AIpparel: A Multimodal Foundation Model for Digital Garments

Kiyohiro Nakayama[1*] Jan Ackermann[1,2*†] Timur Levent Kesdogan[1,2*†]
Yang Zheng[1] Maria Korosteleva[2] Olga Sorkine-Hornung[2]
Leonidas J. Guibas[1] Guandao Yang[1] Gordon Wetzstein[1]
[1]Stanford University [2]ETH Zürich

Figure 1. **AIpparel.** We present a multimodal foundation model for digital garments trained by fine-tuning a large multimodal model on a custom sewing pattern dataset using a novel tokenization scheme for these patterns. AIpparel generates complex, diverse, high-quality sewing patterns based on multimodal inputs, such as text and images, and it unlocks new applications such as language-instructed sewing pattern editing. The generated sewing patterns can be directly used to simulate the corresponding 3D garments.

Abstract

Apparel is essential to human life, offering protection, mirroring cultural identities, and showcasing personal style. Yet, the creation of garments remains a time-consuming process, largely due to the manual work involved in designing them. To simplify this process, we introduce AIpparel, a multimodal foundation model for generating and editing sewing patterns. Our model fine-tunes state-of-the-art large multimodal models (LMMs) on a custom-curated large-scale dataset of over 120,000 unique garments, each with multimodal annotations including text, images, and sewing patterns. Additionally, we propose a novel tokenization scheme that concisely encodes these complex sewing patterns so that LLMs can learn to predict them efficiently. AIpparel achieves state-of-the-art performance in single-modal tasks, including text-to-garment and image-to-garment prediction, and enables novel multimodal garment generation applications such as interactive garment editing. The project website is at `https://georgenakayama.github.io/AIpparel/`.

*Equal Contribution.
†Work done as a visiting researcher at Stanford.

1. Introduction

Clothing plays a crucial role in society, providing protection from the weather, reflecting societal norms, and serving as a means of personal expression. A key stage in garment production is the development of sewing patterns—a set of flat 2D panels with standardized assembly instructions that form a complete 3D garment [4]. Pattern making is a challenging task due to the complex geometric relationship between the 2D pattern and the draped 3D shape of the sewn garment. Even an experienced tailor must go through multiple iterations, incorporating feedback from various sources, including verbal descriptions of the garment's fit and feel, as well as visual references of its appearance. To simplify the pattern-making process, we explore strategies to leverage emerging generative models with multimodal input, such as text and images.

State-of-the-art sewing pattern prediction methods are designed to work with one specific input modality, such as 3D points [6, 14, 29], images [46, 50, 70, 75, 79], or language [21]. While effective within their respective domains, these single-modal approaches are often challenging to adapt to garment prediction tasks requiring different or combined input modalities. Expanding these methods to multimodal pattern prediction presents two primary challenges. First, no large-scale multimodal sewing pattern dataset is publicly available. Second, the capacity to accurately interpret multimodal inputs typically only emerges in large models with billions of parameters [1, 36]. It remains uncertain how to efficiently scale existing methods to models of this size.

In this paper, we propose to build such multimodal garment generative models by extending an existing large multimodal model (LMM) [36, 62, 74] to understand sewing patterns with complex geometries. To achieve this, we annotate the largest sewing pattern dataset [31] with multimodal labels. Our annotated dataset is ten times larger than those used by previous state-of-the-art generative methods [21, 29, 38], including over 120,000 unique sewing patterns paired with detailed text descriptions, images, edited sewing patterns, and editing instructions. Fine-tuning an LMM to perform multimodal garment generation requires representing complex garments in a format that LMMs can understand—tokens. For this task, we develop a novel tokenization method that is both expressive in representing the complex geometries of sewing patterns and concise enough to fit within the limited context length of existing LMMs, making fine-tuning computationally efficient.

Combining these components, we present AIpparel, a large multimodal model for generating sewing patterns. AIpparel can predict sewing patterns with complex geometries and outperforms state-of-the-art methods in single-modal garment prediction, often by a large margin. Moreover, our approach unlocks entirely new multimodal garment generation tasks. Our contributions include:

- We present GCD-MM, a multimodal sewing pattern dataset extending the largest public dataset of sewing patterns with multimodal annotations. We plan to publicly release the dataset to inspire innovative garment prediction capabilities and further enable research in multimodal garment generation.
- We develop a novel tokenization scheme and a new training objective for fine-tuning LMMs to predict sewing patterns. This tokenization method is critical for retargeting LMMs to multimodal garment prediction tasks efficiently.
- We present AIpparel, the first multimodal foundation model for sewing pattern prediction capable of taking language, images, and sewing patterns as input.

2. Related Works

Garment Generation. Prior works have studied learning-based garment generation represented in various formats, including images [5, 27], 3D meshes [17, 26, 33, 40, 41, 44, 49, 54, 57, 60, 81, 83, 85, 86], and sewing patterns [21, 29, 38, 50, 68]. Our paper focuses on generating sewing patterns, which, compared to other representations, are the industry standard and can be directly used for downstream simulation and manufacturing. Earlier works have explored a variety of different ways to generate and predict sewing patterns, including retrieval-based methods [14, 20], predicting sewing pattern templates with few parameters [25, 64, 65, 76], or cutting 3D scans into 2D panels [16, 18, 37, 43, 55, 67]. These algorithms usually require heuristics, such as the output garment templates. This limits their flexibility in extending to different input modalities or more complex garment types. Further, researchers have successfully applied deep learning methods to generate sewing patterns [21, 29, 38]. While these methods can predict accurate sewing patterns based on input conditioning, they are task-specific models designed to work well only in a single modality. Extending these single-modal methods to a novel modality is difficult, in part because of the lack of large-scale multimodal sewing-pattern datasets and the requirement to redesign the network architecture. While Wang et al. [68] can predict sewing patterns from multiple modalities, including images, 3D garments, and body measurements, their method is limited to predicting simple garments with a predefined set of parameters. In this paper, we aim to tackle the challenge of creating a large multimodal generative model by curating the first multimodal garment dataset with complex garment geometries and providing a scalable recipe building on existing large multimodal models.

Extending Large Multimodal Models. Large multimodal models have gained significant attention for their ability to understand language and images [2, 15, 48, 58, 62]. Efforts to extend LMMs to additional domains typically fall into two categories. Optimization-free approaches [24, 39, 56, 69, 71, 74, 77] employ prompt engineering. The other option is to fine-tune LMMs to take the new modality as input and/or output. The latter approach was first introduced for vision-language models [36, 61, 84] and subsequent works extended it to other modalities [11, 19, 32, 34, 72, 80]. Their approaches typically involve using pre-trained encoders [32, 34, 72] or standard discrete representations [11, 80] to convert the input modalities into tokens and align them with the text feature space of the LMMs. In particular, LLaVA [36] is pioneering in fine-tuning Large Language Models (LLMs) for visual understanding. It uses a pre-trained vision encoder to encode images into tokens, and a trainable projection layer to project the visual tokens into the LLM's feature space. We build our work on top of LLaVA by fine-tuning it to understand sewing patterns. This presents unique challenges, however, due to the lack of pretrained encoders or learning-efficient representations for sewing patterns. This motivates us to design an efficient, learning-friendly tokenizer and a fine-tuning objective for sewing pattern prediction.

Garment Datasets. Garment datasets mostly fall into one of the following three categories: 1) datasets based on 3D scans of real-world garments [3, 9, 22, 41, 59, 73, 85], 2) datasets of designer-created garments [10, 87], and 3)

Dataset	Total	Text	Image	Edits
Wang et al. [68]	8k	✗	✓	✓
Korosteleva and Lee [28]	23.5k	✗	✓	✗
Sewfactory [38]	19.1k	✗	✓	✗
DressCode [21]	20.3k	✓	✗	✗
GCD [31]	130k	✗	✓	✗
GCD-MM (Ours)	120k	✓	✓	✓

Table 1. **Modalities of Sewing Patter Datasets.** GCD-MM is a large-scale sewing pattern dataset with multimodal annotations, including text, images, and edited patterns.

datasets containing mostly procedurally generated sewing patterns [7, 26, 28, 38, 45, 51, 63, 66, 68]. While 3D garment scans and designer-created garments can accurately capture the real-world complexity of garments, they are expensive to obtain, which limits the scale of these categories of data. Our work focuses on leveraging large-scale procedurally generated sewing pattern datasets. To the best of our knowledge, the largest synthetic sewing pattern datasets available are DressCode [21], SewFactory [38], and GarmentCodeData (GCD) [30, 31]. None of their annotations, however, contain the full combinations of text, images, and sewing pattern edits, making them insufficient for training a multimodal sewing pattern generative model. To overcome this data gap, we curate the first large-scale multimodal sewing pattern dataset by expanding GCD with annotations including text, editing pairs, and editing instructions. Tab. 1 compares different sewing pattern datasets and their annotation modalities.

3. Method

We propose a large multimodal generative model for sewing patterns by fine-tuning existing LMMs on a multimodal sewing pattern dataset. For this purpose, we first curate a sewing pattern dataset with multimodal annotations (Sec. 3.1). We then describe how to train our model, *AIpparel*, using an efficient tokenization scheme for sewing patterns, with LlaVA 1.5-7B [36] as a base model (Sec. 3.2).

3.1. Multimodal GarmentCode Dataset

We create annotations covering many modalities to train a multimodal sewing pattern generative model. Specifically, we build on top of the largest existing sewing-pattern dataset, GarmentCodeData (GCD) [31], to incorporate two other modalities: text descriptions and sewing pattern pairs with editing instructions. We dub our dataset GarmentCodeData-MultiModal (GCD-MM).

Text description of sewing patterns. To enable applications such as text-conditioned sewing pattern generation, it is important to obtain detailed text annotation describing the sewing patterns [8, 21]. He et al. [21] created short keyword descriptions of sewing patterns by prompting GPT4V with rendered images. However, this method suffers from hallucination, and the short keywords are insufficient to describe the garments in detail, leading to ambiguities. Our pipeline improves on this by leveraging the design parameters associated with each synthetically generated sewing pattern to create accurate descriptions that capture the garment's key features. Specifically, we develop a rule-based algorithm to generate a set of short phrases, including a garment type (e.g., "midi dress", "godet skirt") and brief descriptions based on distinctive characteristics (e.g., "flared hem", "V-neckline"). To obtain the final sewing pattern description, we prompt GPT-4o [74] using the rule-based short captions and the rendered views of the draped garment. Our approach reduces GPT-4o's hallucination and results in more accurate descriptions in natural language. Please refer to the supplementary for caption comparison with DressCode and the prompts and rules we used to generate them.

Language-instructed Sewing Pattern Editing. We also augment GCD with language-instructed editing annotations. Specifically, we use the programming abstraction from GarmentCode [30] to create paired sewing patterns with corresponding text instructions describing the applied edits. We first manually specify a series of sewing pattern edits using the abstraction. This includes edits such as adjustments in skirt and pants length, changing insert and neckline styles, and adding or excluding a hood or sleeve. For each modification, we generate captions using a text template to describe the applied changes. See the supplementary for editing templates and captions examples.

3.2. AIpparel

AIpparel fine-tunes LLaVA 1.5-7B on our GCD-MM dataset to generate sewing patterns from multimodal input. For this purpose, we need to encode sewing patterns into a compact list of tokens for LLaVA's input. We also propose a novel fine-tuning objective that allows AIpparel to generate both discrete tokens and continuous parameters. Figure 2 shows an overview of our method.

Pattern Representation. Following GCD [31], we define sewing patterns as a set of 2D panels in 3D with stitching information. A sewing pattern $\mathcal{P} = (P, S)$ is a tuple consisting of N panels $P = \{P_1, \ldots, P_N\}$ and stitching information S. Each panel P_i is a planar surface with vertices $V_i = \{v_1^{(i)}, \ldots, v_{n_i}^{(i)}\}$ and edges $E_i = (e_1^{(i)}, \ldots, e_{n_i}^{(i)})$, where each edge contains two endpoints connecting $(v_k^{(i)}, v_{k'}^{(i)})$ with $k' = k \mod n_i + 1$. Since each panel is defined in its own coordinate frame, we always set $v_1^{(i)} = 0 \in \mathbb{R}^2$. An edge can be a straight line, a quadratic or cubic Bézier curve, or an arc, and includes its corresponding control vertices $\boldsymbol{c}_k^{(i)}$. Each panel also includes a rigid 3D transformation R that transforms P_i into the global coordinate frame for draping. Lastly, each panel contains a unique name indicating the panel type for the designers. We define stitching information S as a set of edge pairs among panel

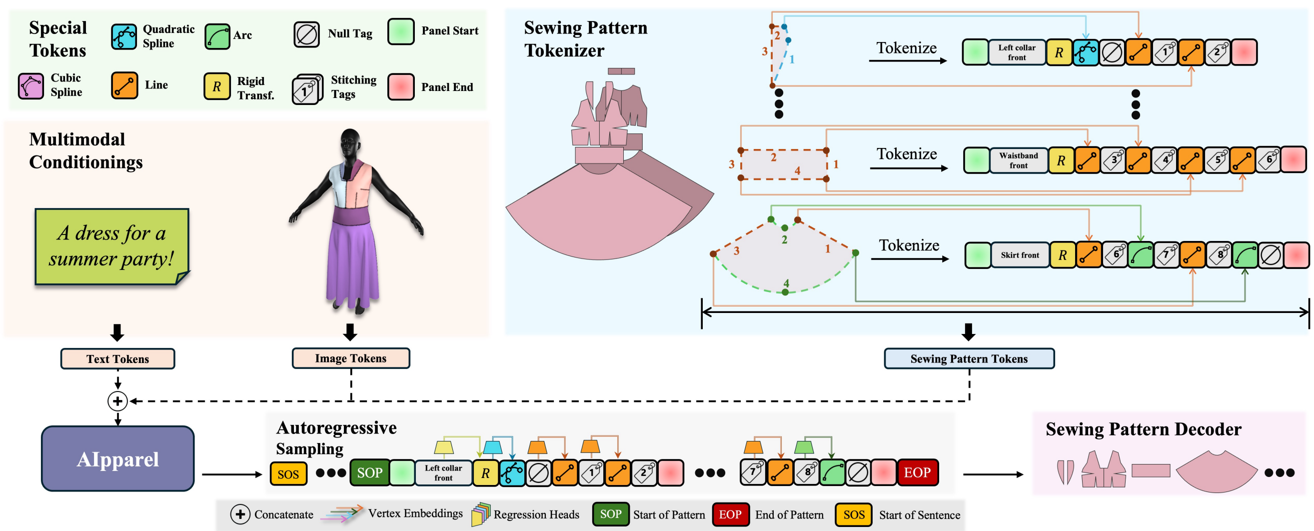


Figure 2. **Illustration of Our Method**. AIpparel uses a novel sewing pattern tokenizer (light blue region) to tokenize each panel into a set of special tokens (light green region). Panel vertex positions and 3D transformations are incorporated using positional embeddings (colored arrows) to the tokens. AIpparel takes in multimodal inputs, such as images and texts (light orange region), to output sewing patterns using autoregressive sampling (light grey region). Finally, the output is decoded to produce simulation-ready sewing patterns (light pink region). See Section 3 for method details.

edges, i.e., $S = \{(e_{k_1}^{(i_1)}, e_{l_1}^{(j_1)}), \dots, (e_{k_m}^{(i_m)}, e_{l_m}^{(j_m)})\}$ where each $(e_{k_s}^{(i_s)}, e_{l_s}^{(j_s)})$ indicates that edge $e_{k_s}^{(i_s)}$ from panel P_{i_s} will be stitched with edge $e_{l_s}^{(j_s)}$. See the supplementary for representation details.

Sewing Pattern Tokenization. The sewing pattern representation in GCD contains both continuous parameters, such as panel vertex coordinates, and discrete parameters, such as the number of panels and stitches. This poses challenges in representing each sewing pattern compactly as a set of tokens for the transformer's prediction. Prior works rely on extensive zero-padding to ensure that all sewing patterns can be represented as a fixed-length vector [21, 29, 38]. This approach is impractical for the complex sewing patterns found in GCD-MM, as it produces an excessively long context. For example, the tokenization scheme of He et al. [21] requires more than 30k tokens to represent a typical sewing pattern in the GCD-MM dataset, making it extremely inefficient for generation and learning.[1]

Inspired by recent work on vector graphics generation [13], we develop a tokenization scheme that efficiently represents sewing patterns as a sequence of drawing commands. Specifically, we introduce four special tokens to indicate the start of a garment (<SoG>) and the end of a garment (<EoG>), as well as the start of a panel (<SoP>) and the end of a panel (<EoP>). With these tokens, each sewing pattern can be represented as

$$\mathrm{E}_g(\mathcal{P}) = \texttt{<SoG>}\mathrm{E}_p(P_1, S)\cdots\mathrm{E}_p(P_n, S)\texttt{<EoG>}, \quad (1)$$

where E_p tokenizes panel P in the form of <SoP>...<EoP>. E_p consists of three pieces of panel information: name, transformation, and edges. The panel name is tokenized using LLaVA-1.5-7B's text tokenizer and inserted after <SoP>. We introduce a new token <R> and place it after the panel name to represent the panel's transformation. Each edge type also corresponds to two special tokens, depending on whether the edge ends at the starting endpoint: line (<L>, <cL>), quadratic Bézier curve (<Q>, <cQ>), cubic Bézier curve (<B>, <cB>), and arc (<A>, <cA>). We also introduce a set of stitching tag tokens $\{\texttt{<t1>}, \dots, \texttt{<tM>}, \texttt{<tN>}\}$ to represent stitching information S. We associate each edge with a stitching tag so that $(e_{k_s}^{(i_s)}, e_{l_s}^{(j_s)}) \in S$ iff there exists $a \in \{1, \dots, M\}$ such that $e_{k_s}^{(i_s)}$ and $e_{l_s}^{(j_s)}$ are both associated with <Ta>. If an edge is not stitched to another edge, it is associated with the null tag <tN>. For example, a panel consisting of two lines stitched together, one cubic Bézier curve and an arc is tokenized as

<SoP>[panel name]<R><L><t1><L><t1>
<B><tN><cA><tN><EoP>.

Compared to the DressCode tokenizer [21], our proposed scheme uses around 100 times fewer tokens to describe the same garment. On average, we represent a sewing pattern with around 250 tokens with a maximum of 838 tokens on GCD-MM, whereas DressCode uses more than 30k tokens for each sewing pattern on the same data.

Notation. From now on, we use bold letters (e.g., $\boldsymbol{X} \in \mathbb{R}^{N \times D}$) to denote the input embedding sequence to the

[1] See supplementary for a detailed analysis.

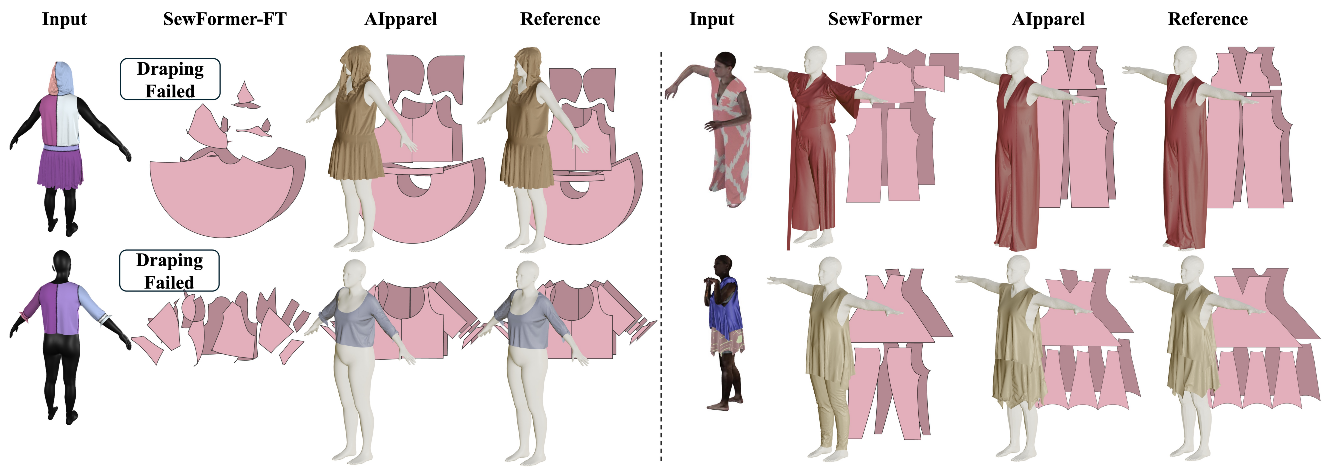


Figure 3. **Image-to-Garment Prediction (Qualitative).** *GCD-MM (Left)*: our model can reconstruct suitable sewing patterns from the input image alone. In contrast, SewFormer does not produce simulation-ready sewing patterns despite fine-tuning. *SewFactory (Right)*: SewFormer produces inaccurate panels (top row) and incorrect garment types (bottom row) while AIpparel accurately recovers sewing patterns from the images, resulting in superior simulation results. See Sec. 4.1.

Dataset	Method	Panel L2 (↓)	#Panel Acc (↑)	#Edge Acc (↑)	Rot L2 (↓)	Transl L2 (↓)	#Stitch Acc (↑)
Sewfactory	SewFormer	3.3	89.8	99.3	.008	0.8	99.2
	AIpparel	**2.8**	**93.9**	**99.9**	**.005**	**0.6**	**99.8**
GCD-MM	SewFormer-FT	12.3	79.4	44.7	.040	4.5	2.8
	AIpparel	**5.4**	**85.2**	**82.7**	**.020**	**2.7**	**77.2**

Table 2. **Image-to-Garment Prediction (Quantitative).** AIpparel achieves state-of-the-art performance in both datasets and surpasses SewFormer-FT by a large margin on GCD-MM.

transformer. We denote the i-th embedding in $\boldsymbol{X}$ as $\boldsymbol{X}_i$, and $\boldsymbol{X}_{<i}, \boldsymbol{X}_{>i}$ are sliced sequences before or after the i-the embedding, respectively. We use f_ϕ to denote the language transformer from LLaVA. We use $\boldsymbol{X}$ to denote tokens before passing through f_ϕ and $\boldsymbol{H} = f_\phi(\boldsymbol{X})$ as the output hidden features from the transformer.

Continuous Parameters. The tokenization scheme in Eq. 1 does not include any continuous parameters such as vertex positions, control points for edges, or rigid transformation of panels. Prior works represent continuous parameters as quantized tokens in discrete space [21, 47]. This introduces quantization error for the continuous parameters and uses more tokens per panel, leading to a longer, inefficient representation. Inspired by recent approaches of extending LMMs [19, 32], we propose using small regression heads to map hidden features of the transformer to the continuous parameters. Specifically, we define an MLP $g_\theta^{(\mathrm{e})} : \mathbb{R}^D \to \mathbb{R}^C$ to map LLaVA's hidden features from the last layer to vertices and control points. As illustrated in Fig. 2, $g_\theta^{(\mathrm{e})}$ takes the output embedding corresponding to the token right before the edge type token. Concretely, if the i-th token, $\boldsymbol{X}_i$, corresponds to an edge-type token for edge e, its associated output embedding $\boldsymbol{H}_{<i} = f_\phi(\boldsymbol{X}_{<i}) \in \mathbb{R}^{(i-1)\times D}$ is used to predict e's endpoint and control parameters via $g_\theta^{(\mathrm{e})}(\boldsymbol{H}_{i-1})$. Similarly, we also define a transformation regression head $g_{\theta'}^{(R)} : \mathbb{R}^D \to \mathbb{R}^7$ mapping the hidden features of <R>'s previous token to a translation $T \in \mathbb{R}^3$ and a rotation quaternion $q \in \mathbb{H}$. In this way, continuous parameters are regressed using small regression heads that are jointly trained with the transformer, leading to more efficient context length usage in representing sewing patterns. At training time, we use ground-truth parameters for supervision, and during generation, we use the last hidden feature from the output for regression should an edge-type token or a transformation token be sampled.

Positional Embeddings. To include information on continuous parameters in the sewing pattern tokenization defined in Eq. 1, we include the second endpoint of each edge as a positional embedding added to the token embedding. Specifically, we define $h_\varphi^{(e)} : \mathbb{R}^2 \to \mathbb{R}^D$ as a two-layer perceptron. For each edge $e = (v_1, v_2)$ with an edge-type token embedding of $\boldsymbol{X}_i$, we add $h_\varphi^{(e)}(v_2)$ to $\boldsymbol{X}_i$ to inform the language model f_ϕ of the vertex positions. We also define

$h_{\varphi'}^{(R)} : \mathbb{R}^7 \rightarrow \mathbb{R}^D$ to be the projection function for transformations. The transformation parameters $(t, q) \in \mathbb{R}^7$ for each panel are projected using $h_{\varphi}^{(R)}$ and added to the token embedding for <R>. At training time, we use ground-truth vertex and 3D transformations as positional embeddings, and during generation, we use the parameters predicted by the regression heads.

Training. We keep both the vision encoder and projection frozen, and fine-tune all weights in the language model f_ϕ, regression heads $g_\theta^{(\mathrm{e})}, g_{\theta'}^{(R)}$, and the positional embedding projection layers $h_{\varphi}^{(\mathrm{e})}, h_{\varphi'}^{(R)}$. The fine-tuning loss is defined as a combination of cross-entropy (CE) on the discrete tokens and L_2 loss on the continuous parameters:

$$\begin{aligned} \mathcal{L} = &\, \mathrm{CE}(f_\phi(\boldsymbol{X}_{<-1}), \boldsymbol{X}_{1>}) \\ &+ \lambda \sum_{e'} \left\| g_\theta^{(\mathrm{e})} \circ f_\phi\left(\boldsymbol{X}_{<i_{e'}}\right) - v_2^{e'} \right\|_2 \\ &+ \lambda \sum_{R'} \left\| g_{\theta'}^{(R)} \circ f_\phi\left(\boldsymbol{X}_{<i_{R'}}\right) - R' \right\|_2 . \end{aligned} \tag{2}$$

Here, $\sum_{e'}$ is the sum over all edges in the sewing pattern's sequence $\boldsymbol{X}$. For each edge e', its second endpoint is denoted as $v_2^{e'}$. Similarly, $\sum_{R'}$ is the sum over all the transformations. We do not explicitly include the positional embedding in Eq. 2, but it is added according to the rules defined in the previous paragraph.

4. Experiments

We validate the effectiveness of our model on multiple tasks and conduct an ablation study on the key technical designs.

Training Details. We train AIpparel on GCD-MM multimodal data samples for image-to-garment and text-to-garment generation, as well as text-based garment editing. We randomly split GCD-MM into train-validation-test subsets with a 90:5:5 ratio. All of our results on GCD-MM below are predicted using a single model. See the supplementary for a complete implementation and training setup.

Metrics. To quantitatively measure our sewing pattern predictions, we use reconstruction metrics established by previous works [29, 38]. Given a pair of generated and ground-truth sewing patterns, we measure 1) *Panel L2, Rot L2, and Transl L2*: average vertex, rotation and translation L2 distance between predicted and ground-truth panels; 2) *#Panel Accuracy*: percentage of sewing patterns with correctly predicted number of panels; 3) *#Edge Accuracy*: percentage of correctly predicted edges in each correctly predicted panel; 4) *#Stitch Accuracy*: accuracy of predicted stitches compared to ground truth. To save space in compact tables, we report *Accuracy*, the product of *#Panel Accuracy* and *#Edge Accuracy*, to provide a comprehensive measurement of garment reconstruction quality. All L2-based metrics are measured in centimeters except for rotation.

4.1. Image to Sewing Pattern Prediction

We test our model's capability to reconstruct garments from a single image using two datasets: SewFactory [38] and GCD-MM. For the baseline, we compare with SewFormer's pre-trained model on the SewFactory dataset. Because SewFormer did not release their train–test split for their pre-trained model, we use a custom split for a fair comparison. For GCD-MM, we fine-tune SewFormer until its validation loss no longer improves. We denote it as *SewFormer-FT*.

Tab. 2 shows quantitative comparisons on the two datasets. AIpparel outperforms the baselines on both datasets, suggesting that our method outputs more accurate sewing patterns than the baseline. In particular, AIpparel shows a large performance improvement over SewFormer-FT on the difficult GCD-MM dataset, indicating the effectiveness of our method in predicting more complex sewing patterns. Fig. 3 shows qualitative results. The two examples on the left show that SewFormer-FT fails to reconstruct simulatable garments despite fine-tuning. This suggests that SewFormer cannot adapt to complex garments with small panels and diverse edge types. In contrast, our model predicts sewing patterns matching the input images, including small panels such as the waistband on the top row and the sleeve cuffs at the bottom. The two examples on the right show results on SewFactory [38]. The pre-trained SewFormer fails to predict the garment in the top row with sleeves and the bottom row's skirt as a pair of pants, while AIpparel correctly predicts the sewing patterns based on the inputs.

4.2. Multimodal Garment Generation

We evaluate the effectiveness of AIpparel in various multimodal garment generation scenarios. Specifically, we assess its performance on a set of 100 garments with 5 types of multimodal inputs (20 test samples each): 1) texts, 2) images, 3) a combination of text and images, 4) open-ended prompts that require reasoning, and 5) editing instructions. Success in such a benchmark requires the model to make accurate predictions conditioned on a variety of different multimodal input formats, as well as having an understanding of common-sense knowledge. Refer to Fig. 4 for generation examples in these tasks. We abbreviate the text input for compactness. Refer to the supplementary for complete examples.

Because no existing work handles multimodal sewing pattern generation, we adopt state-of-the-art (SOTA) single-modal generative methods, i.e., SewFormer-FT and DressCode, to perform multimodal tasks. For this purpose, we augment these baselines using multimodal models, e.g., GPT-4o [74] and DALL-E [8], to translate the multimodal inputs to their input domains (i.e., images and short keyword description). To ensure translation accuracy, we manually inspect the results before querying SewFormer-FT and DressCode. We denote them as *Sewformer-FT*† and *Dress-*

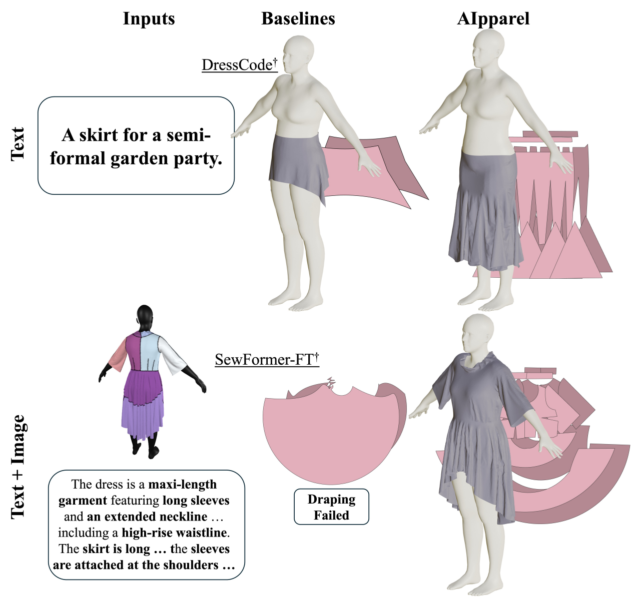


Figure 4. **Multimodal Sewing Pattern Prediction (Qualitative).** AIpparel accurately predicts sewing patterns that follows the inputs better than the baselines. See Sec. 4.2.

Method	Accuracy (↑)	Panel L2 (↓)
SewFormer-FT†	10.3	22.4
DressCode†	0.6	31.0
AIpparel	**59.0**	**6.1**

Table 3. **Multimodal Sewing Pattern Prediction.** Compared to single-modal methods augmented with existing LMMs, our model outperforms both baselines by a large margin.

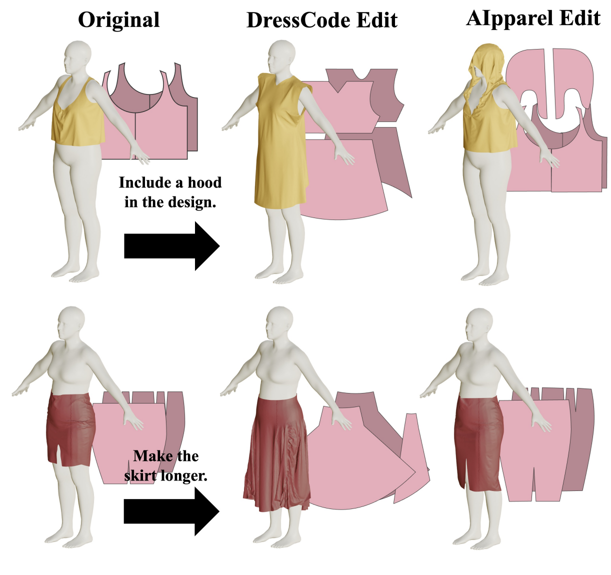


Figure 5. **Sewing Pattern Editing (Qualitative).** Our model follows the editing instructions more accurately compared with the baseline by accurately including a hood to the tank top (top row) and elongating the skirt (bottom row). See Sec. 4.3.

Method	Accuracy (↑)	Panel L2 (↓)
Sewformer*	9.5	18.6
DressCode*	37.0	13.7
AIpparel	**83.4**	**1.5**

Table 4. **Sewing Pattern Editing.** We use SOTA LMMs such as GPT-4V and DALL-E to facilitate both baselines to perform this multimodal editing task. Our model still outperforms both baselines by a large margin.

Code†, respectively. In comparison, our model can directly perform all five categories of multimodal tasks without relying on external modules.

We report quantitative comparisons in Tab. 3, measured between the reference and predicted sewing patterns. Compared with single-modal baselines, AIpparel performs significantly better. Fig. 4 shows qualitative comparisons. The first row validates the method's reasoning ability by asking for a suitable sewing pattern for a specific occasion (e.g., a "semi-formal garden party"). We use DressCode† as our baseline. Notice that DressCode† generates a mini-skirt that does not match well the description of a "semi-formal garden party". Our model outputs a godet skirt that is more appropriate for this occasion. The second row shows an example of sewing pattern generation given a combination of image and text. SewFormer-FT† fails to generate a plausible sewing pattern due to the garment's complexity, whereas AIpparel reconstructs the complex garment closely following both visual and textual cues, such as the sleeve and skirt length in the image, and the waistline and neckline descriptions in the text.

4.3. Language-driven Sewing Pattern Editing

We validate AIpparel's ability to perform sewing pattern editing. Given a sewing pattern and text-based editing instructions, the model is tasked with editing the pattern according to the prompt without altering the overall style of the garment. Since the existing sewing pattern generation models, DressCode and SewFormer, are not designed for this task, we adapt them for editing using a pre-trained InstructPix2Pix [12] and GPT4o [74][2]. We denote them as *DressCode** and *SewFormer**, respectively.

We report quantitative comparisons in Tab. 4. Our model outperforms the baselines in both metrics by a large margin, indicating that AIpparel performs more accurate edits than the baselines while minimally affecting the rest of the sewing patterns. Qualitative results are shown in Fig. 5. DressCode produces results visibly deviating from the input garment. For example, DressCode changes the tank top to a full-length dress in the top row and the tight skirt to

[2]See supplementary for details

Methods	Accuracy (↑)	Panel L2 (↓)	Time (↓)
DressCode	38.4	22.4	52.2s
Ours w.o. reg.	79.0	7.2	3.4s
Ours	**85.0**	**6.1**	**2.1s**

Table 5. **Ablation.** Our tokenizer outperforms DressCode in all metrics while being more than 25 times faster at inference time. Our objective (Eq. 2) also improves performance compared to the cross-entropy-only variant.

a flared one in the bottom row. These mistakes arise because DressCode requires external modules to translate the sewing pattern to short keywords for input, losing important information about the original garment style during the process. In contrast, AIpparel directly accepts the sewing pattern and textual instructions as input, allowing it to accurately perform the minimal edits required to modify the garment according to the instructions, as demonstrated in both examples.

4.4. Ablation Study

We validate our key technical contributions in an ablation study. Specifically, we compare our tokenizer described in Sec. 3 with the existing tokenization scheme from DressCode [78]. We also perform an ablation study on our proposed mixed fine-tuning objective in Eq. 2, comparing it with a cross-entropy-only objective ("*Ours w.o. reg*"). We use text-to-garment prediction on DressCode's dataset as our ablation task to compare with DressCode's pre-trained model. For a fair comparison, we use the same backbone as DressCode and only change the tokenizer and training objectives. To implement "*Ours w.o. reg.*", we quantize vertex positions, edge control parameters, and 3D transformations into 256 bins, which are then predicted using next token prediction. See the supplementary for implementation details.

Tab. 5 shows the ablation results compared to our full model in text-to-garment tasks. The results are averaged over 100 samples from DressCode's test set. Compared to DressCode, our tokenizer, both with and without regression loss, significantly improves the reconstruction fidelity, as demonstrated by the large improvement in the metric values. Furthermore, by using a mixed training objective in Eq. 2, the reconstruction quality of sewing patterns improves significantly, demonstrating the effectiveness of our objective. In addition to quality improvements, our tokenization drastically accelerates generation (25× speedup) compared to DressCode, as shown in the same table. The reported times show the average wall-clock time required to generate and decode a single garment in seconds, measured for each method on a single A6000 GPU. Fig. 6 displays reconstructed sewing patterns from all three methods. Notice that DressCode's prediction does not accurately reflect the language description (i.e., the top row's skirt is not flared) and shows geometry artifacts (bottom row, boxed region). Meanwhile, our proposed tokenizer and training objective predict garments with the best visual quality and alignment with the textual description.

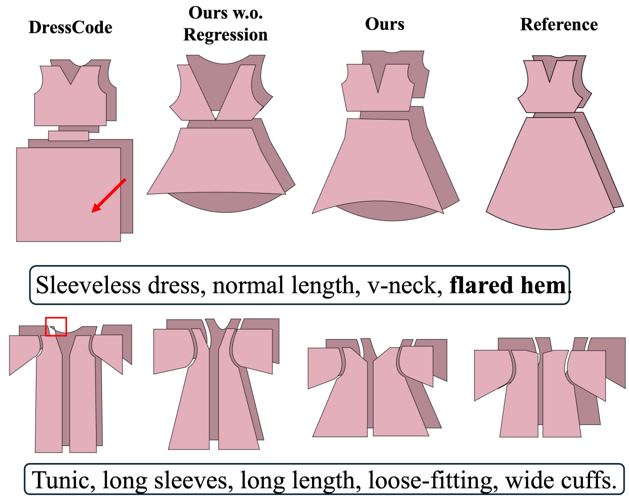


Figure 6. **Ablation (Qualitative).** DressCode's tokenizer produces unrealistic patterns (second row, boxed region) and does not match the text input (i.e., "flared hem"). In contrast, our tokenizer outputs geometrically regular sewing patterns accurately aligning with the inputs. See Sec. 4.4.

5. Discussion

We introduce AIpparel, a 7B-parameter multimodal foundation model for garment sewing patterns. To train AIpparel, we curate GCD-MM, a large-scale dataset with complex sewing patterns and multimodal annotations. Moreover, we develop a novel sewing pattern tokenizer and a mixed training objective for fine-tuning LMMs on GCD-MM. AIpparel achieves state-of-the-art results on single-modal and multimodal sewing-pattern-generation tasks, enabling new applications like language-driven sewing pattern editing.

Limitations and Future Work. While the current representation enables the digitalization of complex sewing patterns, it is still constrained to garments representable by manifold surfaces. Design elements like pockets require non-manifold structures. A promising direction is to develop an efficient representation that accurately models non-manifold features while remaining compatible with LMMs. Fabricating the generated garments is another interesting direction, which requires consideration of physical and material constraints during sewing pattern prediction.

Broader Impacts. While we believe our model can advance AI-assisted fashion design, we acknowledge potential risks we inherit from the pre-trained LLaVA model. For instance, generative AIs can spread misinformation or create biases potentially harmful to society. We do not condone these and other improper usage of our model.

Conclusion. Vision-language and other large multimodal models capture web knowledge and enable reasoning for many downstream applications. By fine-tuning LMMs to understand sewing patterns, we take first steps towards a vision-language-garment model that transfers web knowledge to garment generation and editing, unlocking a plethora of applications for fashion design and fabrication.

6. Acknowledgements

The project is supported by Google, an ERC Consolidator Grant No. 101003104 (MYCLOTH), an ARL grant W911NF-21-2-0104, a Vannevar Bush Faculty Fellowship, and LVMH.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Anthropic. Introducing claude 3.5 sonnet, 2024. Accessed: 2024-09-06. 2

[3] Dimitrije Antić, Garvita Tiwari, Batuhan Ozcomlekci, Riccardo Marin, and Gerard Pons-Moll. Close: A 3d clothing segmentation dataset and model. In *2024 International Conference on 3D Vision (3DV)*, pages 591–601. IEEE, 2024. 2

[4] Helen Joseph Armstrong. *Patternmaking for Fashion Design*. Pearson India, 2009. 1

[5] Alberto Baldrati, Davide Morelli, Giuseppe Cartella, Marcella Cornia, Marco Bertini, and Rita Cucchiara. Multimodal garment designer: Human-centric latent diffusion models for fashion image editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23393–23402, 2023. 2

[6] Seungbae Bang, Maria Korosteleva, and Sung-Hee Lee. Estimating garment patterns from static scan data. In *Computer Graphics Forum*, pages 273–287. Wiley Online Library, 2021. 1

[7] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020. 3

[8] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 3, 6

[9] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5420–5430, 2019. 2

[10] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023. 2

[11] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023. 2

[12] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 7, 23

[13] Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation, 2020. 4

[14] Xiaowu Chen, Bin Zhou, Feixiang Lu, Lin Wang, Lang Bi, and Ping Tan. Garment modeling with a depth camera. *ACM Trans. Graph.*, 34(6), 2015. 1, 2

[15] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2

[16] Hein Daanen and Sung-Ae Hong. Made-to-measure pattern development based on 3d whole body scans. *International Journal of Clothing Science and Technology*, 20:15–25, 2008. 2

[17] Luca De Luigi, Ren Li, Benoit Guillard, Mathieu Salzmann, and Pascal Fua. Drapenet: Garment generation and self-supervised draping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1451–1460, 2023. 2

[18] Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani. Virtual garments: A fully geometric approach for clothing design. *Computer Graphics Forum*, 25(3):625–634, 2006. 2

[19] Yao Feng, Jing Lin, Sai Kumar Dwivedi, Yu Sun, Priyanka Patel, and Michael J. Black. Chatpose: Chatting about 3d human pose. In *CVPR*, 2024. 2, 5, 25

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

[21] Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. Dresscode: Autoregressively sewing and generating garments from text guidance, 2024. 1, 2, 3, 4, 5, 14, 17, 18, 19, 23

[22] Hsuan-I Ho, Lixin Xue, Jie Song, and Otmar Hilliges. Learning locally editable virtual humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21024–21035, 2023. 2

[23] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. 24

[24] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, Yi Ren, Zhou Zhao, and Shinji Watanabe. Audiogpt: Understanding and generating speech, music, sound, and talking head, 2023. 2

[25] Moon-Hwan Jeong, Dong-Hoon Han, and Hyeong-Seok Ko. Garment capture from a photograph. *Comput. Animat. Virtual Worlds*, 26(3–4):291–300, 2015. 2

[26] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 18–35. Springer, 2020. 2, 3

[27] Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2human: Text-driven controllable human image generation. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022. 2

[28] Maria Korosteleva and Sung-Hee Lee. Generating datasets of 3d garments with sewing patterns. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 3, 17

[29] Maria Korosteleva and Sung-Hee Lee. Neuraltailor: Reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Trans. Graph.*, 41(4), 2022. 1, 2, 4, 6, 19

[30] Maria Korosteleva and Olga Sorkine-Hornung. Garmentcode: Programming parametric sewing patterns. *ACM Transactions on Graphics (TOG)*, 42(6):1–15, 2023. 3, 14, 16, 17, 25

[31] 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. 2, 3, 17, 27

[32] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 2, 5, 25

[33] 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. *arXiv preprint arXiv:2405.12420*, 2024. 2

[34] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2

[35] Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. Codimensional incremental potential contact. *ACM Trans. Graph. (SIGGRAPH)*, 40(4), 2021. 25

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 2, 3, 18

[37] Kaixuan Liu, Xianyi Zeng, Pascal Bruniaux, Xuyuan Tao, Xiaofeng Yao, Victoria Li, and Jianping Wang. 3d interactive garment pattern-making technology. *Computer-Aided Design*, 104:113–124, 2018. 2

[38] Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2023. 2, 3, 4, 6, 14, 17, 18, 19

[39] Zhaoyang Liu, Yinan He, Wenhai Wang, Weiyun Wang, Yi Wang, Shoufa Chen, Qinglong Zhang, Zeqiang Lai, Yang Yang, Qingyun Li, Jiashuo Yu, et al. Interngpt: Solving vision-centric tasks by interacting with chatgpt beyond language. *arXiv preprint arXiv:2305.05662*, 2023. 2

[40] Zhen Luo, Tianxing Li, and Takashi Kanai. Garmatnet: A learning-based method for predicting 3d garment mesh with parameterized materials. In *Proceedings of the 14th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–10, 2021. 2

[41] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 2

[42] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. `https://github.com/nvidia/warp`, 2022. NVIDIA GPU Technology Conference (GTC). 25

[43] Yuwei Meng, Charlie C.L. Wang, and Xiaogang Jin. Flexible shape control for automatic resizing of apparel products. *Computer-Aided Design*, 44(1):68–76, 2012. Digital Human Modeling in Product Design. 2

[44] 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. 2

[45] Rahul Narain, Armin Samii, and James F O'brien. Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG)*, 31(6):1–10, 2012. 3

[46] Fumiya Narita, Shunsuke Saito, Takuya Kato, Tsukasa Fukusato, and Shigeo Morishima. Pose-independent garment transfer. 2014. 1

[47] Charlie Nash, Yaroslav Ganin, S. M. Ali Eslami, and Peter W. Battaglia. Polygen: An autoregressive generative model of 3d meshes, 2020. 5

[48] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo

Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. 2

[49] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7365–7375, 2020. 2

[50] Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa A Vidal-Calleja, and Olga Sorkine-Hornung. Computational pattern making from 3d garment models. *ACM Trans. Graph.*, 41(4):157–1, 2022. 1, 2

[51] Albert Pumarola, Jordi Sanchez-Riera, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3dpeople: Modeling the geometry of dressed humans. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2242–2251, 2019. 3

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 18

[53] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 3505–3506, New York, NY, USA, 2020. Association for Computing Machinery. 19

[54] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019. 2

[55] Nicholas Sharp and Keenan Crane. Variational surface cutting. *ACM Trans. Graph.*, 37(4), 2018. 2

[56] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: solving ai tasks with chatgpt and its friends in hugging face. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 2

[57] 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. 2

[58] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023. 2

[59] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 1–18. Springer, 2020. 2

[60] Lokender Tiwari and Brojeshwar Bhowmick. Deepdraper: Fast and accurate 3d garment draping over a 3d human body. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1416–1426, 2021. 2

[61] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1:

A fully open, vision-centric exploration of multimodal llms, 2024. 2

[62] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 2, 18

[63] Raquel Vidaurre, Igor Santesteban, Elena Garces, and Dan Casas. Fully convolutional graph neural networks for parametric virtual try-on. In *Computer Graphics Forum*, pages 145–156. Wiley Online Library, 2020. 3

[64] Charlie C.L. Wang, Yu Wang, and Matthew M.F. Yuen. Feature based 3d garment design through 2d sketches. *Computer-Aided Design*, 35(7):659–672, 2003. 2

[65] Charlie C. L. Wang, Yu Wang, and Matthew Ming-Fai Yuen. Design automation for customized apparel products. *Comput. Aided Des.*, 37:675–691, 2005. 2

[66] Huamin Wang, James F O'Brien, and Ravi Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011. 3

[67] Jin Wang, Guodong Lu, Weilong Li, Long Chen, and Yoshiyuki Sakaguti. Interactive 3d garment design with constrained contour curves and style curves. *Comput. Aided Des.*, 41(9):614–625, 2009. 2

[68] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popović, and Niloy Jyoti Mitra. Learning a shared shape space for multimodal garment design. *ACM Transactions on Graphics (TOG)*, 37:1 – 13, 2018. 2, 3

[69] Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, and Jifeng Dai. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks, 2023. 2

[70] Katja Wolff, Philipp Herholz, Verena Ziegler, Frauke Link, Nico Brügel, and Olga Sorkine-Hornung. 3d custom fit garment design with body movement. *arXiv preprint arXiv:2102.05462*, pages 1–12, 2021. 1

[71] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models, 2023. 2

[72] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023. 2

[73] Wenqiang Xu, Wenxin Du, Han Xue, Yutong Li, Ruolin Ye, Yan-Feng Wang, and Cewu Lu. Clothpose: A real-world benchmark for visual analysis of garment pose via an indirect recording solution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 58–68, 2023. 2

[74] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. Gpt4tools: Teaching llm to use tools via self-instruction, 2023. 2, 3, 6, 7

[75] Shan Yang, Tanya Ambert, Zherong Pan, Ke Wang, Licheng Yu, Tamara Berg, and Ming C. Lin. Detailed garment recovery from a single-view image, 2016. 1

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

[77] Zhengyuan Yang*, Linjie Li*, Jianfeng Wang*, Kevin Lin*, Ehsan Azarnasab*, Faisal Ahmed*, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. 2023. 2

[78] Hsiao yu Chen, Egor Larionov, Ladislav Kavan, Gene Lin, Doug Roble, Olga Sorkine-Hornung, and Tuur Stuyck. Dress anyone : Automatic physically-based garment pattern refitting, 2024. 8

[79] Yang Yunchu and Zhang Weiyuan. Prototype garment pattern flattening based on individual 3d virtual dummy. *International Journal of Clothing Science and Technology*, 19(5): 334–348, 2007. 1

[80] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities, 2023. 2

[81] 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*, pages 1748–1758, 2024. 2

[82] Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, and Gordon Wetzstein. Physavatar: Learning the physics of dressed 3d avatars from visual observations. 2024. 25

[83] Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, et al. Physavatar: Learning the physics of dressed 3d avatars from visual observations. *arXiv preprint arXiv:2404.04421*, 2024. 2

[84] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2

[85] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 512–530. Springer, 2020. 2

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

[87] Xingxing Zou, Xintong Han, and Waikeung Wong. Cloth4d: A dataset for clothed human reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12847–12857, 2023. 2

Supplementary Materials

Contents

S1. Details on GarmentCodeData-Multimodal (GCD-MM)

We expand on the data curation process of GCD-MM. Specifically, in Sec. S1.1, we detail the specifics of generating text descriptions for the sewing patterns. In Sec. S1.2, we elaborate on how we generate the edited sewing patterns and their associated editing instructions. Lastly, in Sec. S1.3, we show statistics comparing sewing patterns in GCD-MM and sewing patterns in previous datasets used by DressCode [21] and SewFormer [38].

S1.1. Text Description Generation

We generate two types of sewing pattern descriptions for each garment in GCD-MM. The first is a detailed natural language description of the sewing pattern, while the second outlines a suitable occasion for wearing the garment. For captioning purposes, we use a standardized body type corresponding to the mean shape and pose derived from SMPL.

Obtaining pattern descriptions happens in two steps. First, we generate keywords describing the simulated garments using the design parametrization of each garment. Generated based on GarmentCode [30], each garment is characterized by a set of continuous and categorical parameters. We generate descriptions for each garment using the following rules:

- **Categorical parameters**: We assign the categorical label when appropriate. For instance, a `godet skirt` is classified as such. Some categorical parameters do not suffice - a `shirt` can signify anything from a crop top to a dress. For these instances, we add additional checks consulting additional parameters.
- **Continuous parameters**: We define thresholds and assign different qualitative labels for garments above and below them. Parameters such as `sleeve length` or `collar width` are obvious examples.
- **Dependent parameters**: Most parameters have no impact on the final garment, as they only become relevant when certain categorical parameters are set. We design rules that consider these edge cases. Only when a `godet skirt` is set, does the `num inserts` become relevant. We include all relevant dependent parameters that have a structural effect on the garment.

Similar to DressCode, we first generate a garment type description and a collection of keywords that contain the specific description based on our rule-based approach. Note that each rule can contribute several keywords. See Figure S1 for the examples.

In the second stage, we use these generated keywords in combination with a render of the front and back of the garment to prompt GPT-4o. We construct the prompt such that GPT-4o objectively describe the garment using the characteristic features of the garment provided by the generated keywords and renders. In addition, we include instructions to focus on information crucial for our learning problems, such as panel connectivity and stitching patterns, while ignoring irrelevant information, such as colors or interpretations.

The following is the system prompt that we used:

DressCode: jacket; short sleeves; with a hood; fitted garment
Ours: An upper-body garment; both sleeves; short sleeves; with lapels

DressCode: trousers; long length; wide fit; front slit; high waist
Ours: A maxi skirt; narrow waistband; skirt with front slit; skirt with back slit; skirt with side slit

Figure S1. **Comparison between our Short Captions and DressCodes'.** This figure shows the short captions created by DressCode and our method for two different garments. DressCode produces keywords that do not align with the garment (red).

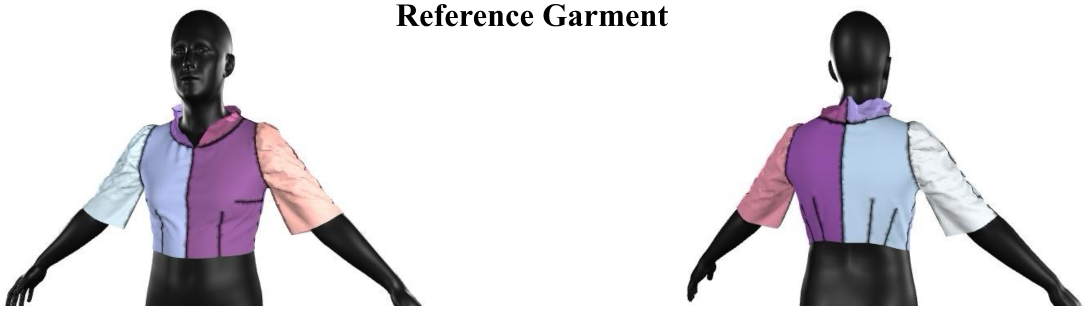
You are a fashion expert tasked with providing concise and neutral descriptions of garments based on the provided textual information. Your descriptions should focus on specific stitching details and how different panels are connected (such as seam placements and stitching patterns), as well as any distinctive characteristics and design elements of the garment. When describing the garment's appearance, use precise and concrete language, avoiding generic phrases or broad descriptions. Do not mention that seams are visible; instead, describe where seams or panels are located to indicate construction details. Do not include any impressions, subjective interpretations, or unobservable aspects. Avoid mentioning colors or any references to images. Keep the descriptions brief and to the point, avoiding unnecessary words. Use only the information provided.

Here we present the user prompt:

Please generate a concise and neutral description of a garment, focusing on specific stitching details, how different panels are connected, and including any distinctive characteristics and design elements, based on the following information:
- **Title**: {title}
- **Description**: {description}
Provide a brief description that emphasizes stitching and construction details (such as seam placements, panel connections, and stitching patterns), along with precise visual observations about the garment's appearance, including style, silhouette, length, and any unique design features or distinctive characteristics. Avoid using generic phrases or broad descriptions; instead, provide specific details about the garment's features. Do not mention that seams are visible; instead, describe where seams or panels are located to indicate construction details. Do not include any impressions, subjective interpretations, or unobservable aspects. Avoid mentioning colors or any references to images. Keep the description succinct and avoid unnecessary words. Use only the information provided.

The second type of caption describes an occasion for which a garment is suitable. In this prompt, we ask the model not to pay attention to the garment's colors which only highlight different panels and are not semantically relevant. Instead, we ask it to focus on the shape and description. We use the same information as before to prompt GPT-4o. This is the system prompt:

Reference Garment



Ours: This upper-body garment features **short sleeves and lapels**, indicating a structured yet casual design…

DressCode: This **jacket** features a fitted silhouette with short sleeves and an **attached hood**…

Figure S2. **Captions generated using GPT-4V.**

> You are an expert in fashion design and garment analysis. When provided with images of garments and their metadata, focus solely on their shape and stitching. Note that different colors in the images represent different panels of the garment and are not indicative of style or color choices. Ignore colors and any visible seams meant only for stitching information. The metadata includes a title and a description, which is a list of short attributes; use these to inform your understanding. Based on this information, provide only a detailed, but concise, description of a single occasion where the given garment would be appropriate to wear. Do not include any other information in your response.

and here is the user prompt:

> Given the following garment's metadata and images (remember that colors and seams are only for panel representation and stitching information), please provide only a detailed, but concise, description of a single occasion where this garment can be worn. Do not include any other information in your response.
> Here is the metadata:
> Title: {title}
> Description: {description}

Effect of GPT version in caption quality. While GPT-4o potentially increases the accuracy of generated captions, the in-context knowledge about various design parameters crucially helps the model to generate captions more faithful to the garment design. Fig. S2 shows the same captions in Fig. S1 of supp, generated instead using GPT-4V. Notice that DressCode's caption contains severe flaws (in red) due to inaccurate in-context prompting. Ours do not have these flaws because we prompt GPT-4V with design-parameter-inspired content. We will update Fig.S1 to include this example in the revision.

S1.2. Generation of Editing Data Sample

To generate paired garments representing before-and-after edits, we use design parameters from the GCD dataset and systematically apply one of five pre-defined transformation rules. The modified design parameters are then converted into garments using GarmentCode [30].

Each garment from GCD is first evaluated to determine which transformation rules are applicable. One rule is then randomly selected and applied. Due to limitations in GarmentCode's design space, not all edited design parameters can be converted into sewing patterns. As a result, GCD-MM comprises 120k garment pairs that are successfully generated from the 130k garments in GCD, while approximately 10k garments remain unpaired.

The transformation rules include adjustments to garment lengths (sleeves, pants, skirts), collar type changes, modifications to garment symmetry, toggling the presence of hoods, and structural edits to style elements (e.g., changing the number of inserts in godet skirts). Each rule takes the existing design parameters as input and applies a targeted change. For instance,

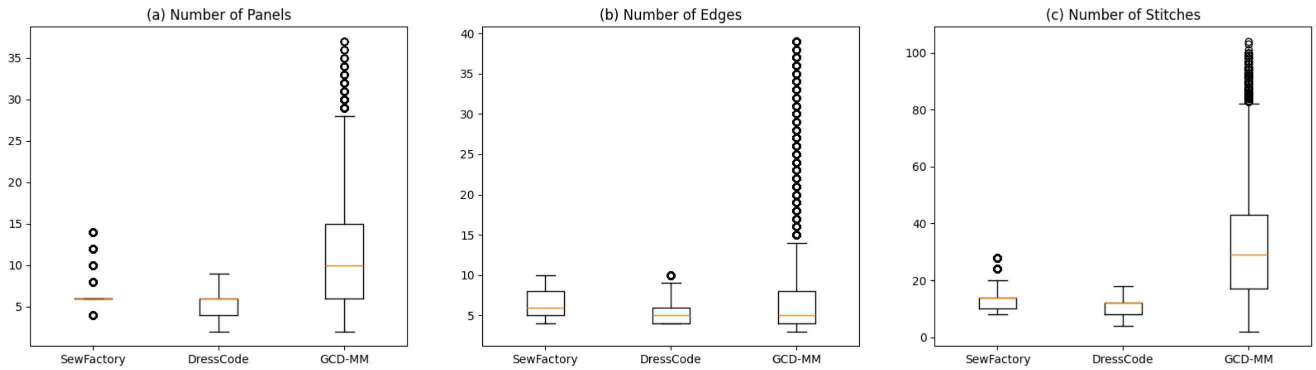


Figure S3. **Dataset Statistics Comparisons.** Notice that GCD-MM in general contains larger variations in the number of panels, edges, and stitches in the sewing patterns. This poses additional challenges in designing a sewing pattern generation method with GCD-MM.

	SewFactory [38]	DressCode's Dataset [21, 28]	**GCD-MM**
Edge Types	L, QB	L, QB	L, QB, CB, A
Number of Sewing Patterns	13700	20292	127629

Table S1. **Dataset Statistics Comparison.** L=Line, QB=Quadratic Beziér, CB=Cubic Beziér, A=Arc. GCD-MM shows a larger variation in both numbers of panels, edges, and stitches than previous sewing pattern datasets. For Panel, edge, and stitching statistics, refer to Figure S3.

length adjustments alter sleeves, pants, or skirts by 50% of their initial length, constrained by the maximum length specified in GarmentCode. Similarly, collar types are randomly reassigned from a predefined set, garment symmetry is toggled, and hoods are added or removed.

These rules are designed for three key reasons: (1) they produce clear and concise edits that can be succinctly summarized; (2) they encompass varying levels of editing complexity, from minor panel length adjustments to major structural modifications involving new panels and altered stitching; and (3) for all garments in the dataset, at least one rule can always be applied.

To document each transformation, we generate descriptive sentences for the edited garments using a rule-based approach. Here are a set of examples:

Godet skirt:	*"Increase the number of inserts in the skirt by $x."*
Pants:	*"Make the pants longer."*
Shirt:	*"Switch the collar type from $currCollar to $newCollar."*

In total, the defined rules enable 52 distinct, describable modifications, ensuring a diverse and well-documented dataset of garment editing pairs.

S1.3. Sewing Pattern Statistics

GCD-MM uses sewing patterns fitted on a default body from the GarmentCodeData (GCD) dataset [31], which are procedurally generated sewing patterns using the programming abstraction of GarmentCode [30]. Compared with the sewing patterns used by SewFormer [38] and DressCode [21], GCD contains more complicated and diverse sewing patterns. For detailed documentation and comparison with existing datasets and procedural sewing pattern generators, please refer to GarmentCodeData [31]. Here, we briefly show some statistics comparing these different datasets.

GCD exhibits more diverse and detailed garment feature variations than the previous dataset, including fitted garments, correct sleeve shapes, more collar types, more skirt types, cuffs, and asymmetric features (tops, asymmetric skirt cuts). All of these characteristics make sewing patterns from GCD more complicated than existing sewing pattern datasets.

Comparatively, datasets used by SewFormer [38] and DressCode [21] are procedurally generated sewing patterns from an older programming abstraction [28]. While this programming abstraction can also generate sewing patterns for the types of garments described above, all its variations are from changes in the vertex and control point positions while fixing the number of panels, edges, and stitches the same. This constraint significantly limits the variations exhibited in the datasets used by

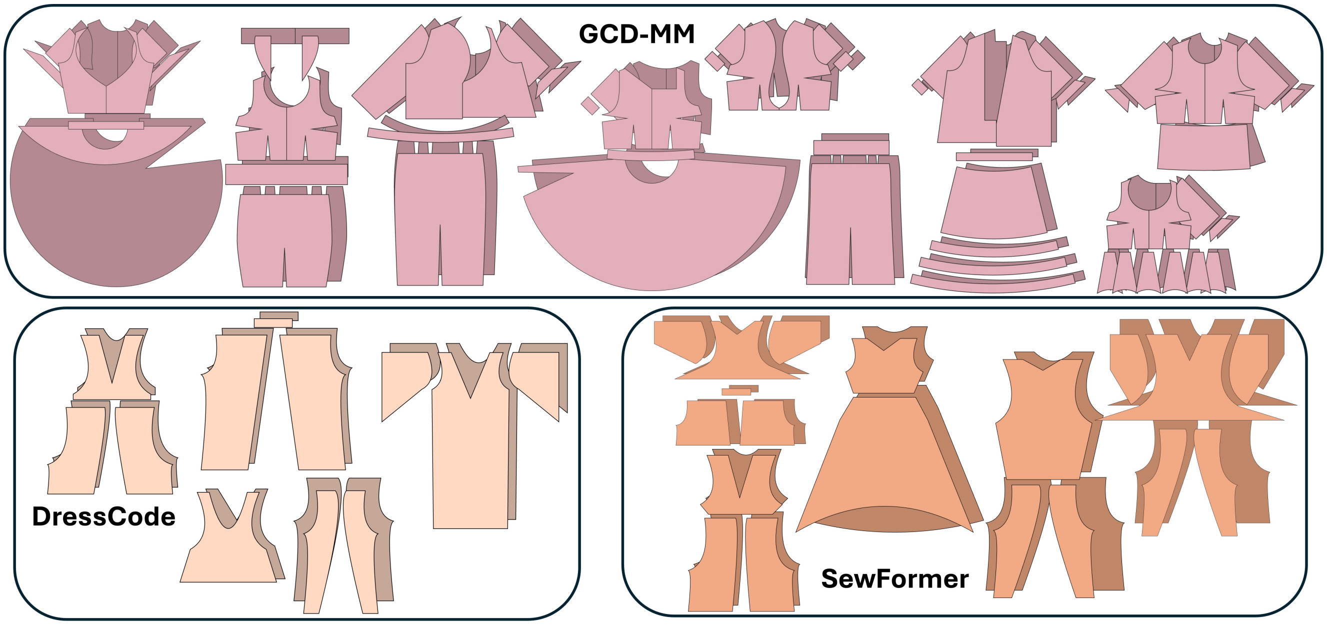


Figure S4. **Visualization of Sewing Patterns**. Random sewing pattern samples from the datasets used by AIpparel and the baselines are visualized. Notice that compared to prior works, GCD-MM exhibits more complex sewing patterns in general.

SewFormer and DressCode. Figure S4 showcases randomly sampled sewing patterns as well as their draped renderings from GCD and sewing patterns used by SewFormer and DressCode. We see that sewing patterns from GCD are generally more complex and diverse than the previous dataset. Table S1 and Figure S3 show a statistical comparison in terms of the number of edges, panels, stitches, and edge types between sewing patterns in GCD-MM, SewFactory [38], and dataset used by DressCode [21]. Notice that comparatively sewing patterns in GCD-MM exhibit the largest variation in all of the statistics, demonstrating the difficulty of the dataset. In particular, because of this difficulty gap, previous methods such as SewFormer and DressCode exhibit poor performance despite fine-tuning their network on GCD-MM. See Section S3 for details.

S2. Implementation Details on AIpparel

We include details about the network architecture and training hyperparameters of AIpparel.

S2.1. Network Architecture

As described in Section 3 of the main paper, AIpparel is built on top of LLaVA-1.5 7B [36]. Therefore, the majority of the network, except for the newly added regression heads $g_\theta^{(\mathrm{e})}, g_{\theta'}^{(R)}$, and the positional embedding projection layers $h_\varphi^{(\mathrm{e})}, h_{\varphi'}^{(R)}$, are identical to LLaVA-1.5 7B. For completeness, we only summarize the key parameter values we used here. Please refer to their paper for architectural details. LLaVA-1.5 7B fine-tunes LLama 2 [62] with a vision encoder on a visual question-answer dataset. Specifically, it has a context length of 4049 and a hidden dimension of 4096. Its language model is a 32-layer transformer with 32 head attention layers. Its vision encoder is CLIP [52]. Each image is converted into 255 clip tokens before getting projected into the language model's embedding space using a custom projector.

To extend LLaVA-1.5 7B for sewing pattern prediction, we expand the vocabulary of the model to include the special tokens defined in Section 3.2 of the main paper. In total, this results in 122 additional tokens added to the vocabulary of LLaVA-1.5 7B. Each of the tokens is initialized to be the average embedding from the existing vocabulary.

Besides additional vocabulary, we also add two additional regression heads $g_\theta^{(\mathrm{e})}, g_{\theta'}^{(R)}$, and the positional embedding projection layers $h_\varphi^{(\mathrm{e})}, h_{\varphi'}^{(R)}$ to the architecture described above. As described in Section 3 of the main paper, the regression heads will take the output hidden embedding from the language transformer to regress vertex and control point positions using $g_\theta^{(\mathrm{e})}$ and the transformation with $g_\theta^{(R)}$. Specifically, both of the regression heads are two-layer perceptrons with ReLU non-linearity. Both heads map the 4096-dimensional output embedding to the parameter space. For $g_\theta^{(\mathrm{e})}$, the output dimension is 8, representing vertex and control points in different channels. Specifically, the first two channels as vertex regression, mapping to the second endpoint of the associated edge. The next four are used for control points to the quadratic and cubic

Bézier curves. Finally, if the associated edge is an arc, the last two channels are used to map to an additional point on the arc besides the two endpoints. During training, only the associated channels for each edge are supervised and the unused channels are masked out for back propagation. With the same architecture, $g_\theta^{(R)}$ has an output dimension of 7, with the first 3 being the translation and the last four being the rotation represented in quaternion.

Finally, the positional embedding layers are also two-layer perceptions with ReLU non-linearity. $h_\varphi^{(\mathrm{e})}$ maps the 2-dimensional vertex coordinate to a 4096-dimensional hidden embedding. The output is then added to that edge type token's vocabulary embedding before inputting through the language transformer. Similarly, $h_\varphi^{(R)}$ maps the 7-dimensional transformation for each panel to a 4096-dimensional hidden embedding. Then the output is added to the vocabulary embedding of the transformation token `<R>`.

Both the regression heads and the positional embedding projection layers are initialized to have zero weights in the final layer so that the output before fine-tuning is unaltered.

S2.2. Training Details

AIpparel is trained for a total of 12,750 steps with a total batch size of 320, and a learning rate of 0.00005 with cosine learning rate decay to zero in 15,000 steps. We also warm-start the fine-tuning from zero learning rate to the default in the first 100 steps. We use $\lambda = 0.1$ to balance the regression losses and the cross-entropy loss in Equation (2) of the main paper. We use DeepSpeed ZeRO Stage 2 [53] to parallelize the training on 8×H100 GPUs. The entire training took around 312 H100 GPU hours. We train on all modalities in our GCD-MM jointly. Specifically, we include four different modalities from GCD-MM: *text* $\rightarrow$ *sewing pattern, image* $\rightarrow$ *sewing pattern, text and image* $\rightarrow$ *sewing pattern*, and *sewing pattern and editing instruction* $\rightarrow$ *edited sewing pattern*. During each training step, the batch is formed by randomly sampling each of the four modalities with a preset sampling ratio. Specifically, we sample images, texts, image+text, and editing data with the ratio of 3:2:4:1. We randomly split our dataset into 90%, 5%, and 5% for training, validation, and testing. All of our qualitative results are samples from the testing split. While previous works [21, 29, 38] use relative coordinates to represent the control point coordinate, we use absolute coordinates to represent the additional edge parameters. Prior to training, we normalize vertex coordinates and transformation using the global mean and standard deviation computed from all sewing patterns in GCD-MM. Additionally, for input to the positional embedding projection layers, we discretize the input into 256 discrete values ranging between ± 4 standard deviation values for robustness during generation.

S3. Experiment Details And Additional Results

We detail the experiment setup and baselines for the result section (Section 4 of the main paper). Further, we also include additional ablation results and qualitative comparisons.

S3.1. Sewing Pattern Prediction from Images

Setup & Baseline Details. We will describe the image-to-garment prediction experiment showcased in Section 4.1 of the main paper in detail. We will also report comparisons on two datasets: GCD-MM and SewFactory.

For GCD-MM, we use our model trained with multimodal data described in Section S2.2 to evaluate the qualitative and quantitative results showcased in Table 2 and Figure 3 of the main paper. To compare with SewFormer [38], we adapt its pre-trained model for sewing pattern prediction on GCD-MM. Specifically, we expand the per-panel query embedding from its default number of 23 to 75 to accommodate all the different panel classes present in GCD-MM. We initialize the newly added panel query embeddings as the average embedding from the pre-trained weights. Similarly, we expand the per-edge embedding from 14 to 39. Furthermore, because GCD-MM contains cubic Bézier curves and arcs, which the SewFactory dataset does not have, we also extend per-edge parameterization from using four channels (2+2: endpoint + optional quadratic Bézier control points) to seven channels (2+4+1: endpoint, control point parameters, arc flag). Specifically, the arc flag takes a value of 0 or 1, indicating if the edge is an arc. If the arc flag is 1, the first two control points would take a value equal to the relative coordinate of the third point on the arc. If the arc flag is zero, then the four channels will be the relative coordinates of the two control points in the Bézier curve. We keep the network architecture the same except for the above modifications. We fine-tune the pre-trained SewFormer model adapted as above for a total of 16 epochs on the same training split AIpparel is trained on, using a learning rate of 0.00005 and a batch size of 8 on 2×Quadro RTX 8000 GPUs. Except for these, we use the default hyperparameters provided by SewFormer. The validation loss no longer increases after 16 epochs, so we stop the training and use it for comparison.

For comparison on SewFactory, we use the pre-trained SewFormer model as our baseline. However, because the SewFormer authors did not release their train and test split, we show a comparison on a custom test set for this experiment. Specifically, we first train AIpparel on SewFactory data, with a different random split, from scratch for a total of 3750 steps on 8×A100 GPUs using the same hyperparameter settings as described in Section S2.2. Then, we evaluate our model on the custom test set. In this way, we ensure a fair comparison with the baseline as the test set should contain a mixture of training and testing examples for both methods.

Additional Qualitative Visualization. Figure S5 showcases additional image-to-garment prediction result comparisons to the SewFormer baseline in both the GCD-MM (left) and SewFactory (right) datasets. Our model in general predicts more correct sewing patterns following the guidance of the input image than SewFormer.

Sewing Pattern Prediction from In-the-wild MultiModal Inputs. While AIpparel is trained on procedurally generated sewing patterns and annotations, it is able to generalize the in-the-wild input due to the large-scale data it trains on, as well as the world-level knowledge that it inherits from the large multimodal model. Figure S7 showcases our model's sewing pattern prediction from an in-the-wild image with GPT-generated text descriptions.

S3.2. Sewing Pattern Prediction from Texts

We showcase additional text-to-sewing pattern generation visualization from AIpparel in Figure S6. Notice that our method is able to output correct sewing patterns from long, detailed text descriptions. Moreover, our generated sewing patterns also closely follow the key characteristics described in the text input.

S3.3. Sewing Pattern Prediction from Multimodal Input

Setup. For our multimodal evaluation, we utilize 20 samples for each of the following modality combinations: (1) image, (2) text, (3) image + text, (4) occasion, and (5) editing. These samples are generated following the procedure outlined in Section S1. To ensure proper testing, these test samples are entirely distinct from the training and validation sets used in other experiments.

To benchmark our method, we compare it against two state-of-the-art baselines: SewFormer and DressCode. SewFormer processes image-based inputs, while DressCode is designed for text-based inputs. Since these baselines are limited to specific modalities, we convert multimodal inputs into formats compatible with their architectures. For SewFormer, we use DALL-E 2 to generate a single 512x512 image from non-image inputs using tailored prompts. For DressCode, we convert inputs into keyword-based formats with GPT-4o.

The evaluation of our method and these baselines is conducted using Garment Accuracy, a metric defined as the product of Panel Accuracy and Edge Accuracy, which quantifies the percentage of garments reconstructed with the correct number of panels and edges. Additionally, we measure the squared distance between the predicted and ground-truth vertex positions to assess the geometric accuracy of the reconstructions.

Baselines. To generate an image input from a non-image modality, we use DALL-E 2 to produce a single 512x512 image. The prompt used for generation always begins with:

> Create an image of a single garment worn by a mannequin. The mannequin should be front-facing and in t-pose.

The prompt is tailored to each input modality by appending the following continuations.

> - **Text:** Make sure that the garment follows this description: + text
> - **Occasion:** Make sure the garment suits the following occasion: + text
> - **Editing:** Make sure the garment looks like if this edit + edit + was applied to the garment.

Similarly, to convert any input modality into a keyword-based format compatible with DressCode, we design distinct prompts based on the modality. Each prompt is constructed as a concatenation of the following starting phrase:

> Describe the garment in a list of comma separated keywords. Give a maximum of 5 keywords.

and a modality specific continuation:

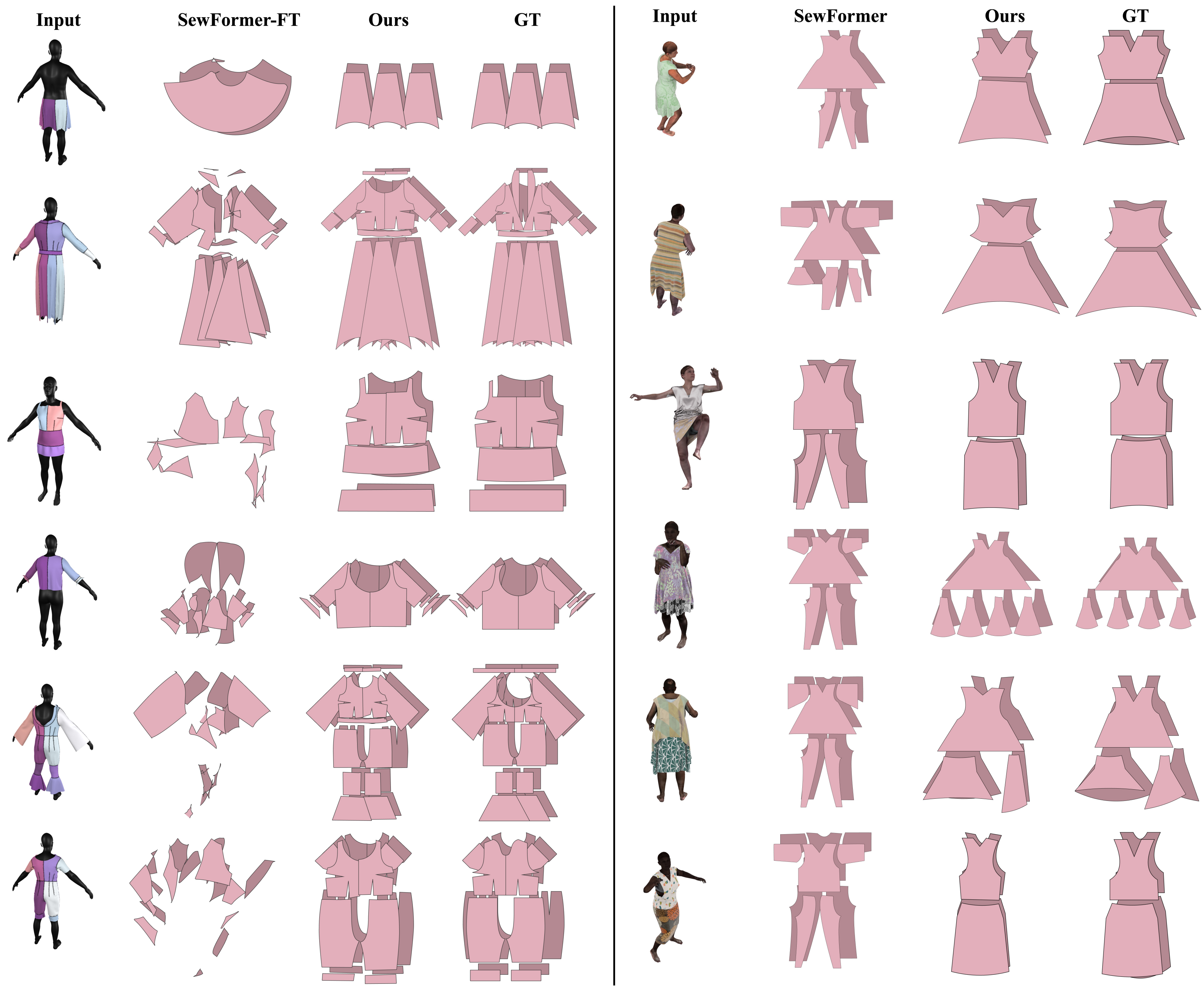


Figure S5. **Additional Image to Sewing Pattern Visualizations.**

Input Text

The garment is a **sleeveless hoodie** with a **deep cut neckline**. The hood is attached to the main body with seams running along the back neckline and shoulder areas. **The front panel includes a deep neckline, with seams attaching it to the side and shoulder panels**. The side panels extend under the arms and connect to the back panel, which comprises multiple sections joined with vertical seams. **The silhouette is fitted**, accentuating the upper torso, while the hood adds a distinctive layered look to the upper garment.

The **dress** is a **mini-length garment** featuring a **sleeveless** design with **a short square neckline** and a short square back. It includes a **wide waistband** positioned at a **high rise**. The front and back panels connect seamlessly at the shoulders and sides, with the waistband acting as a horizontal dividing panel, emphasizing the high-rise waistline. **The overall silhouette is fitted**, with the bodice and skirt sections distinctly separated by the wide waistband, ensuring a structured and defined shape.

The **jumpsuit** features a **maxi silhouette with long sleeves**. **It has an extended neckline creating a deep cut at the front**. The garment is constructed with multiple panels joined throughout, including horizontal and vertical connections. **The sleeves are attached at the shoulder seams and feature a wide, loose fit**. **The lower part of the jumpsuit includes flared panels** that add volume to the hem. The design displays a combination of fitted and loose sections, enhancing the overall silhouette with distinctive seam placements.

The **dress** features **an asymmetric design with a single right long sleeve**. **The neckline is also asymmetric,** providing a unique contour to the upper section. Panels are carefully constructed to achieve the asymmetric top, with precise seams joining the right sleeve to the body. **The silhouette is elongated with a clean, straight cut extending to full length**. Distinctive characteristics include the single sleeve and the non-traditional neckline, producing a visually striking and modern appearance.

The jumpsuit features **a maxi silhouette** with **long sleeves** and **a short v-neckline** in both the front and back. **The front panel is divided by a central seam running from the neckline to the waist**, where it meets **the waistband seam**. The bodice and lower body panels are connected at the waistband, with additional **vertical seams along the torso sides extending to the hem**. The back contains a yoke panel that connects to the lower body panels with vertical seams. The sleeves are attached to the bodice with shoulder seams extending to the cuffs.

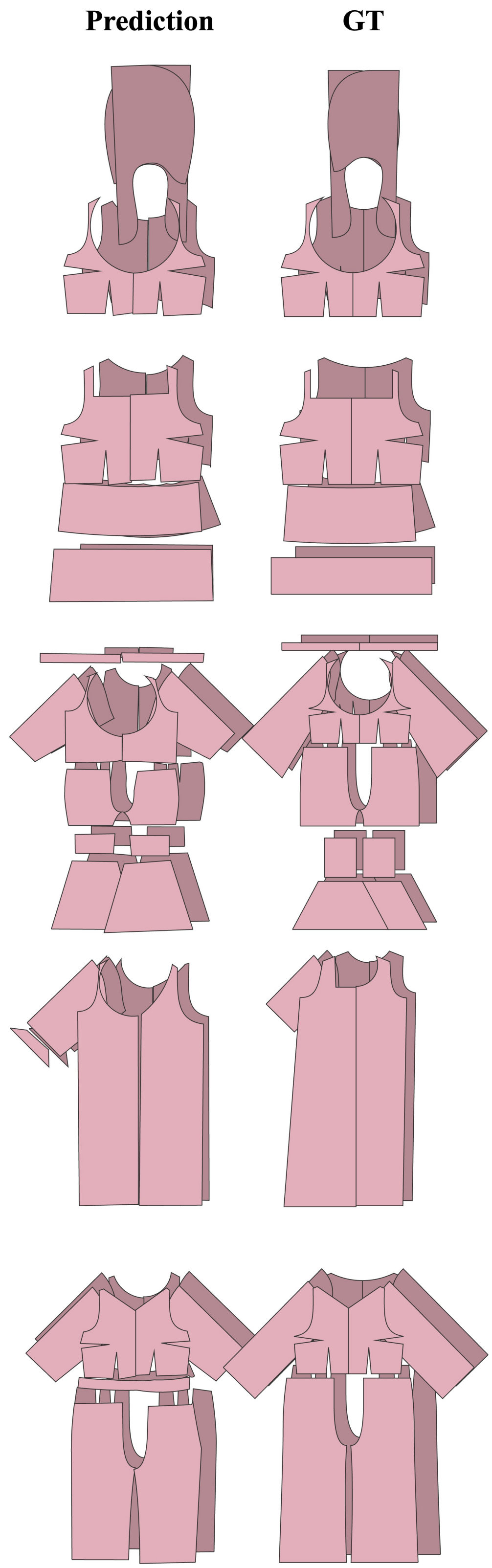


Figure S6. **Text-conditioned sewing pattern generation.** AIpparel generates accurate sewing patterns closely following the text descriptions. Notice that the characteristics described in the bolded phrases all appear in the generated sewing patterns.

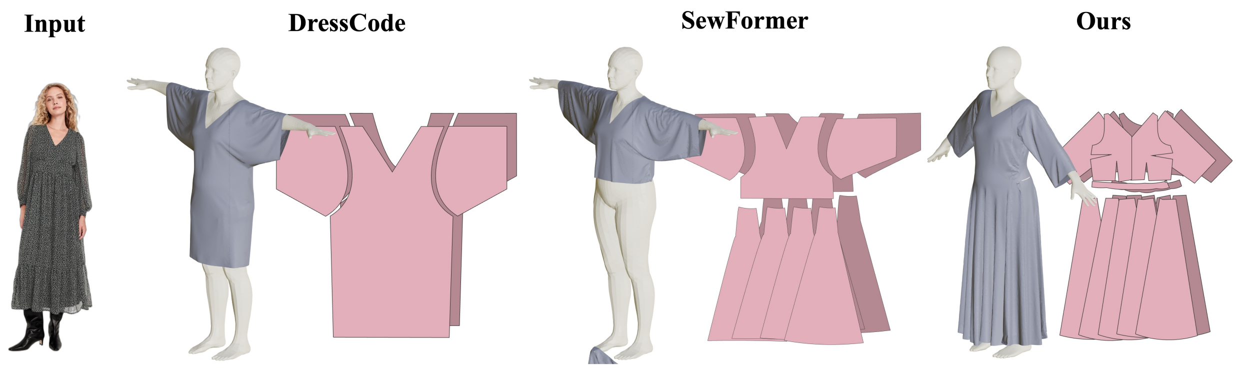


Figure S7. **In-the-wild Image to Garment Example.** Our model is able to predict a sewing pattern more aligned with the input image compared to the baselines. Notice that SewFormer did not drape correctly, resulting in a missing bottom.

- **Text:** Make sure that the garment follows this description: + text
- **Image:** Make sure that the garment looks like this image.
- **Text + Image:** Make sure that the garment looks like this image and follows this description: + text.
- **Occasion:** Make sure the garment suits the following occasion: + text
- **Editing:** Make sure the garment looks like if this edit + edit + was applied to the garment.

S3.4. Sewing Pattern Editing

We detail the baseline methods we used for Table 4 and Figure 5 in the main paper. Using existing models, we extend SewFormer and DressCode to translate the sewing pattern and editing instructions to their input domains. Specifically, for SewFormer, we take the editing instruction and rendered image from GCD-MM and translate the rendering image using a pre-trained InstructPix2Pix [12] with the editing instruction as input. The output from InstructPix2Pix is a garment image generated based on the editing instructions and the input rendering. With this input image, we query the SewFormer-FT baseline to obtain the final sewing pattern. For DressCode, we use GPT4V to translate the editing instructions and rendered image into short keywords that describe the edited garment. This is then used to query the pre-trained DressCode and obtain the sewing pattern. The text prompt we use for querying GPT4V is the following:

> You are given a list of attributes describing a garment. Your task is to modify the list according to an editing instruction provided.
> To accomplish this: 1. If the attribute related to the instruction already exists in the description, locate and modify it to reflect the new information. 2. If the attribute is not present, add a new entry to the description that fulfills the instruction.
> 3. Ensure that no other attributes are altered unless necessary for consistency or clarity following the modification.
> Once the changes are complete, return the list of attributes, without any additional information.

We evaluate this task using the test split of GCD-MM, containing approximately 6,000 editing samples.

Additional Qualitative Visualization Figure S8 shows additional visualization of the editing tasks as shown in Figure 5 of the main paper. Notice that our model is able to correctly edit the sewing pattern with a diverse set of instructions.

S3.5. Ablation Study

Setup & Baseline Details. Table 5 in the main paper shows an ablation study on our proposed tokenization scheme in Section 3.2 of the main paper. As described in Section 4.4, we use text-to-image as our ablation task to conduct an equal comparison of our model with DressCode [21]'s pre-trained model. Futhermore, we swap our tokenizer into DressCode's model, to ensure an equal comparison. We also do the same for the configuration, *Ours w.o. reg.*, which uses the proposed sewing pattern tokenization scheme without the usage regression heads. We train both models from scratch with a learning rate of 0.0006 and a total batch size of 512 on 2×Quadro RTX 8000 GPUs, for a total of 30,400 steps until convergence.

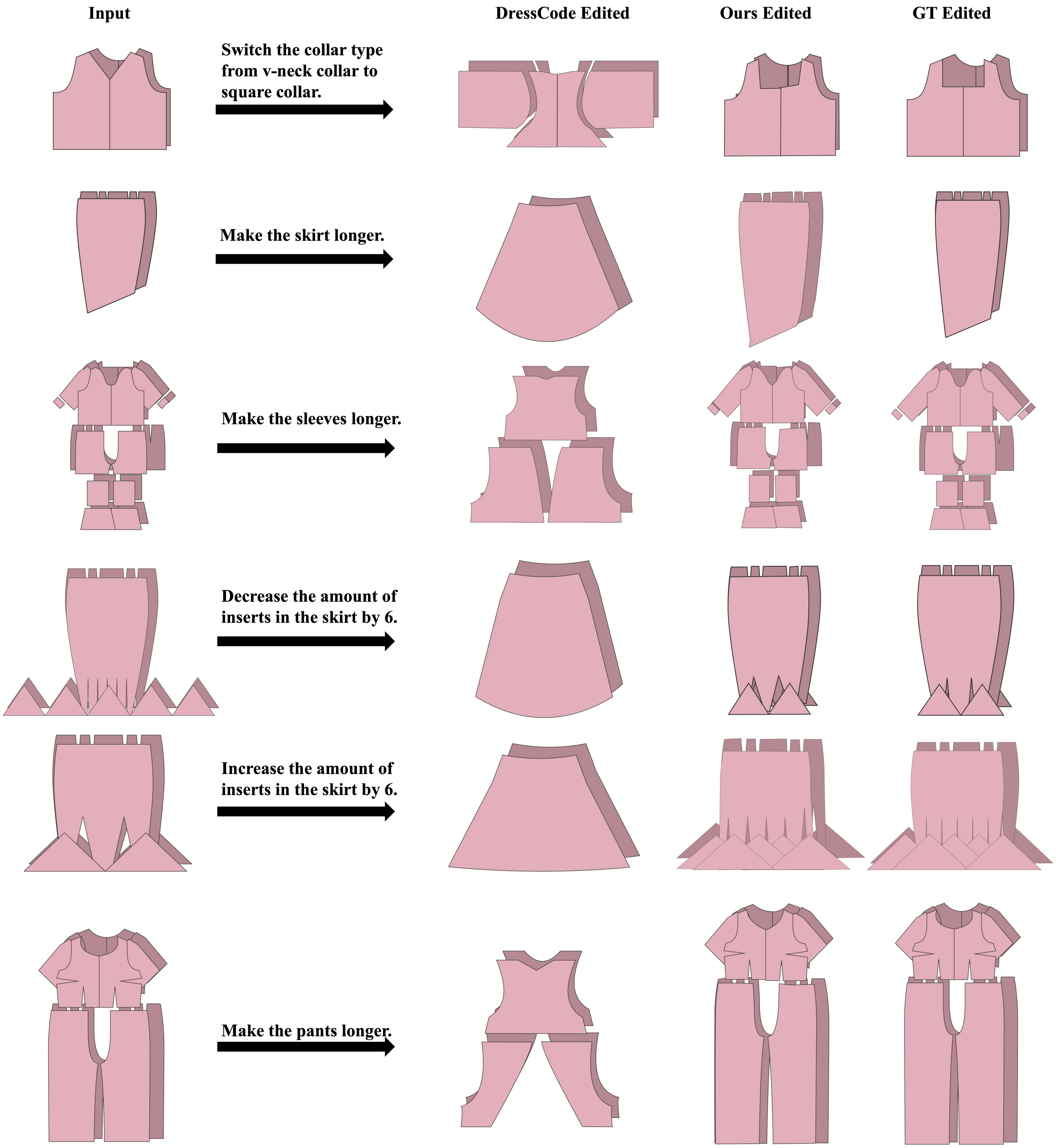


Figure S8. **Additional Visualization for Sewing Pattern Editing.** The task is to predict a sewing pattern that closely matches the input sewing pattern while following the editing instructions (text above the arrow). Notice that despite the diverse kinds of editing instructions we give, our methods can output sewing patterns that closely follow the instructions and the input sewing pattern. In the meanwhile, the baseline cannot achieve a similar effect because it takes only takes in text as input, losing structural details.

Additional Ablation Study. Table S2 shows a qualitative comparison studying the effectiveness of full model fine-tuning versus LoRA [23] fine-tuning. The table reports reconstruction metrics on the image-to-garment prediction task on GCD-

Method	Panel L2 (↓)	#Panel Acc (↑)	#Edge Acc (↑)	Rot L2 (↓)	Transl L2 (↓)	#Stitch Acc (↑)
LoRA	13.7	31.6	45.4	.020	5.1	.088
AIpparel	**5.4**	**85.2**	**82.7**	**.020**	**2.7**	**77.2**

Table S2. **Ablation Study: Fine-tuning Comparison.** The scores are reported on the image-to-garment prediction tasks on GCD-MM dataset. The metrics indicate that full model fine-tuning significantly outperforms LoRA fine-tuning, allowing the base model to better adapt to sewing pattern understanding.

Method	Panel L2 (↓)	#Panel Acc (↑)	#Edge Acc (↑)	Rot L2 (↓)	Transl L2 (↓)	#Stitch Acc (↑)
6 layers	5.93	83.6	81.0	.008	2.9	74.3
5 layers	6.10	84.2	80.7	0.010	2.8	73.4
4 layers	5.94	83.2	81.3	0.011	3.0	74.7
3 layers	5.92	83.7	80.9	0.010	2.9	73.7
2 layers	**5.4**	**85.2**	**82.7**	**.020**	**2.7**	**77.2**

Table S3. **Ablation Study: Number of Layers in Regression Heads.** The scores are reported on the image-to-garment prediction tasks on GCD-MM dataset.

MM dataset. For the LoRA model, we use rank 8 and only fine-tune the query and key projection layers following previous works [19, 32]. The model is trained with the same hyperparameter settings described in Section S2.2 for 8250 steps. The metrics indicate that the full fine-tuning model significantly outperforms the LoRA fine-tuned version, indicating that fine-tuning all weights in the language transformer is essential for understanding sewing patterns.

Additional Qualitative Visualization. Figure S9 shows additional visualizations for the ablation study in Section 4.4 of the main paper. Notice that our model in general demonstrates better sewing pattern prediction ability than DressCode. This can be seen in the pants prediction in the second and third rows of the figure, where DressCode does not predict the correct sewing pattern.

S3.6. Draping Details

We use the draping pipeline provided by GarmentCode [30] for converting sewing patterns to a 3D mesh of the garment draped on a standard female SMPL model in A-pose. Specifically, the draping process consists of creating the boxed mesh and using Nvidia-Warp [42] for cloth simulation. To obtain the garment in arbitrary poses and in a motion sequence, we follow the simulation pipeline provided by PhysAvatar [82], which uses Codimensional Incremental Potential Contact (C-IPC) [35] simulation for cloth simulation. For simulation details, please refer to Zheng et al. [82]. Finally, the simulated mesh sequence is imported to Blender for texturing and rendering.

S3.7. Human Study

We conducted a user study to compare sewing patterns generated from multimodal inputs using AIpparel with those using baselines. Specifically, we deployed 10 multiple-choice questions asking which garment better aligns with the input prompts while maintaining realism. The questions contain a combination of sewing patterns generated from images, texts, and editions of existing sewing patterns from different methods. We collected responses from 73 participants and Fig. S10 shows the favorability comparison for each modes of generation. AIpparel is more favorable in all modes, aligning with our quantitative and qualitative results.

S4. Discussion

We expand our discussion in Sec. 5 of the main paper to include further limitations, future work, and social impact.

S4.1. More Discussion on Limitations and Future Work

Due to computational resource constraints, we only train AIpparel on part of the GCD data, and AIpparel outputs a single modality, sewing pattern. As the community gets more computing resources, we are excited to see works extending our methods to larger datasets with richer annotations. It is an interesting direction to further scale up AIpparel to study the emergence of abilities like few-shot or in-context generalization to novel garment generation tasks or perform chain-of-thoughts to achieve a complex garment design. It is also an interesting direction to study how to further enlarge sewing

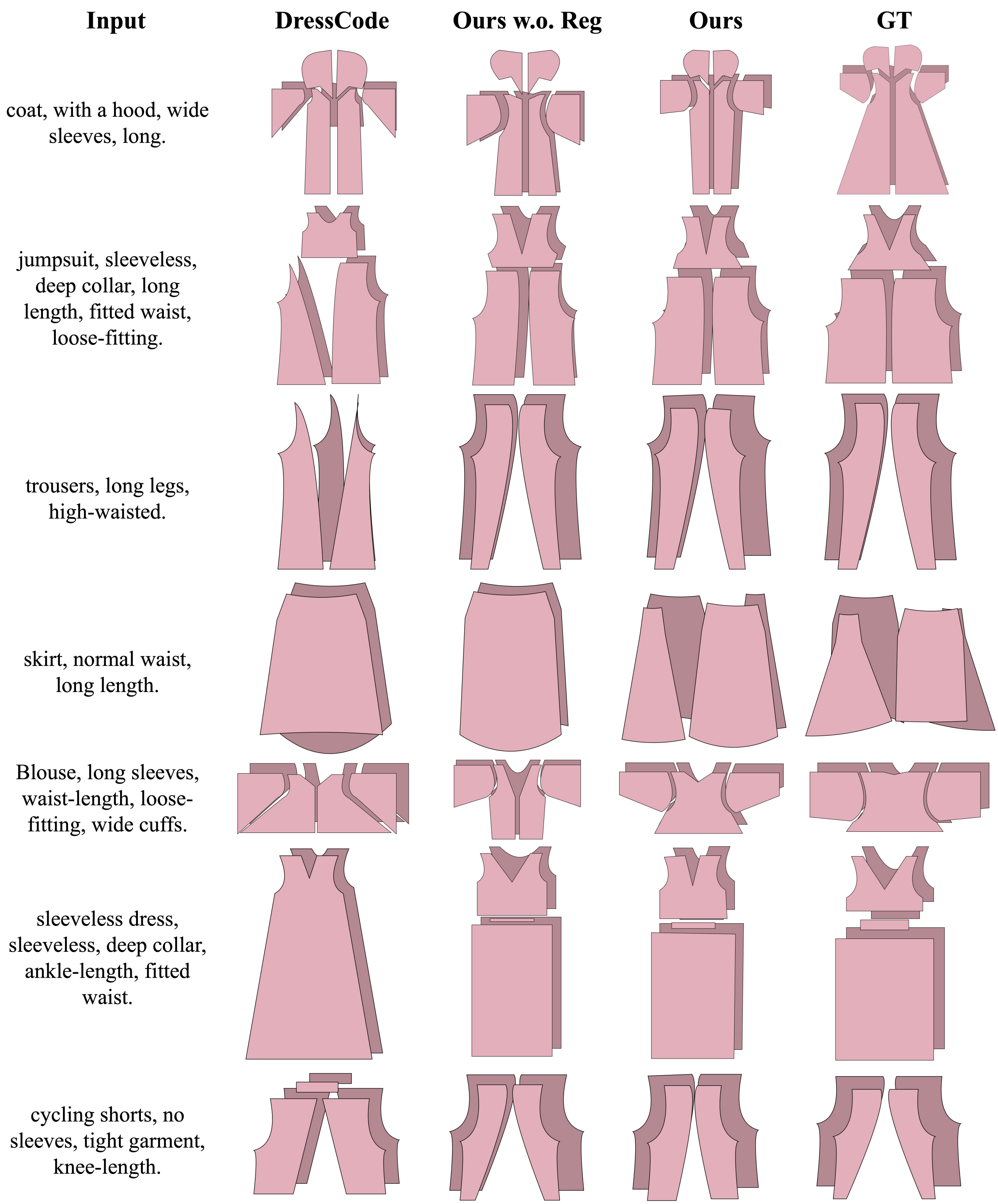


Figure S9. **Additional Visualizations for Ablation Study.**

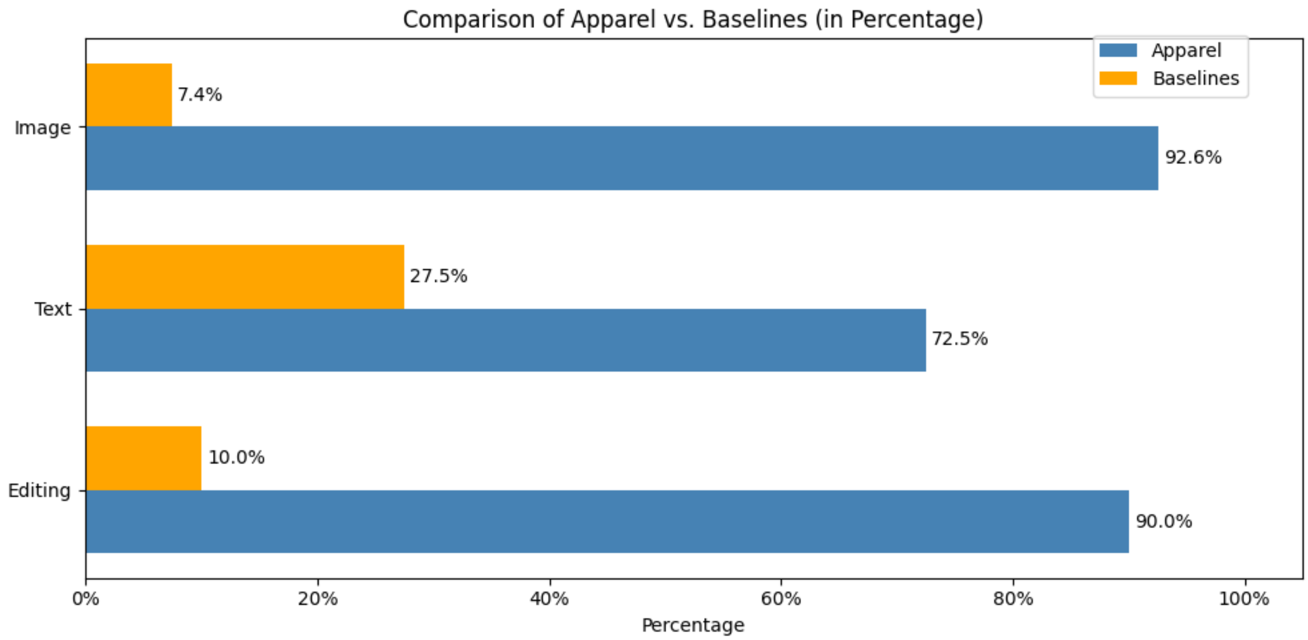


Figure S10. **User Favorability Comparison of Apparel vs. Baselines for Multimodal Generation.**

pattern datasets with more variations and more annotations. For example, reflecting realistic variations of fabric properties can enable more accurate sewing pattern prediction.

Bias and Comprehensiveness of GCD-MM. AIpparel can inherit the bias from the sewing pattern dataset used to create GCD-MM. In fact, GarmentCodeData [31] discusses such biases in its limitation section including only sewing patterns fitted to statistical models computed from a pool of healthy European and North American adults, hence limiting the size variations within the sewing patterns of GCD. However, we note that our data curation pipeline outlined in the paper can be used for other sources. By applying our pipeline to other, less biased, and more comprehensive sewing pattern datasets, we can still improve their quality by creating annotations for the sewing patterns.

S4.2. Further Societal Impacts

Besides the concerns of hallucination and bias that we inherit from our base model, LLaVA, we also acknowledge that our generated sewing patterns might not produce suitable garments for all communities, due to the limited body type and style selections within the data we trained on. It is important to study how to improve our method and dataset annotation on more diverse sewing patterns and body types in the future.

Another potential risk of our work is the potential bias we inherit from foundation models in our annotation generation process. Because we use large models such as GPT-4V for data generation, existing biases in these models will also be included in our generated annotations. However, because the prompts we used (see Section S1.1) encourage the model to generate descriptions based on the given images and keyword phrases, we did not find any immediate systematic bias present in our annotations.