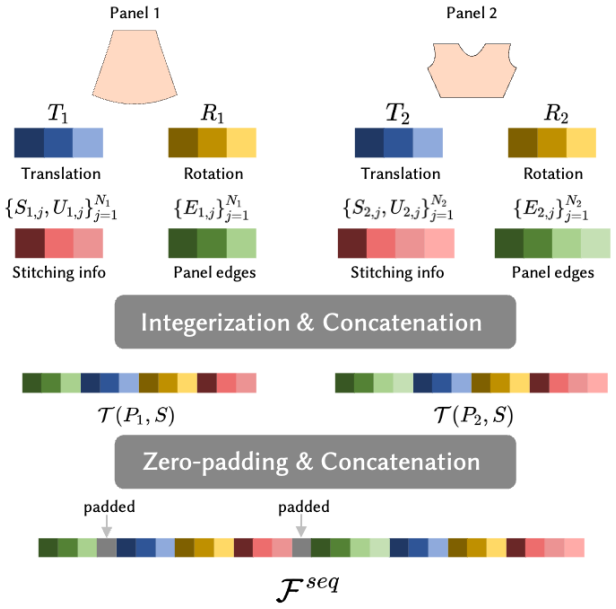

Fig. 3. **Details of our quantization.** We present an example of a part of a sleeveless dress, including a *skirt panel* (Panel 1) and a *top panel* (Panel 2). Assuming $N_1 < N_2 = K$, we require zero-padding for tokens from Panel 1.

parameters, subsequently converting them into tokens. Specifically, for panel P_i, we model all parameters as discrete variables by multiplying predefined constants C_E, C_R, C_T, C_S by edge vectors, rotation, translation, and stitching feature vectors respectively, and maintain stitching flags as 0 or 1. We flatten and concatenate N_i edges, one rotation quaternion, one translation vector, N_i stitching vectors, and N_i stitching flags, into a sequence of tokens. We carefully select these constants to offer a good trade-off between maintaining the fidelity of sewing patterns and managing the vocabulary size.

Overall, we can represent the quantization process as

$$\mathcal{T}(P_i, S) = C_E\{E_{i,j}\}_{j=1}^{N_i} \oplus C_R R_i \oplus C_T T_i \oplus C_S\{S_{i,j}\}_{j=1}^{N_i} \oplus \{U_{i,j}\}_{j=1}^{N_i} \quad (1)$$

where we denote $\mathcal{T}$ as the quantization function, and $\oplus$ as the linear concatenation of tokens. These tokens are then formed into a linear sequence. We set a maximum limit, denoted as K, for the number of edges in each panel. To maintain a uniform token count across panels, we apply zero-padding to panels with $N_i < K$. Subsequently, all panels are flattened and merged into a single sequence, starting with a start token and ending with an end token. Owing to the uniformity of token counts for each panel, inserting padding tokens between panels is not required. Consequently, the resultant sequence, as illustrated in Figure 3, denoted as $\mathcal{F}^{\text{seq}}$, spans a length of $L_t = (8K+7)N_P$, with each token denoted by f_n for $n = 1, \ldots, L_t$. Eventually, we can represent it as

$$\mathcal{F}^{\text{seq}} = \{\mathcal{T}(P_i, S) + C\}_{i=1}^{N_P}, \quad (2)$$

where C is a constant to ensure all tokens are non-negative.

3.2 Generation with Autoregressive Model

Utilizing GPT-based architectures, we adopt a decoder-only transformer to generate token sequences for sewing patterns. Inspired by PolyGen [Nash et al. 2020], we design the triple embedding for each input token: positional embedding, denoting which panel it belongs to; parameter embedding, classifying the token as edge coordinates, rotation, translation, or stitching feature vectors; and value embedding for the quantized sewing pattern values. The source tokens are then input into the transformer decoder to predict the probability distribution of the next potential token at each step. Consequently, the objective is to maximize the log-likelihood of the training sequences:

$$\mathcal{L} = -\prod_{i=1}^{L_t} p\left(f_i \mid f_{<i}; \theta\right). \quad (3)$$

By optimizing this objective, our SewingGPT learns the intricate relationships among the shape, placement, and stitching information of each panel. In the inference stage, target tokens begin with the

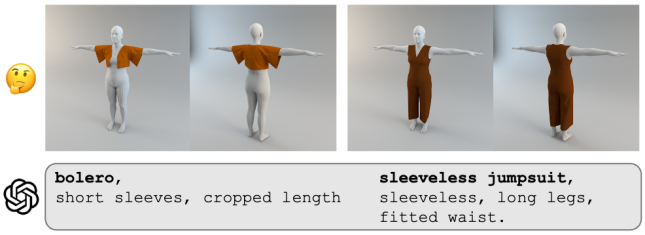

Fig. 4. **Examples of our data captions.** We utilize the rendered images and ask GPT-4V with the designed prompt for detailed captions.

start token and are recursively sampled from the predicted distribution $p\left(\hat{f}_i \mid \hat{f}_{<i}; \theta\right)$ until the end token. Following autoregressive token sequence generation, we reverse token quantization, converting the generated data to its original sewing pattern representation.

Conditional Generation with Text Prompts. To guide the sewing pattern generation, our model integrates cross-attention with text-conditioned embeddings $\mathbf{h}$. Initially, we utilize the CLIP model to obtain the CLIP embedding from input text prompts. Then, we project it into a feature embedding through a trainable compact Multilayer Perceptron (MLP) to condense the dimensionality of CLIP embeddings, matching the Transformer's dimensionality. This approach also boosts memory efficiency and inference speed. Subsequently, the Transformer decoder conducts cross-attention with the feature embedding [Li et al. 2022]. We train the model with pairwise data, facilitating condition-specific token generation.

3.3 Implementation Details

Dataset. We utilize the extensive sewing pattern garment dataset from [Korosteleva and Lee 2021], notable for its comprehensive range of sewing patterns and styles of garments, including shirts, hoods, jackets, dresses, pants, skirts, jumpsuits, vests, etc. Our experiments use approximately 19264 samples across 11 fundamental categories. Each garment in the dataset contains a sewing pattern file, a 3D garment mesh draped on a T-pose human model, and a rendered image. We employ the GPT-4V [Achiam et al. 2023] to generate captions for garments from the rendered images of the front view and the back view. For each garment, we first prompt GPT-4V to generate its common name (e.g., hood, T-shirt, blouse) if available, followed by a request for specific geometric features (e.g., long sleeves, wide garment, deep collar), as demonstrated in Figure 4. We combine these two descriptions to form the caption for each garment. In our experiments, we use the pre-defined order of panels in the dataset for training. Additionally, We utilize about 90% of the data from each category for training and the remaining for validation.

Training. We set $K = 14$, and the maximum length of tokens is 1500. Our decoder-only Transformer consists of 24 layers with position embedding dimensionality of $d_{pos} = 512$, parameter embedding dimensionality of $d_{para} = 512$, value embedding dimensionality of $d_{val} = 512$, and text feature embedding dimensionality of $d_f = 512$. We set constants $C_E = 50, C_R = 1000, C_T = 1000, C_S = 1000$ and, $C = 1000$. Our CLIP embeddings have a dimension of $d_{\text{CLIP}} = 1024$, and condensed feature embeddings have a dimension of $d_{\text{feature}} = 512$. We train our model using Adam optimizer, with a learning rate of 10^{-4} and a batch size of 4. The model is trained on a single A6000 GPU for 30 hours.

4 CUSTOMIZED GARMENT GENERATION

With SewingGPT, we have the capability to generate diverse sewing patterns directly from text prompts. Recognizing appearance's crucial role in the CG pipeline, we aim to generate corresponding Physically-based Rendering (PBR) textures for these patterns, aligning more closely with garment design workflows. By leveraging the SewingGPT and PBR texture generator, our framework DressCode further utilizes a large language model to create customized garments for users through natural language interaction.

4.1 PBR Texture Generation

In some production software commonly utilized by fashion designers, designers often create the texture of the garments after completing the pattern design. For garments, designers usually employ tile-based and physically-based textures such as diffuse, roughness, and normal maps to enhance the realistic appearance of the fabric. Therefore, to generate customized garments, we tailor a pre-trained Stable Diffusion [Rombach et al. 2022] and employ a progressive training approach to generate PBR textures guided by text.

Latent Diffusion Finetuning. Text-to-image generation has advanced significantly with the latent diffusion model (LDM). Existing foundation models, such as Stable Diffusion, trained on billions of data points, demonstrate extensive generalization capabilities. As the original LDM is trained on natural images, adapting it to generate tile-based images is necessary. To achieve this while maintaining the model's generalizability, we collect a PBR dataset with captions and fine-tune the pre-trained LDM on this dataset. We freeze the original encoder $\mathcal{E}$ and decoder $\mathcal{D}$, fine-tuning the U-Net denoiser at this stage. During inference, our fine-tuned LDM is capable of generating tile-based diffuse maps using text prompts.

VAE Finetuning. As we can generate high-quality and tile-based diffuse maps, achieving realistic CG rendering requires us to further generate normal maps U_n and roughness maps U_r based on our generated diffuse maps U_d. In addition to the pretrained LDM encoder $\mathcal{E}$ and decoder $\mathcal{D}$, we fine-tune another two specific decoders $\mathcal{D}_n$ and $\mathcal{D}_r$. With a denoised texture latent code z by text input, which can be decoded into diffuse maps through D, we utilize $\mathcal{D}_n$ and $\mathcal{D}_r$ to decode z into normal maps and roughness maps respectively.

4.2 Customized Generation through User-friendly Interaction

Guided by natural language. Following the implementation of generating sewing patterns and textures through text prompts, our framework in practical scenarios enables designers to interact with the generator using natural language instead of relying on dataset-like formatted prompts. We adopt GPT-4 [Achiam et al. 2023] with content learning to interpret users' natural language inputs, subsequently producing shape prompts and texture prompts. These

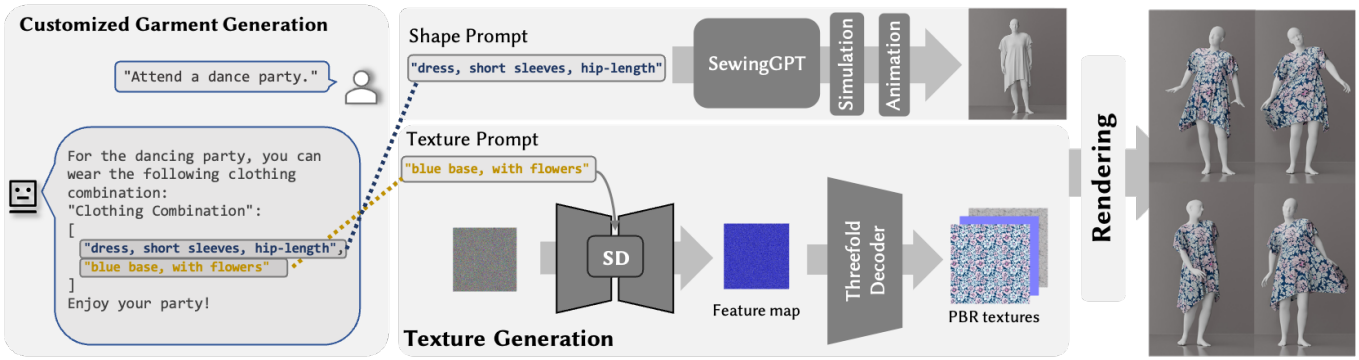

Fig. 5. **Overview of our entire DressCode pipeline for customized garment generation.** We employ a large language model to obtain shape prompts and texture prompts with natural language interaction and utilize the SewingGPT and a fine-tuned Stable Diffusion for high-quality and CG-friendly garment generation.

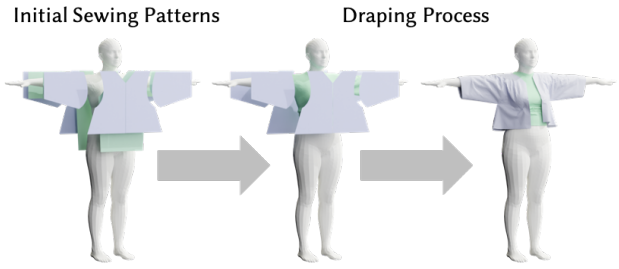

Fig. 6. **Examples of our multiple garments draping process.** Starting with initial sewing patterns, we first drape the inside T-shirt, followed by draping the outside jacket onto the model's body.

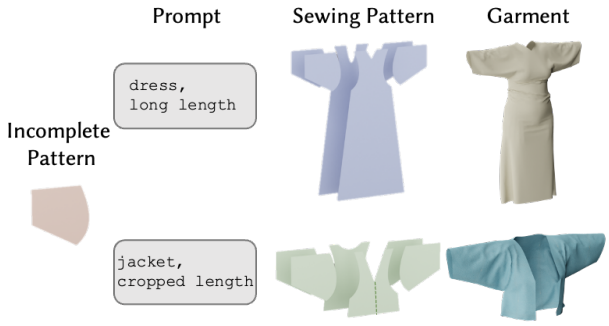

Fig. 7. **Examples of pattern completion.** Given an incomplete pattern, our method infers reasonable sewing pattern completions with various text prompts.

prompts are then fed to the SewingGPT and the PBR texture generator, respectively. Once sewing patterns are generated, we stitch them onto a T-pose human model. Subsequently, the generated garments, along with PBR textures, seamlessly integrate into industrial software, allowing for animation with the human model and rendering under various lighting, ensuring vivid, realistic results.

Multiple garments draping. Production settings usually necessitate generating multiple clothing items (e.g., daily outfits like pants, T-shirts, and jackets) simultaneously. Past 3D content generation studies based on mesh or implicit fields face challenges in effectively achieving layered draping of multiple garments on a target human model. The adoption of sewing pattern representation enables the respective generation of multiple garments and their natural draping onto the human model. In our work, for T-pose results, we use the Qualoth simulator [Choi and Ko 2002] as the physics simulator. We utilize the same material parameters and 3D human model from [Korosteleva and Lee 2021]. In the process of draping multiple garments, we employ an automated sequential multi-garment draping technique, as depicted in Figure 6. Specifically, for a set of clothes, we drape the garment onto the model's body from the inside out. After each simulation, we combine the mesh of the simulated garment and the human model, then perform the next simulation with the subsequent garment.

Pattern Completion. Benefiting from the autoregressive model, our method can complete the entire sewing pattern by utilizing probabilistic predictions provided by the model upon receiving partial pattern information. Additionally, inputting a text prompt can guide the model in completing the sewing patterns. Our work, as illustrated in Figure 7, demonstrates that with a given sleeve, our model adapts to complete various sewing patterns based on different prompts. This enables users to design partial patterns manually and utilize SewingGPT for inspiration and completion of the garments guided by the text prompts.

Texture Editing. In the majority of recent 3D generation tasks, the inability to produce structured UV maps has been a significant impediment, particularly for generating garments. However, our generation method, utilizing sewing pattern representation, enables the creation of distinct and structured UV mappings of each panel. This facilitates convenient texture editing at specific locations, allowing efficient post-processing on the textures. As shown in Figure 8, we demonstrate the SIGGRAPH icon drawn on a cream-colored T-shirt's diffuse map and a duck seamlessly blended with the original grey-colored pants' diffuse map by creating hand-made sketches

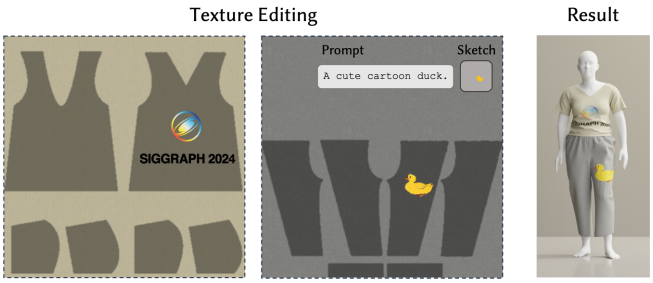

Fig. 8. **Examples of texture editing.** By manually drawing or creating sketches with text prompts, our method facilitates user-friendly texture editing.

	Wonder3D*	RichDreamer	Ours
CLIP score ↑	0.302	0.324	**0.327**
Runtime ↓	~ 4 mins	~ 4 hours	~ **3 mins**
PBR Texture	✗	✓	✓
Texture Editing	✗	✗	✓
Draping	✗	✗	✓

Table 1. **Quantitative and characteristic comparisons on different methods.** Compared to other methods, our method achieves the highest CLIP score and yields several CG-friendly characteristics.

with text prompts in the masked areas (refer to stable-diffusion-webui [AUTOMATIC1111 2022] for detailed implementation).

5 EXPERIMENTS

In this section, we first conduct qualitative and quantitative comparison experiments with other state-of-the-art 3D generation methods to demonstrate the generation capability of our method. We then present ablation studies and validation to evaluate our pipeline. Furthermore, we conduct a qualitative comparison experiment with parametric templates and perform a comprehensive user study to showcase our results compared to other methods. We also verify our results in Marvelous Designer [CLO3D 2024]; we manually load generated garments into the software and then animate the garments with the human model in non-T-pose. We then show a results gallery in Figure 9 of generated high-quality garments using our method with various text prompts, including generated sewing patterns, PBR textures, draping results on a T-pose human model, and animated results with various poses under different illuminations.

5.1 Comparison of 3D Garment Generation

Comparisons on sewing pattern generation. We show some qualitative comparisons with two state-of-the-art sewing pattern works [Korosteleva and Lee 2022; Liu et al. 2023d] in Figure 10, where we present the panel prediction, draped garment on a T-pose human model, and corresponding inputs for each method. Given that NeuralTailor is designed for 3D point cloud inputs and trained on open-surface meshes, we utilize a 3D generation method Surf-D [Yu et al. 2023] using the UDF representation to create meshes conditioned on specific garment categories as inputs (denoted as **NeuralTailor***). Note that Surf-D is trained on the Deep Fashion3D dataset. Although our method supports complex prompts, we only select category names such as *skirt* and *sleeveless dress*, which are presented in the Deep Fashion3D dataset, in our experiment as prompts to ensure a fair comparison. For Sewformer, designed for image inputs, we utilize DALLE-3 [Betker et al. 2023] to synthesize input images from text prompts (denoted as **Sewformer***). While Sewformer is trained on images of human models wearing both upper and lower garments, we synthesize the images with the same rule and extract partial target panels from predicted results for comparison. For the first row of Figure 10, our generated image involves a model wearing both a top shirt and a skirt, not only a sole skirt, which aligns with the training dataset in Sewformer. For fairness, we manually extract skirt patterns from Sewformer predictions, as shown in the comparison. Since the generated input meshes are mostly out of the domain of NeuralTailor's training dataset, the results appear as distorted panels and fail to be stitched together. Sewformer is trained on a new dataset with better generalization; nevertheless, it also encounters issues with irregular and distorted panels, as well as poor garments after stitching. Our method, yielding more accurate results, demonstrates robust generation capabilities with text prompts.

Comparisons on text-to-3D generation. We evaluate the quality of our customized garment generation with various 3D generation methods in Figure 11. We present qualitative comparisons with two state-of-the-art 3D content generation methods: Wonder3D [Long et al. 2023], a 3D creation method from single images, and RichDreamer [Qiu et al. 2023], a text-to-3D work, generating with PBR textures. We adopt DALLE-3 [Betker et al. 2023] to synthesize image inputs for Wonder3D (denoted as **Wonder3D***). Wonder3D takes about 4 minutes to generate garments but fails to retain fine detail and fidelity to the input images, yielding poor geometry. RichDreamer takes approximately 4 hours to optimize and yield more realistic results; however, the generated garments are still blurry for rendering. Furthermore, these generated garments are close-surface meshes, as shown in Figure 11, and fail to adapt to human bodies. In contrast, our method takes about 1 minute to generate sewing patterns, and overall about 3 minutes to generate the simulated garments. It facilitates draping garments on human models in various poses and generating high-quality tile-based PBR textures, achieving realistic rendering.

Additionally, we adopt the CLIP score to quantitatively measure different methods. We generate 15 garments with highly diverse text prompts using each method. Then, we render the generated 3D garments with textures and calculate the CLIP score using the given text prompts. Although our method does not optimize the 3D models for fitting the rendered images to text prompts better, our model achieves the highest CLIP score, demonstrating the effectiveness of our method. These general 3D methods are more broadly applicable than to only 3D garments. However, we also compare several characteristics among different methods, highlighting the advantages of our CG-friendly asset generation in the specific 3D garment domain. The results are shown in Table 1.

Fig. 9. **Our results gallery of DressCode.** We generate sewing patterns, PBR textures, and garments in diverse poses and lighting conditions, guided by various text prompts.

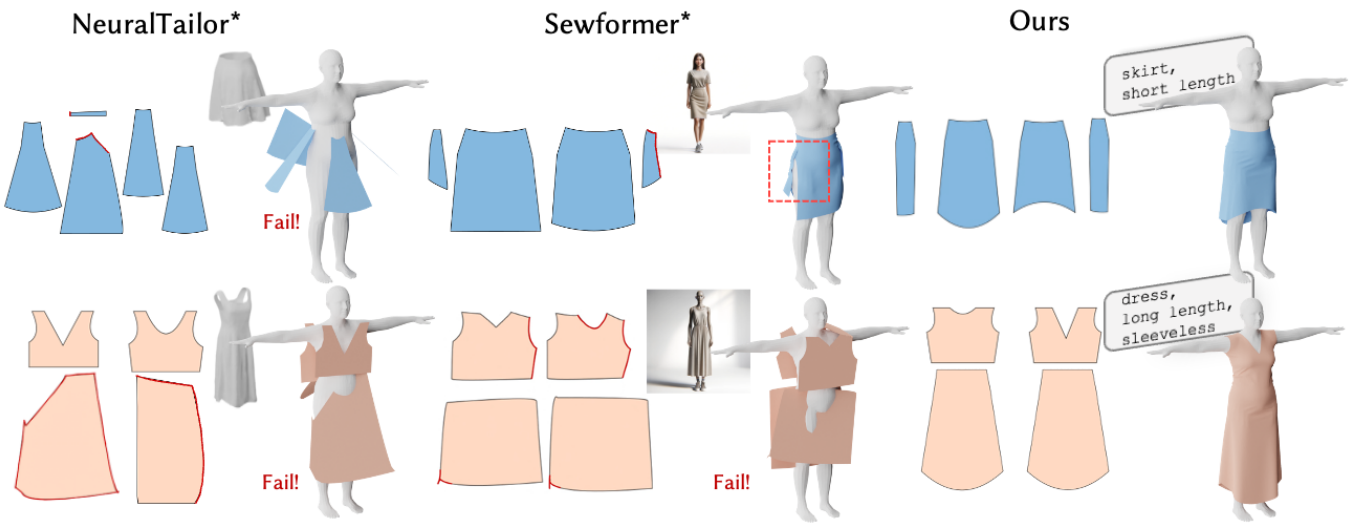

Fig. 10. **Qualitative comparisons on sewing patterns.** Major errors on panels are marked with red edges. The inputs of each method from left to right are meshes, images, and texts, respectively, which are shown along with the results.

Fig. 11. **Qualitative comparisons on text-to-3D generation.** Images on the right side of our methods are the draped garments on human models. Images on the right top of other methods are the top view of each garment. We show our results yield high-quality rendering and the capability to drape on human bodies.

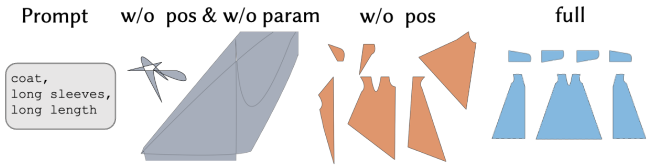

Fig. 12. **Ablation on embeddings.** Better sewing patterns are generated with our designed parameter embedding and positional embedding.

5.2 Ablation Study

We evaluate the performance of our triple embedding used in SewingGPT in Figure 12. We first train the model with only value embedding (denoted as **w/o pos & w/o param**). The results are highly disordered, with strongly distorted panels, containing mismatched stitching information. We then incorporate the parameter embedding (denoted as **w/o pos**), facilitating the model to learn categories (e.g., edge coordinates, rotation parameters, translation parameters, or stitching feature vectors) of each token in a panel, and the results show the better shape of each panel, yet it is still distorted and lacking enough panels. Lastly, we further incorporate the positional embedding (denoted as **full**), enabling the model to distinguish between different panels and the number of panels, leading to the best and complete results.

5.3 Validation

To validate our methods, we design two experiments, as shown in Figure 13. Firstly, we test text prompts from the holdout set as done for training and compare the results in the dataset with our generated outputs. In the first two rows of Figure 13, we test two data points from outside the training dataset and observe that the generated results correspond well to the input prompts. Notably, in the dress example in the first row, although our generated result

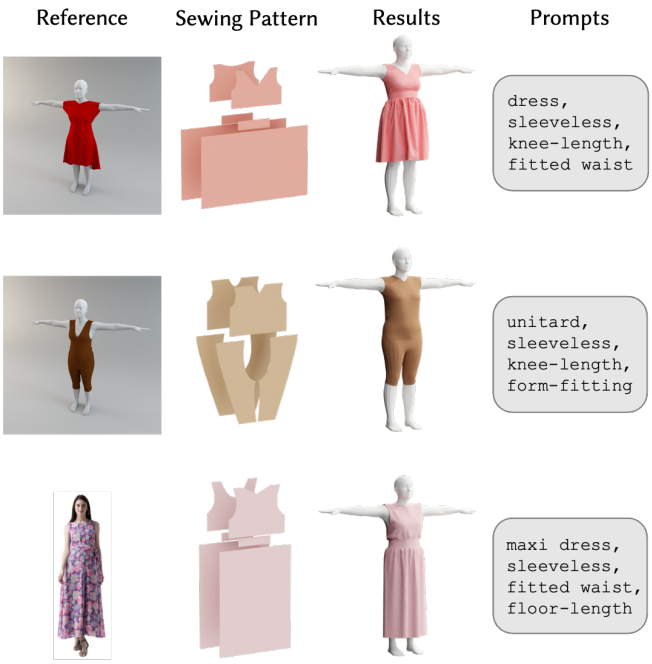

Fig. 13. **Validation.** We validate our generated garments with the text prompts from our dataset out of training and in the wild. Our results align well with the text prompts, demonstrating the effectiveness of our method.

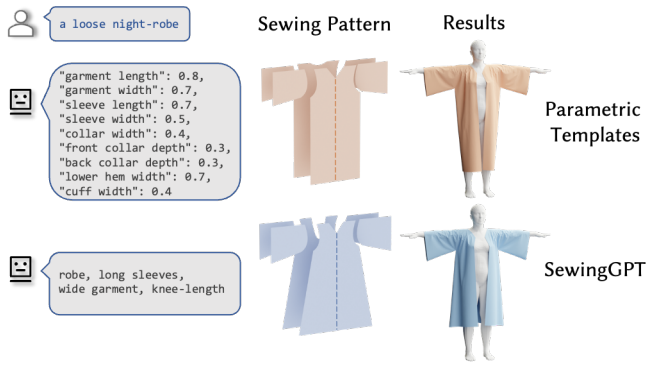

Fig. 14. **Qualitative comparison with Parametric Templates.** The user inputs the prompt "a loose night-robe," and we show the results from Parametric Templates and our proposed SewingGPT.

includes a waistband while the reference example does not, this still aligns well with the "fitted waist" attribute mentioned in the prompt. Furthermore, we test an image in the wild from the Deep Fashion3D dataset [Zhu et al. 2020] and utilize the method described in Section 3.3 with GPT-4V [Achiam et al. 2023] to generate the caption. This serves as the input to qualitatively compare our result with the reference image, revealing that our result closely resembles the input image. This demonstrates that our work effectively bridges the gap between conceptual textual descriptions and their practical, visual counterparts in garment design.

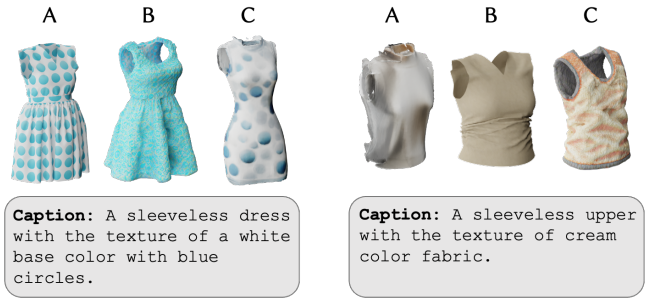

Fig. 15. **Examples from our user study.** We present two cases for users to select the best results from three options: **A**, **B**, and **C**. The left group of results is generated by *Ours*, *RichDreamer*, and *Wonder3D**, respectively, and the right group of results is generated by *Wonder3D**, *Ours*, and *RichDreamer*, respectively. The generation method of each image is not disclosed to the users.

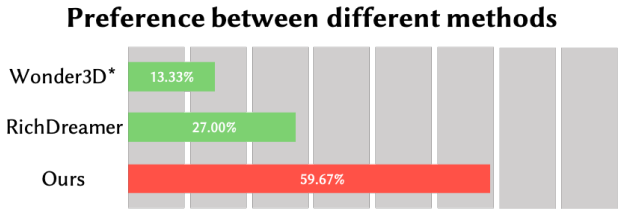

Table 2. **Quantitative results of user study.** It demonstrates the higher user preference for our method compared to other methods.

5.4 Comparison with Parametric Templates

[Korosteleva and Lee 2021] propose a method of a flexible description structure for specifying parametric sewing pattern templates. An intuitive idea is that we can control the parameters of the predefined parametric sewing pattern templates to generate diverse garments (denoted as **Parametric Templates**). To facilitate interaction through natural language and benefit from the strong capability of content learning in ChatGPT, we employ GPT-4 [Achiam et al. 2023] in our experiment, designing prompts to enable its role as a garment design assistant, providing formatted outputs. Subsequently, we inquire about the description of garments, prompting it to output parameters for a specific template. As illustrated in the middle left of Figure 14, GPT-4 responds with several parameters when we ask with "a loose night-robe." We qualitatively compare the results from Parametric Templates and our SewingGPT in Figure 14, observing that both methods generate reasonable results. Nevertheless, our proposed SewingGPT is more adaptable to the diverse categories of data, as it does not require selecting pre-defined templates and designing prompts specifically for ChatGPT's content learning. Additionally, our methods enable us to extend beyond such parametrized datasets for more complex sewing patterns.

5.5 User Study

We conduct a comprehensive user study to evaluate the quality of our generated customized garments, particularly their alignment with given text prompts and overall quality. Given 20 text prompts,

including descriptions of the shape and texture of garments, we then render the generated results for each method and shuffle the order of results obtained by different methods, as shown in Figure 15. Then we ask 30 users to select the best results from all candidates with comprehensive consideration for two aspects: conformity to the captions of prompt texts regarding garments' shape and texture, and the visualized quality and fidelity of the rendered garments. As illustrated in Table 2, the preference results clearly indicate a significant advantage of our method over competing approaches in both aspects, highlighting its superiority in aligning closely with the textual prompts and producing visually appealing and high-fidelity garments.

5.6 Limitations and Discussions

Despite producing high-quality garment generation from text guidance, our method encounters certain limitations. One limitation is that the current sewing pattern dataset limits the generation of multi-layered garments, such as "hoodie jacket with a pocket," as depicted in the first row of Figure 16. It underscores the importance of dataset expansion to include more complex stitching relationships. Another limitation is that our model struggles with prompts outside the domain of our dataset. For instance, we test prompts like "one-shoulder dress." As shown in the second row of Figure 16, the model still generates a "two-shoulder dress," due to the absence of "one-shoulder" garments in our dataset, which hinders its ability to recognize this attribute. Similarly, we experiment with integrating unusual characteristics into garments, combining specific attributes from different categories of garments, such as a "dress with a hood," a style not commonly encountered in real life. Our results shown in the last row of Figure 16 display a very loose hoodie jacket. Although the results somewhat resemble a dress with its loose style, a dress should not be open-front. This outcome is due to the presence of only hoodie jackets as hooded garments in our dataset, leading to a bias toward producing results within the hoodie jacket category when the prompt includes a "hood." We believe that enriching the dataset with a wider variety of garments can significantly enhance the model's versatility. Additionally, inspired by recent breakthroughs in the generation domain, distilling knowledge from the pre-trained foundation model, such as SDS [Poole et al. 2022], to improve generalization is a worthy direction for future work. Lastly, although our framework is pioneering in generating garments with text prompts, incorporating multi-modality inputs could prove more effective. Generating sewing patterns and textures controlled by both text and images presents a particularly intriguing and challenging problem yet to be addressed in real-world applications.

Potential ethical implications. The text-driven generation method is subject to biases inherent in underlying pre-trained models such as CLIP and Stable Diffusion. Its user-friendliness and high-quality outputs also carry potential risks for misuse, emphasizing the necessity for future initiatives to tackle these ethical concerns through bias mitigation and thorough review. Additionally, the utilization of Stable Diffusion for both fine-tuning and inference purposes raises significant concerns regarding potential copyright issues, as the model may inadvertently generate content that mirrors proprietary works without explicit authorization. It is important for future work to ensure that the training data and generated content of these models are carefully reviewed and selected.

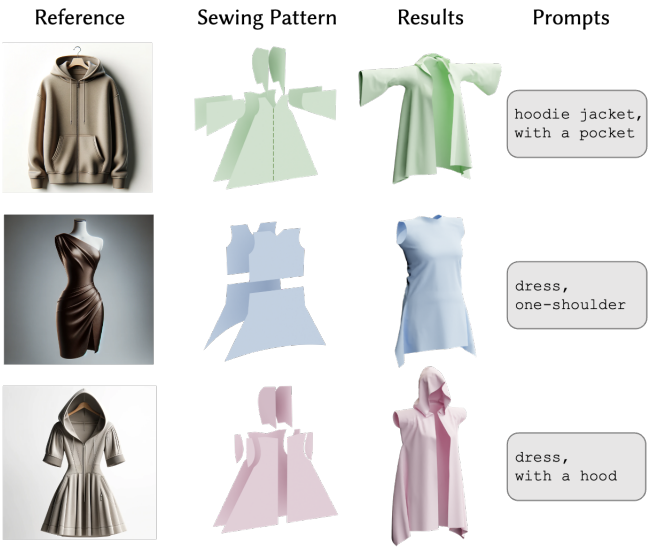

Fig. 16. **Failure cases.** Our method struggles to generate garments that fall outside the domain of the training dataset. We present three examples of generated results: a "hoodie jacket with a pocket," a "one-shoulder dress," and a "dress with a hood." The reference images are generated by DALLE-3 [Betker et al. 2023].

6 CONCLUSIONS

In conclusion, this paper introduces *DressCode*, a novel customized garment generation framework featuring a text-driven sewing pattern generator SewingGPT. This framework democratizes garment design by making it accessible and interactive, enabling both novices and experts to generate detailed sewing patterns and high-quality PBR textures through simple textual prompts. Additionally, our framework supports interaction-friendly applications for garment generation, completion, and editing, providing powerful tools that enable designers to unleash their creative imagination. Our experimental results and user study demonstrate the effectiveness of our method in producing CG-friendly garments that excel in quality and alignment with input prompts.

In contrast to past work, our approach democratizes fashion design through enhanced accessibility and interactivity and improves the practical utility of digital garments in CG pipelines for post-editing, animation, and realistic rendering. The ease of use and innovative approach of DressCode promise exciting developments for future advancements in digital garments, potentially transforming digital garment creation and customization. We envision our method benefiting CG production and advancing the digital garment landscape for virtual try-on, fashion design, and digital human creation.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2022YFF0902301), NSFC programs (61976138, 61977047), STCSM (2015F0203-000-06), and SHMEC(2019-01-07-00-01-E00003). We also acknowledge support from the Shanghai Frontiers Science Center of Human-centered Artificial Intelligence and MoE Key Lab of Intelligent Perception and Human-Machine Collaboration (ShanghaiTech University).

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

Autodesk, INC. 2019. *Maya.* https://autodesk.com/maya

AUTOMATIC1111. 2022. *Stable Diffusion Web UI.* https://github.com/AUTOMATIC1111/stable-diffusion-webui

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

Floraine Berthouzoz, Akash Garg, Danny M Kaufman, Eitan Grinspun, and Maneesh Agrawala. 2013. Parsing sewing patterns into 3D garments. *Acm Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf* 2 (2023), 3.

Blender Foundation. 2022. *Blender.* https://www.blender.org/

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023b. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396* (2023).

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *arXiv preprint arXiv:2303.13873* (2023).

Xipeng Chen, Guangrun Wang, Dizhong Zhu, Xiaodan Liang, Philip Torr, and Liang Lin. 2022. Structure-Preserving 3D Garment Modeling with Neural Sewing Machines. *Advances in Neural Information Processing Systems* 35 (2022), 15147–15159.

Xiaowu Chen, Bin Zhou, Feixiang Lu, Lin Wang, Lang Bi, and Ping Tan. 2015. Garment modeling with a depth camera. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–12.

Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but responsive cloth. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 604–611.

CLO3D. 2024. *Marvelous Designer.* https://www.marvelousdesigner.com/

Luca De Luigi, Ren Li, Benoît Guillard, Mathieu Salzmann, and Pascal Fua. 2023. DrapeNet: Garment Generation and Self-Supervised Draping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1451–1460.

Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. 2023. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. *arXiv preprint arXiv:2303.17015* (2023).

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Yangyi Huang, Hongwei Yi, Yuliang Xiu, Tingting Liao, Jiaxiang Tang, Deng Cai, and Justus Thies. 2023. Tech: Text-guided reconstruction of lifelike clothed humans. *arXiv preprint arXiv:2308.08545* (2023).

Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. 2022. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 867–876.

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.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

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*, J. Vanschoren and S. Yeung (Eds.), Vol. 1. https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/013d407166ec4fa56eb1e1f8cbe183b9-Paper-round1.pdf

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

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*. PMLR, 12888–12900.

Ren Li, Benoît Guillard, and Pascal Fua. 2023b. ISP: Multi-Layered Garment Draping with Implicit Sewing Patterns. *arXiv preprint arXiv:2305.14100* (2023).

Yifei Li, Hsiao-yu Chen, Egor Larionov, Nikolaos Sarafianos, Wojciech Matusik, and Tuur Stuyck. 2023a. DiffAvatar: Simulation-Ready Garment Optimization with Differentiable Simulation. *arXiv preprint arXiv:2311.12194* (2023).

Tingting Liao, Hongwei Yi, Yuliang Xiu, Jiaxaing Tang, Yangyi Huang, Justus Thies, and Michael J Black. 2023. Tada! text to animatable digital avatars. *arXiv preprint arXiv:2308.10899* (2023).

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 300–309.

Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. 2023d. Towards Garment Sewing Pattern Reconstruction from a Single Image. *ACM Transactions on Graphics (SIGGRAPH Asia)* (2023).

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. 2023c. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *arXiv preprint arXiv:2306.16928* (2023).

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023b. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9298–9309.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2023a. SyncDreamer: Generating Multiview-consistent Images from a Single-view Image. *arXiv preprint arXiv:2309.03453* (2023).

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. 2023. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008* (2023).

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11461–11471.

Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. 2023. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8446–8455.

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-nerf for shape-guided generation of 3d shapes and textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12663–12673.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. 2022. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8280–8290.

Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. 2020. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*. PMLR, 7220–7229.

Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational pattern making from 3D garment models. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2022. DreamFusion: Text-to-3D using 2D Diffusion. *arXiv* (2022).

Anran Qi, Sauradip Nag, Xiatian Zhu, and Ariel Shamir. 2023. PersonalTailor: Personalizing 2D Pattern Design from 3D Garment Point Clouds. *arXiv preprint arXiv:2303.09695* (2023).

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. 2023. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843* (2023).

Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. 2023. RichDreamer: A Generalizable Normal-Depth Diffusion Model for Detail Richness in Text-to-3D. *arXiv preprint arXiv:2311.16918* (2023).

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

Amit Raj, Srinivas Kaza, Ben Poole, Michael Niemeyer, Nataniel Ruiz, Ben Mildenhall, Shiran Zada, Kfir Aberman, Michael Rubinstein, Jonathan Barron, et al. 2023.

Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508* (2023).

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721* (2023).

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 conference on computer vision and pattern recognition*. 10684–10695.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

Nikolaos Sarafianos, Tuur Stuyck, Xiaoyu Xiang, Yilei Li, Jovan Popovic, and Rakesh Ranjan. 2024. Garment3DGen: 3D Garment Stylization and Texture Generation. *arXiv preprint arXiv:2403.18816* (2024).

Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. 2023. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937* (2023).

Nicholas Sharp and Keenan Crane. 2018. Variational surface cutting. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 6087–6101.

Yu Shen, Junbang Liang, and Ming C Lin. 2020. Gan-based garment generation using sewing pattern images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*. Springer, 225–247.

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. 2023a. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110* (2023).

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. 2023b. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512* (2023).

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2023. MeshGPT: Generating Triangle Meshes with Decoder-Only Transformers. *arXiv preprint arXiv:2311.15475* (2023).

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 Transactions on Visualization and Computer Graphics* 28, 4 (2020), 1862–1879.

Zhaoqi Su, Tao Yu, Yangang Wang, and Yebin Liu. 2022. Deepcloth: Neural garment representation for shape and style editing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2022), 1581–1593.

Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. 2023. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184* (2023).

Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. 2023. TextMesh: Generation of Realistic 3D Meshes From Text Prompts. *arXiv preprint arXiv:2304.12439* (2023).

Nobuyuki Umetani, Danny M Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive couture for interactive garment modeling and editing. *ACM Trans. Graph.* 30, 4 (2011), 90.

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. 2023a. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12619–12629.

Tuanfeng Y Wang, Duygu Ceylan, Jovan Popović, and Niloy J Mitra. 2018. Learning a shared shape space for multimodal garment design. *ACM Transactions on Graphics* 37, 6 (2018), 1–13.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023b. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. *arXiv preprint arXiv:2305.16213* (2023).

Tong Wu, Zhibing Li, Shuai Yang, Pan Zhang, Xingang Pan, Jiaqi Wang, Dahua Lin, and Ziwei Liu. 2023. HyperDreamer: Hyper-Realistic 3D Content Generation and Editing from a Single Image. In *SIGGRAPH Asia 2023 Conference Papers*. 1–10.

Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. 2023. NeuralLift-360: Lifting an In-the-Wild 2D Photo to a 3D Object With 360deg Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4479–4489.

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 Transactions on Graphics (TOG)* 37, 5 (2018), 1–14.

Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. 2023. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. *arXiv preprint arXiv:2310.03020* (2023).

Zhengming Yu, Zhiyang Dou, Xiaoxiao Long, Cheng Lin, Zekun Li, Yuan Liu, Norman Müller, Taku Komura, Marc Habermann, Christian Theobalt, et al. 2023. Surf-D: High-Quality Surface Generation for Arbitrary Topologies using Diffusion Models. *arXiv preprint arXiv:2311.17050* (2023).

Longwen Zhang, Qiwei Qiu, Hongyang Lin, Qixuan Zhang, Cheng Shi, Wei Yang, Ye Shi, Sibei Yang, Lan Xu, and Jingyi Yu. 2023a. DreamFace: Progressive Generation of Animatable 3D Faces under Text Guidance. *arXiv preprint arXiv:2304.03117* (2023).

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023b. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.

Minda Zhao, Chaoyi Zhao, Xinyue Liang, Lincheng Li, Zeng Zhao, Zhipeng Hu, Changjie Fan, and Xin Yu. 2023. EfficientDreamer: High-Fidelity and Robust 3D Creation via Orthogonal-view Diffusion Prior. *arXiv preprint arXiv:2308.13223* (2023).

Yuxiao Zhou, Menglei Chai, Alessandro Pepe, Markus Gross, and Thabo Beeler. 2023. GroomGen: A High-Quality Generative Hair Model Using Hierarchical Latent Representations. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–16.

Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. 2020. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 512–530.