

Multimodal Latent Diffusion Model for Complex Sewing Pattern Generation

Supplementary Material

In this supplementary document, we first provide comprehensive details of complex sewing patterns (Sec. 1). Afterward, we compare with the parametric method [4], which needs a delicate selection of values and professional knowledge of garment designs (Sec. 2). We also compare with concurrent open-sourced methods (Sec. 3). Then, we provide the annotation details of our dataset to better demonstrate the precision and accuracy of the dataset (Sec. 4). To substantiate the efficacy of extended representation and our two-step training strategy, we perform an ablation study on representation (Sec. 5) and the training strategy (Sec. 6). Additionally, we provide examples of our user study, which ensure a fair and objective evaluation of our method compared to others (Sec. 7). We further include a user case (Sec. 8) and examples of generated garments that demonstrate the robustness and generative capabilities of our method across various sketches and body types (Sec. 9). We also provide a supplementary video demonstrating that our generated garments can be directly used in CG pipelines for animation production, showcasing high-fidelity simulation of cloth collisions and wrinkle formation. The garment visualization results are rendered following a circular trajectory, effectively emphasizing the superior fit of garments to body shape compared to other methods.

1. Representation Details

Extended Representation. The binary concrete representations of different edge types, attachment types, and stitches are depicted in Fig. 1 alongside their corresponding annotations. For edges, in addition to the vector $V_{i,j}$ representing from the start point to the endpoint, the cubic line employs the control parameters $C_{i,j}^b \in \mathbb{R}^4$ to define two control points (x_1, y_1) and (x_2, y_2) in the 2D coordinate. The circle line uses additional control parameters $C_{i,j}^r \in \mathbb{R}^3$, which specify the radius r and four rotations with two binary flags, including the counterclockwise acute angle $([0,0])$, the clockwise acute angle $([0,1])$, the counterclockwise reflex angle $([1,0])$, and the clockwise reflex angle $([1,1])$. Furthermore, edge types are denoted as follows: $E_{i,j}^t = [0,0]$ for the straight line, $E_{i,j}^t = [0,1]$ for the quadratic line, $E_{i,j}^t = [0,0]$ for the cubic line, and $E_{i,j}^t = [1,1]$ for the circle line. The attachments are visually distinguished by highlighting the associated edges in red and annotating them with the name and value of $A_{i,j}$. Edges without attachment are not highlighted and use the default value of $A_{i,j} = [0,0,0]$. There are six kinds of attachment types, *i.e.*, lower interface $([0,0,1])$, right collar $([0,1,0])$, left collar $([0,1,1])$, strapless top $([1,0,0])$, right

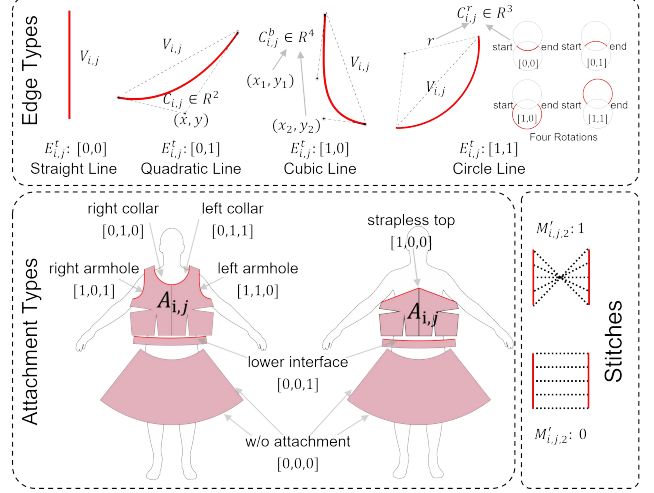


Figure 1. **Representation details.** We present various kinds of edges, attachments, and stitches with detailed annotations.

armhole $([1,1,0])$, and left armhole $([1,0,1])$. For reversal stitch $\{M'_{i,j,2} \in \{0,1\}\}$, 0 means the stitch direction does not need reversal, while 1 means the stitch direction needs to be reversed. With the detailed representation of sewing patterns, users can convert the sewing patterns into vector representations as input for neural networks.

Panel Order invariance. Since all panels in the dataset are fixed, we define an order from the upper body to the lower body to construct the vector representation, guaranteeing the panel order invariance. The panel order is represented as below:

```

1 {
2   "1": ["right_sleeve_b"],
3   "2": ["right_sleeve_f"],
4   "3": ["right_ftorso"],
5   "4": ["right_btorso"],
6   "5": ["left_ftorso"],
7   "6": ["left_btorso"],
8   "7": ["left_sleeve_f"],
9   "8": ["left_sleeve_b"],
10  "9": ["left_collar_back"],
11  "10": ["left_collar_front"],
12  "11": ["right_collar_front"],
13  "12": ["right_collar_back"],
14  "13": ["sl_right_cuff_f"],
15  "14": ["sl_right_cuff_b"],
16  "15": ["sl_right_cuff_skirt_f"],
17  "16": ["sl_right_cuff_skirt_b"],
18  "17": ["sl_left_cuff_skirt_f"],
19  "18": ["sl_left_cuff_skirt_b"],
20  "19": ["sl_left_cuff_f"],
21  "20": ["sl_left_cuff_b"],
22  "21": ["left_hood"],
23  "22": ["right_hood"],
24  "23": ["wb_back"],

```

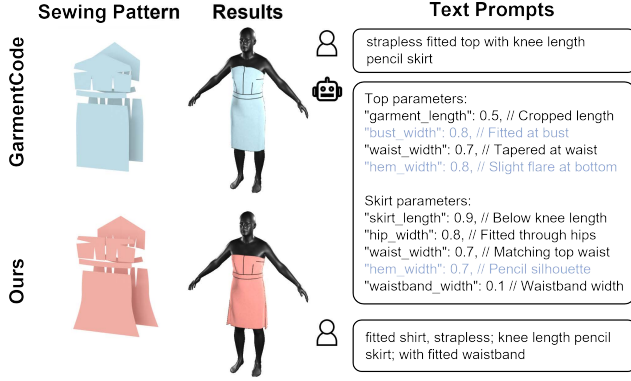


Figure 2. **Comparison with parametric method.** We present the garments and draping results for our SewingLDM and parametric method GarmentCode [4]. GarmentCode needs a delicate selection of values. In contrast, our method can generate garments under intuitive conditions like natural language or sketches, which provide an easier way for garment generation.

```

25 "24": ["wb_front"],
26 "25": ["pant_f_l", "skirt_front",
27 "skirt_panel_0"],
28 "26": ["pant_b_l", "skirt_back",
29 "skirt_panel_1"],
30 "27": ["pant_f_r", "skirt_front_0",
31 "ins_skirt_front_0", "skirt_panel_2"],
32 "28": ["pant_b_r", "skirt_back_0",
33 "ins_skirt_back_0", "skirt_panel_3"],
34 "29": ["pant_r_cuff_skirt_f", "skirt_front_1",
35 "ins_skirt_front_1", "skirt_panel_4"],
36 "29": ["pant_r_cuff_skirt_b", "skirt_back_1",
37 "ins_skirt_back_1", "skirt_panel_5"],
38 "30": ["pant_l_cuff_skirt_f", "
39 ins_skirt_front_2",
40 "skirt_panel_6", "skirt_front_2"],
41 "31": ["pant_l_cuff_skirt_b", "ins_skirt_back_
42 2",
43 "skirt_panel_7", "skirt_back_2"
44 ],
45 "32": ["pant_l_cuff_f", "ins_skirt_front_3",
46 "skirt_panel_8", "skirt_front_3"],
47 "33": ["pant_l_cuff_b", "ins_skirt_back_3",
48 "skirt_panel_9", "skirt_back_3"],
49 "34": ["pant_r_cuff_f", "ins_skirt_front_4",
50 "skirt_panel_10", "skirt_front_4"],
51 "35": ["pant_r_cuff_b", "ins_skirt_back_4",
52 "skirt_panel_11", "skirt_back_4"],
53 "36": ["skirt_panel_12", "ins_skirt_front_5"],
54 "37": ["skirt_panel_13", "ins_skirt_back_5"],
55 "38": ["skirt_panel_14"]
56 }

```

2. Comparison with Parametric Method

Except for generation methods, GarmentCode [4] allows users to model complex garments by selecting different parameters and producing desired sewing patterns. However, selecting various parameters is not intuitive and requires professional knowledge of garment design, limiting

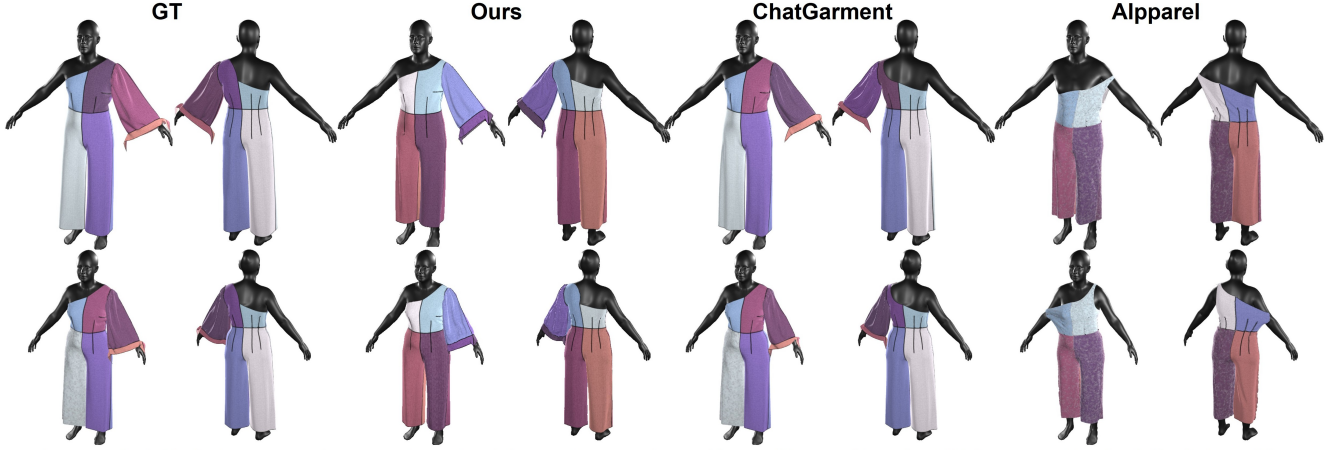
its widespread promotion. To enable the production of the desired sewing patterns through users' prompts, an easy way is to leverage the powerful ability of a large language model, like GPT4 [1]. We simply ask GPT4 to generate various values between 0 to 1 to satisfy the sewing pattern designs of [4], as illustrated in Fig. 2. With the designed prompt, GPT4 can truly provide instructions for garment design. However, most of the generated values are not concerned with garment shape in GarmentCode [4] and still require professional knowledge of garment designs and predefined templates. In summary, GarmentCode [4] needs indispensable manual processing to produce the desired garments. In contrast, our SewingLDM can generate the desired garment through more intuitive conditions, *i.e.*, text prompts and sketches, providing easier tools for garment designs and boosting daily garment production.

3. Concurrent Work Comparison

For concurrent sewing pattern generation methods, both Design2GarmentCode [8] and ChatGarment [3] use autoregressive models to convert inputs into rule-based parameters specially designed in GarmentCode, limiting their generalization beyond pre-defined garment categories. AIpparel [6], enabling the generation of any extended sewing patterns without requiring category-specific parameterization, while it only considers the average body shape draping. We also compared with these open-sourced methods, ChatGarment [3] and AIpparel [6]. Since all methods are trained on the GarmentCodeData [5], we randomly chose some examples in the GarmentCodeData for fair comparison, shown in Fig. 3. ChatGarment directly generates the design files and uses GarmentCode to generate appropriate sewing patterns and drape to other body shapes. AIpparel generates sewing patterns and warps to the average body shapes and drapes to other body shapes by complete pattern scaling manually. Compared with concurrent works, our SewingLDM shows comparable results with ChatGarment and shows much superior results with AIpparel. Moreover, due to the body shape conditions, our method can fit various body shapes, which provides a more user-friendly approach to getting the desired made-to-measure garments.

4. Dataset Annotation Details

Directly employing multimodal models for generating textual descriptions of images may lead to inaccuracies, as these large-scale multimodal models are incapable of fully capturing and characterizing garment-specific features, often resulting in hallucination phenomena. To obtain precise descriptions of garment characteristics, we have established a corresponding descriptive framework based on the parameters of a parametric model, generating distinct descriptions for various garment features, which are then concatenated



Asymmetrical shirt features a strapless right side, while left with wide long sleeve. The neckline combines front B zier curve with a back flip circle neckline.

Figure 3. **Comparison of sewing pattern generation for concurrent works.** Since all the methods were trained on the same dataset, we randomly selected samples from the original dataset to evaluate and compare their performance.

together to form the text annotations of garments. The corresponding descriptions are presented as follows:

```

1 {
2   "sleeve length": {
3     [0, 0.4]: "short",
4     [0.4, 0.6]: "elbow length",
5     [0.6, 0.8]: "three quarter length",
6     [0.8, 1.15]: "long"
7   },
8   "shirt length": {
9     [0, 1]: "short shirt",
10    [1, 1.5]: "shirt",
11    [1.5, 2.4]: "long shirt",
12    [2.4, 2.8]: "knee length shirt dress",
13    [2.8, 3.5]: "long shirt dress"
14  },
15  "neckline width": {
16    [-0.5, -0.1]: "narrow",
17    [-0.1, 0.5]: "normal",
18    [0.5, 1]: "wide"
19  },
20  "neckline depth": {
21    [0.3, 0.4]: "shallow",
22    [0.4, 0.9]: "normal",
23    [0.9, 2]: "deep"
24  },
25  "waist band width": {
26    [0, 0.2]: "narrow waistband",
27    [0.2, 0.5]: "waistband",
28    [0.5, 1]: "wide waistband"
29  },
30  "skirt length": {
31    [-0.2, 0]: "mini",
32    [0, 0.25]: "short",
33    [0.25, 0.4]: "above knee length",
34    [0.4, 0.5]: "knee length",
35    [0.5, 0.55]: "below knee length",
36    [0.55, 0.65]: "midi length",
37    [0.65, 0.85]: "hight ankle length",
38    [0.85, 0.95]: "ankle length"
39  },
40  "godet width": {

```

```

41    [10, 15]: "shallow",
42    [15, 25]: "normal",
43    [25, 50]: "deep"
44  },
45  "godet width": {
46    [10, 15]: "tight",
47    [15, 25]: "normal",
48    [25, 50]: "oversized"
49  },
50  "pant length": {
51    [0.2, 0.3]: "short",
52    [0.3, 0.4]: "above knee length",
53    [0.4, 0.5]: "knee length",
54    [0.5, 0.55]: "below knee length",
55    [0.55, 0.65]: "mid calf length",
56    [0.65, 0.8]: "hight ankle length",
57    [0.8, 0.9]: "full length"
58  }
59 }

```

In addition to characterizing apparel attributes based on numerical data, we also provide corresponding descriptions according to the categories of different garments, for instance, shirt, fitted shirt, straight waistband, fitted waistband, pencil skirt, circle skirt, mermaid skirt, tail dress, etc. Here is an example of garment description: *Upper garment: asymmetrical fitted shirt, right: curve armhole, ruffled shoulder, wide short ruffled sleeve, short shrink cuffs, front wide shallow V neckline and back wide shallow V neckline; left: strapless; Lower garment: ankle length circle skirt with front right deep split.* After getting the detailed description of the garments, we use GPT-4 to refine them, allowing for more diverse descriptions and more general prompt queries. The prompt used to refine them is shown in Tab. 1. For each garment, we utilize PiDiNet [7], a pre-trained edge detection network, to extract garment sketches using its open-source code and checkpoint, thereby enriching the design details of the garment.

- “<prompt>” Please refine the following clothing description to make it more fluent and accurate in English while strictly preserving all technical details, measurements, fabric properties, and design intent. Ensure the language is professional yet natural, suitable for a global audience. Focus on:
 - Clarity: Eliminate ambiguity (if applicable).
 - Consistency: Use standardized terminology (e.g., ‘midi dress’ instead of ‘knee-length dress’).
 - Precision: Highlight functional features (e.g., moisture-wicking, reinforced seams) without exaggeration.
 - Flow: Restructure sentences to enhance readability while retaining all original information.
 - SEO-friendly: Incorporate relevant keywords naturally (if applicable).

Original Text:

[Upper garment: asymmetrical fitted shirt, right: curved armhole, ruffled shoulder, wide short ruffled sleeve, short shrink cuffs, front wide shallow V neckline and back wide shallow V neckline; left: strapless; Lower garment: ankle length circle skirt with front right deep split.]
Provide the revised version.

- This asymmetric tailored shirt features a dual design: Right side: Contoured armhole with ruffled detailing at the shoulder, voluminous short ruffled sleeves, elasticized short cuffs, and dual wide shallow V-necklines (front and back). Left side: A strapless silhouette for bold contrast. Paired with an ankle-length full-circle skirt, the ensemble includes a vented high slit at the right front panel for dynamic movement and versatility.

Table 1. Prompt to refine the dataset annotations.

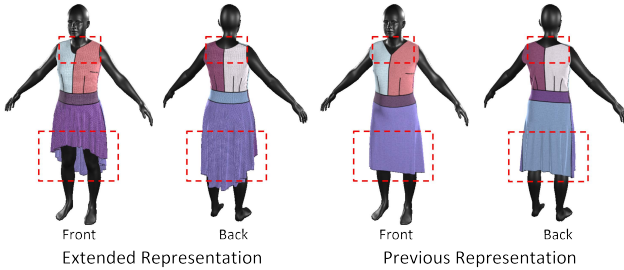


Figure 4. **Visualization of different representation.** We compare the extended representation with the original representation.

5. Extended Vector vs Origin Vector

We also conduct the visualization between the extended representation and original representation to reveal significant advantages in design versatility of the extended representation shown in Fig. 4. With extended representation, our model can generate more diverse garments and more fashion designs, such as the curve neckline and the tail dress. In contrast, the original representation restricts the model to producing only simple garments, failing to capture complex design features and resulting in a notable loss of detail. To address these limitations, the integration of the Garment-CodeData is essential, as it injects detailed knowledge of complex sewing patterns into the generation model. This advancement overcomes the constraints of previous meth-

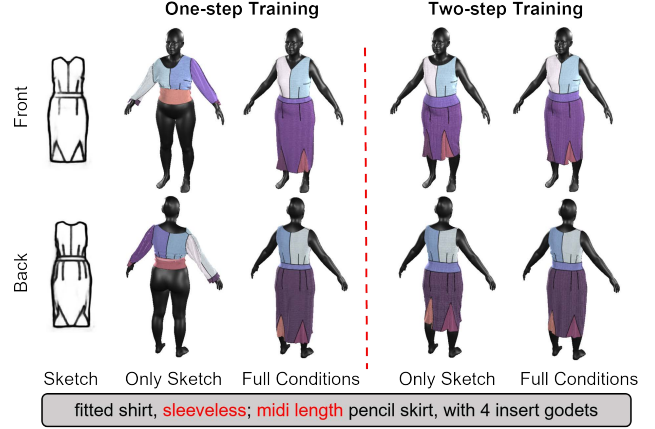


Figure 5. **Ablation on the training strategy.** One-step training shows an unbalance between the multi-modal conditions, failing under only the sketch. In contrast, two-step training helps to faithfully generate the ideal garments with only sketch conditions.

ods, which were unable to represent the extended vector effectively due to more than 30k tokens consumption under current GPU capabilities.

6. One-step Training vs Two-step Training

We additionally take an ablation study on the training strategy. One-step training is unable to balance the multi-modal conditions, so that fails to generate the corresponding garment through only the sketch condition. As shown in Fig. 5, the generated garment loses its midi-length pencil skirt, failing to generate the desired garment. In contrast, the model under two-step training can faithfully generate the corresponding garment with the sketch only, which contains both the sleeveless fitted shirt and the corresponding midi-length pencil skirt. Therefore, two-step training can more effectively inject the sketch conditions into the diffusion model and provide additional control of garment designs, enabling wider usage of our SewingLDM. Moreover, combined with full conditions of text and sketch, SewingLDM can provide more precise control on desired garment generations, meeting users’ requirements.

7. User Study Details

To ensure a fair and objective evaluation of our method compared to other methods, we randomly shuffle the results generated by different methods. Each result is paired with a corresponding textual description, and volunteers are asked to rate the results with a score of 1 – 5 based on the consistency between the results and the texts, as well as the fitness between the clothes and the human bodies. Additionally, we provide 3 supplementary examples as shown in Fig. 6. We show 3D comparison results in the supplementary video.

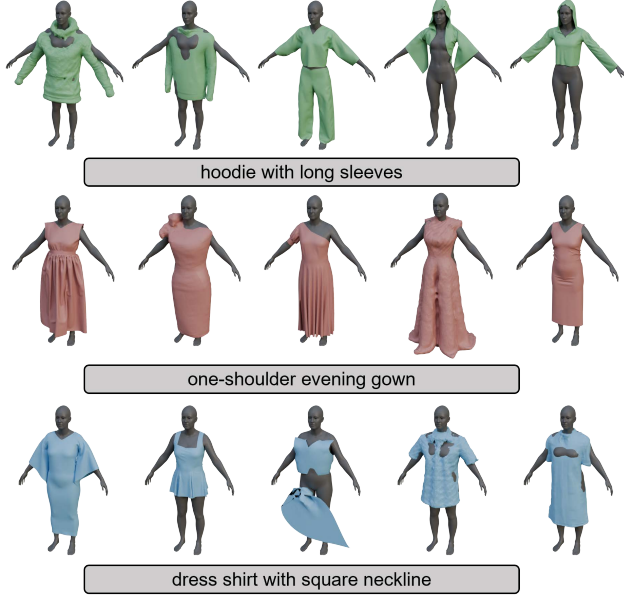


Figure 6. **User study examples.** We present 3 user study examples with random shuffled results.

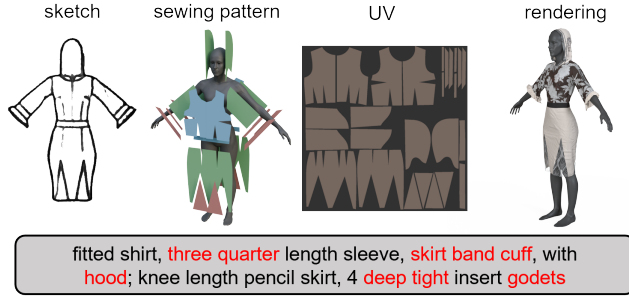


Figure 7. **Use case.** We present an example of an extremely complex sewing pattern generation. With the generated sewing pattern, we can easily paint the UV to produce visually appealing results.

8. Use Case

Our SewingLDM is capable of generating intricately detailed clothing, meeting the current artistic demands for garment design across a wide range of styles, significantly advancing fashion garment design, and supporting everyday users in obtaining apparel tailored precisely to their needs. To demonstrate our superiority in garment generation, we present an extremely complex example of a sewing pattern in Fig. 7. With the detailed textual description and garment sketch, our method faithfully generates the complex sewing pattern, *e.g.*, skirt band cuff, hood, and godets, which significantly helps the artist in creating fantastic texture in UV space, *e.g.*, laces, leather pants, and hat brim.

9. Qualitative Results

For in-domain garment sketches, our method achieves a more granular representation that closely adheres to the

sketch contours. We have also provided a comparative analysis between our approach and other baseline methods within the same domain, as illustrated in Fig. 8. Furthermore, to demonstrate the efficacy of our method, we have conducted additional validation using out-of-domain data that are collected from the Multimodal Garment Designer or hand-drawn, as depicted in Fig. 9 and Fig. 10.

We additionally provide garments tailored to a wide range of body shapes, spanning variations such as short to tall and slim to broad. As illustrated in Fig. 11, our approach enables the creation of garments specifically adapted to different body types. Furthermore, the simulated garments are enriched with physically based rendering (PBR) textures, either generated by DressCode or designed using the Substance 3D Painter software [2], culminating in visually compelling garment representations as shown in Fig. 12.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altmenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2
- [2] Adobe. Substance 3D Painter. <https://creativecloud.adobe.com/apps/all/substance3d-painter>, 2024. 5
- [3] Siyuan Bian, Chenghao Xu, Yuliang Xiu, Artur Grigorev, Zhen Liu, Cewu Lu, Michael J Black, and Yao Feng. Chatgarment: Garment estimation, generation and editing via large language models. In *CVPR*, 2025. 2
- [4] Maria Korosteleva and Olga Sorkine-Hornung. Garmentcode: Programming parametric sewing patterns. *TOG*, 42(6):1–15, 2023. 1, 2
- [5] 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. In *ECCV*, 2024. 2
- [6] Kiyohiro Nakayama, Jan Ackermann, Timur Levent Kesdogan, Yang Zheng, Maria Korosteleva, Olga Sorkine-Hornung, Leonidas Guibas, Guandao Yang, and Gordon Wetzstein. Aiparel: A large multimodal generative model for digital garments. In *CVPR*, 2025. 2
- [7] Zhuo Su, Jiehua Zhang, Longguang Wang, Hua Zhang, Zhen Liu, Matti Pietikäinen, and Li Liu. Lightweight pixel difference networks for efficient visual representation learning. *TPAMI*, 45(12):14956–14974, 2023. 3
- [8] Feng Zhou, Ruiyang Liu, Chen Liu, Gaofeng He, Yong-Lu Li, Xiaogang Jin, and Huamin Wang. Design2garmentcode: Turning design concepts to tangible garments through program synthesis. In *CVPR*, 2025. 2

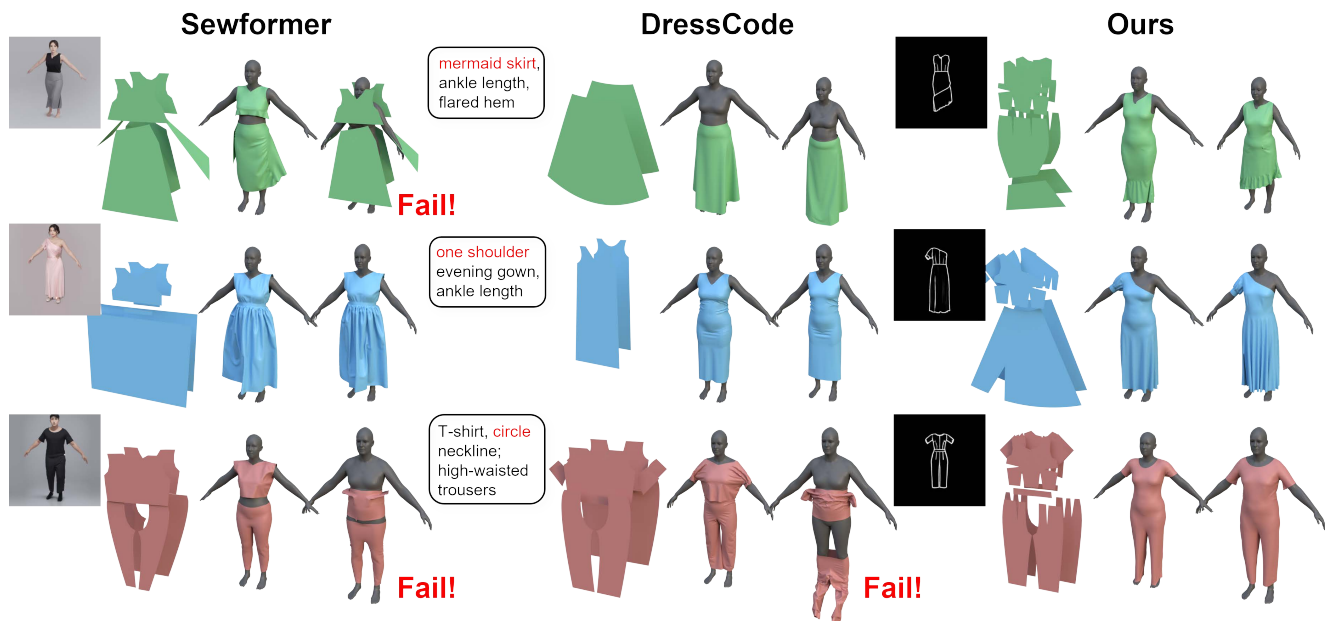


Figure 8. **Additional results for in-domain sketches.** We present the in-domain comparison with baseline methods.



Figure 9. **Additional results for in-the-wild sketches.** We also provide more results for in-the-wild sketches.

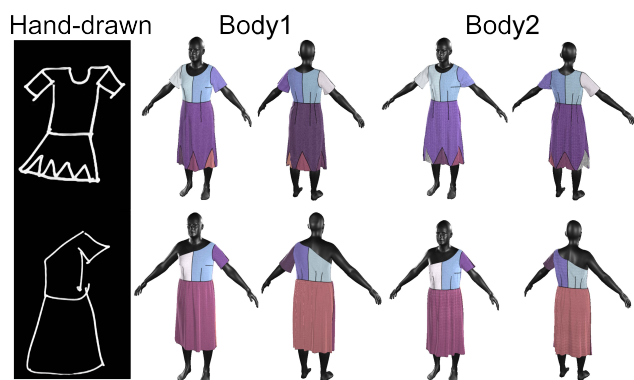


Figure 10. **Additional results for hand-drawn sketches.**

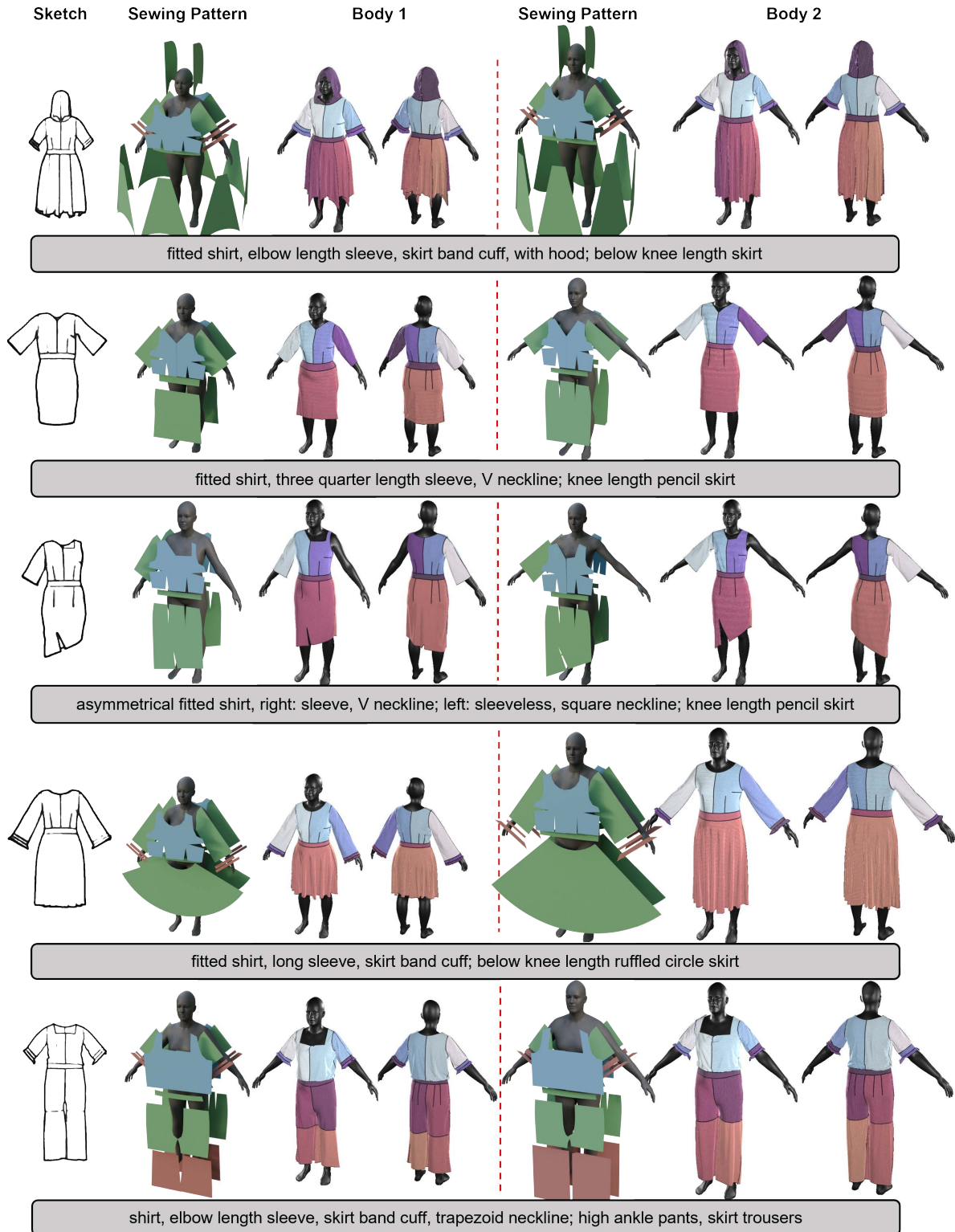


Figure 11. **Additional results for various body shapes.** We present identical garment designs tailored for two distinct body types, encompassing a spectrum of heights and body compositions, to demonstrate the effectiveness of our SewingLDM across diverse bodies.

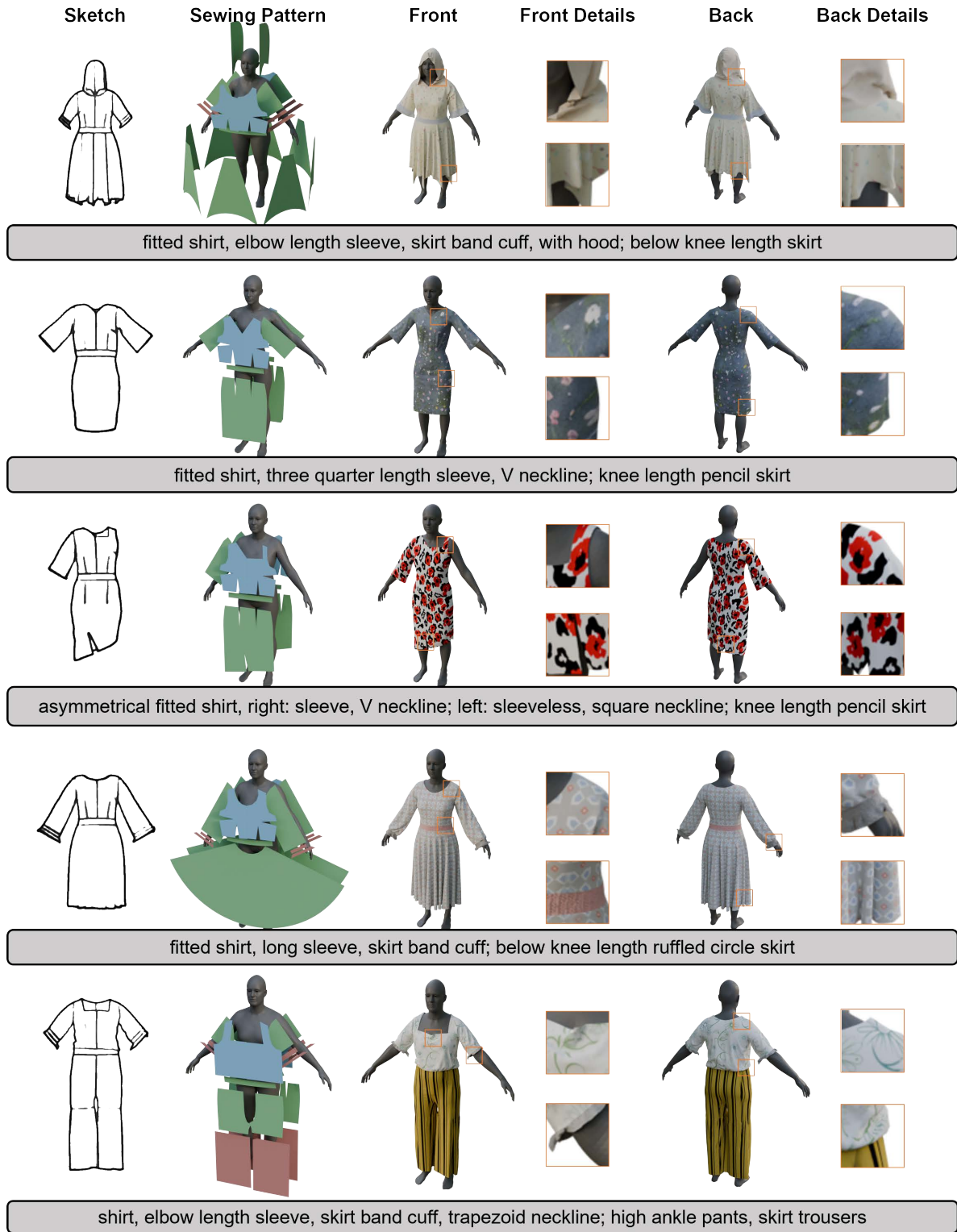


Figure 12. **Additional qualitative results.** By integrating Physically Based Rendering (PBR) textures, our generated outputs achieve visually compelling rendering effects, particularly for a wide range of intricate garment designs.