Supplementary Materials for: "Sketch2Stitch: GANs for Abstract Sketch-Based Dress Synthesis"

Faizan Farooq Khan[1], Eslam Abdelrahman Bakr[1], Davide Morelli[2,3],
Marcella Cornia[2], Rita Cucchiara[2], Mohamed Elhoseiny[1]
[1]King Abdullah University of Science and Technology, Saudi Arabia
[2]University of Modena and Reggio Emilia, Italy [3]University of Pisa, Italy
{faizan.khan, eslam.abdelrahman, mohamed.elhoseiny}@kaust.edu.sa
{davide.morelli, marcella.cornia, rita.cucchiara}@unimore.it

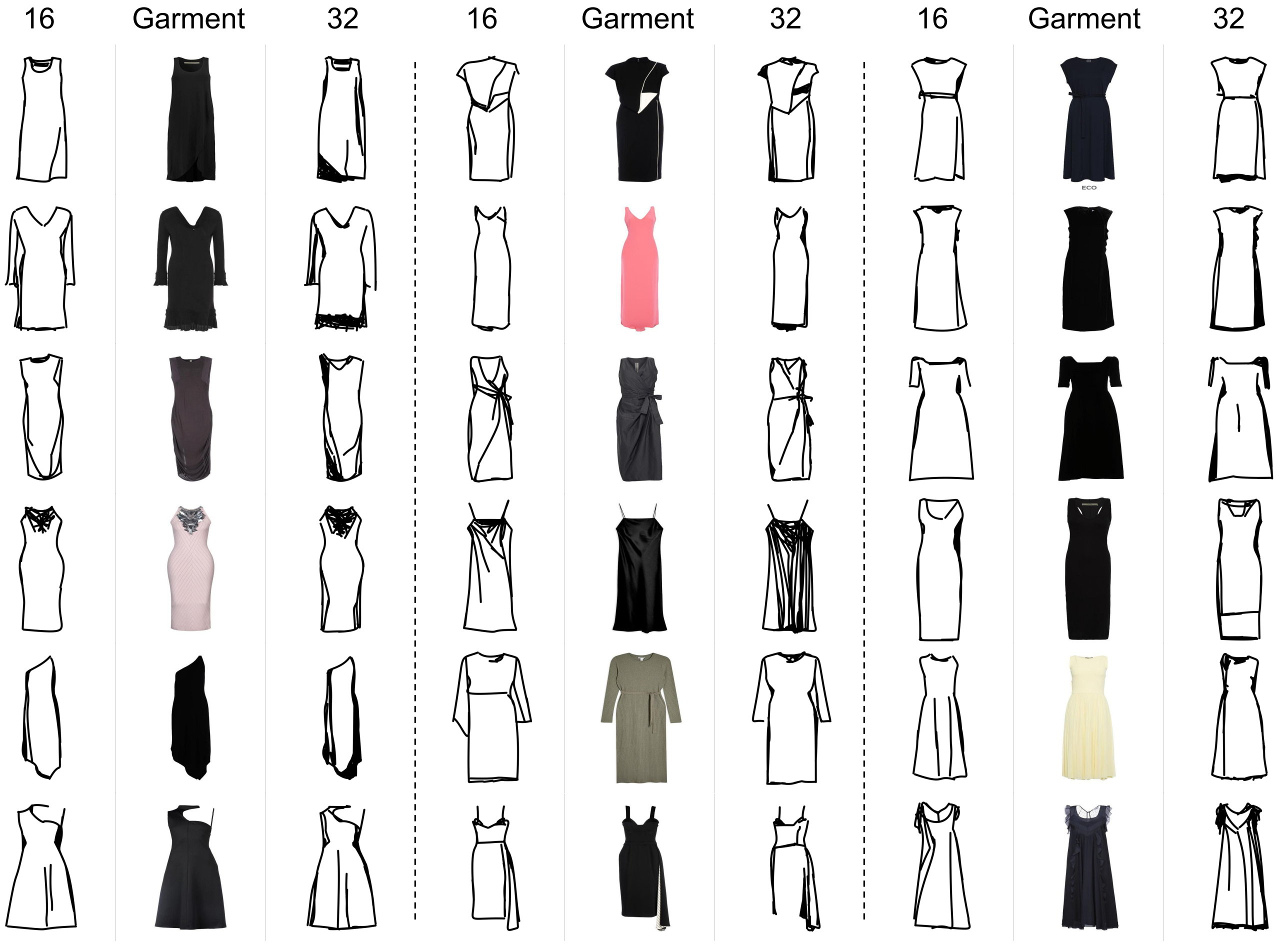

Figure 1: Samples from the Sketch2Stich-DressCode subset. The sketches on the left side of the garment consist of sixteen strokes and the ones on the right consist of thirty-two strokes. We can see that the sketches don't differ much and often sketches with thirty-two strokes use the extra strokes over already existing strokes making them redundant or making the boundaries thicker.

Appendix Overview

This supplementary document is organized as follows

- Section 1 showcases additional samples from our dataset for both sixteen and thirty strokes.
- Section 2 shows some more qualitative examples to compare our sketch-encoder with ControlNet.
- Section 3 showcases qualitative comparisons with our color-encoder and two strong baselines which are pSp and ViT-based.
- Section 4 showcases additional examples for color condition.
- Section 5 provides example generations for partial and noisy sketch inputs.
- Section 6 provides additional analysis of our sketches VS Edge-Maps.
- Section 7 discusses about some limitations of our work.
- Section 8 discusses some potential future work to overcome the limitations.

1. Sketch2Stitch Dataset

In the main paper, we mention that we use sketches with sixteen strokes due to their abstractness and due to thirty-two stroke-based sketches having redundant sketches. In Figure 1, we show samples from both 16 and 32-stroke-based sketches to show the difference between the two sketches. We can notice that most of the time, they are very similar, and sketches with thirty-two strokes end up using many strokes over existing ones, making them redundant. In our work, we specifically focus on abstract sketches drawn by an average user who is not good at sketching, so we chose sketches with sixteen strokes for all our training and testing processes.

2. Comparsion with ControlNet

In this section, we show qualitative examples where we compare our sketch encoder with ControlNet. We can from Figure 4 that while ControlNet generates realistic garments most of the time, it fails to properly condition the output on the sketches. As you can in some cases where the sketch input clearly depicts a half-arm t-shirt, ControlNet still generates t-shirts with full arms.

3. Color Encoder Baselines

In this section, we show qualitative examples where we compare our ResNet-based color encoder with pSp and ViT-based color encoder. We can from Figure 5 that our encoder is better at recognizing the fine details present in the input color strokes.

4. Color Encoder and Geometry Invariance

Our color encoder is trained to generate only w_{color} style vectors. For our color encoder, we do not use a hierarchical encoder. This is because, unlike sketch, the color modality does not contain information at multiple scales [2]. We show more qualitative samples from our color-based encoder in Figure 2 and Figure 6. The examples in both figures use color information as generated from Paint Transformer [3], which is why the color information is very similar to the garment's shape. To show that our color encoder is invariant to the geometry of the input, we provide our color encoder with input as shown in the top row of the Figure 3 and 7. This shows that the color can be chosen from a color palette, or the user can provide detailed color information according to their specific needs.

5. Partial Sketches

We show qualitative samples where we drop some strokes from the sketches and generate output from the partial sketches. The outputs are shown on the left of Figure 8. For the right part of Figure 8, we add random strokes to the input sketches to show that our model is also robust to noisy sketches. The generations show that our model can handle both partial and noisy sketches. This behavior can be credited to our abstract sketches, which can often be incomplete compared to edge maps. We show some examples that depict this behavior in Figure 9. We can see some sketches are not the same as edge maps, and the edge maps often miss out on pleats, whereas our sketches leave the base of the garment sometimes or do not close the arms of the garment. We discuss this further in Section 6.

6. Edge Maps

In Figure 9, we present a comparative analysis between examples derived from the Sketch2Stitch dataset and those generated using Pidinet [5]. We notice three main distinctions: 1) Pidinet tends to largely overlook pleats, a behavior expected given

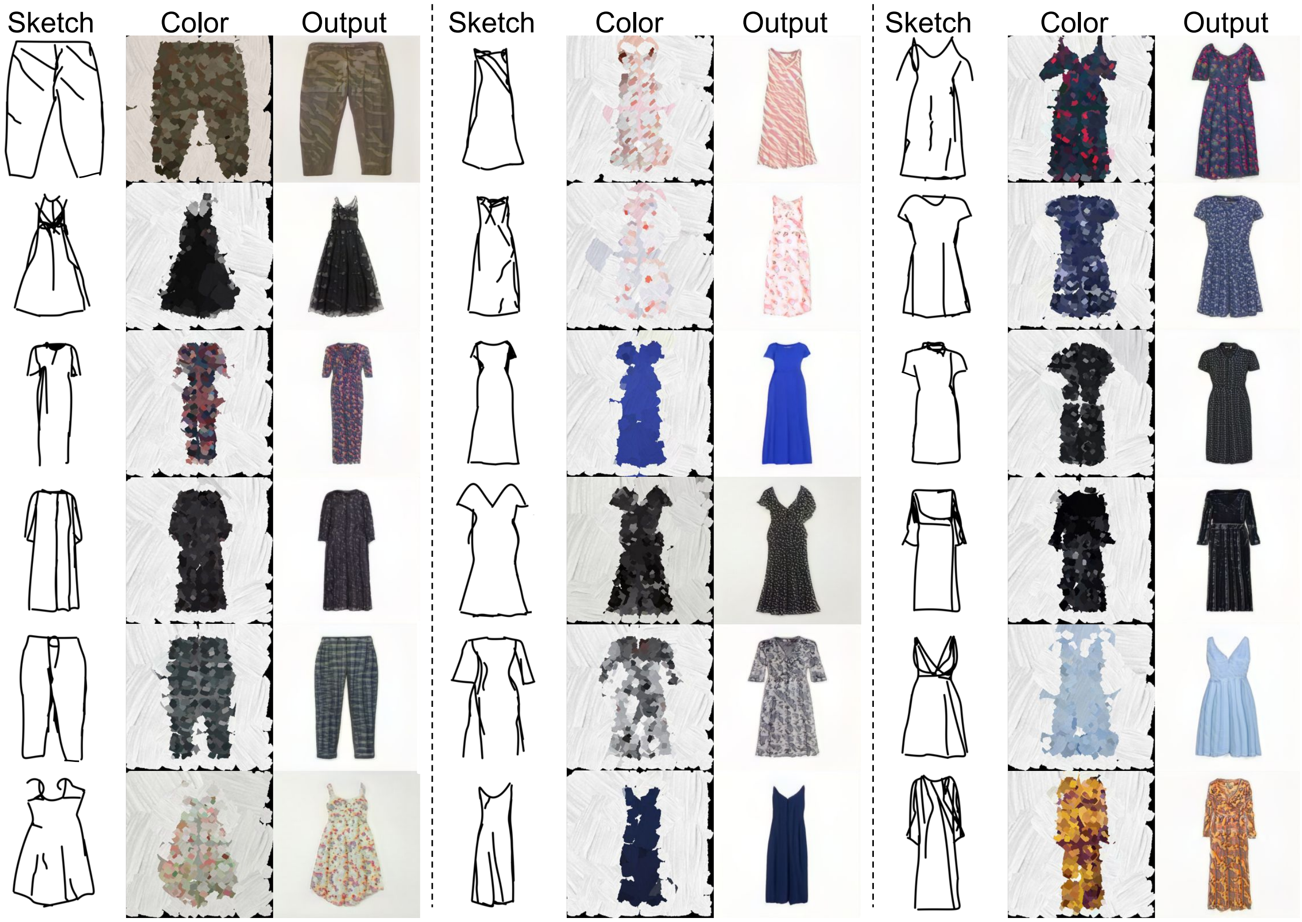

Figure 2. We present qualitative samples from Sketch2Stitch-DressCode subset wherein both sketch and color information are supplied to our network. Each row showcases three samples: each sample has three associated columns, the first column displays the sketch input, the second columns exhibits color strokes fed to color encoder, and the third column shows the output generated by our network.

their non-constituent nature to edges. 2) A variance in edge map intensities exists. Pidinet often produces faint edges, while our sketches exhibit uniform strength in the curves. We believe that this consistency in curve strength simplifies the construction of sketches, particularly for individuals with average sketching skills. 3) Our sketches embrace an abstract quality. For instance, curves like the closure of trouser pant legs at the bottom may not always be completely closed, in contrast to the edge maps. This deliberate abstraction enhances the model's capacity for generalization to diverse human sketching styles.

In Figure 10, we conduct a comparative evaluation of the two models. Both models share identical architectures and hyperparameters. One is trained on sketches from the /dataset dataset, while the other is trained on edge maps from Pidinet [5].

7. Limitations

While our model demonstrates proficiency in generating garment designs from the provided sketches, it bears certain limitations. Firstly, the model lacks awareness of texture information, as this detail is absent from the training data. Consequently, the generated garments may not capture textural nuances. Additionally, the model faces challenges when tasked with generating logos or written text on garments. This is due to CLIP [4] not being able to comprehend or incorporate detailed textual information from the garments into sketches. The model's capability is particularly constrained when producing highly detailed and specific elements, as the training dataset may not comprehensively encapsulate such intricate details. The generated dataset can also produce sketches of poor quality that do not accurately match the garment, which can harm the training process. Some cases are shown in Figure 11.

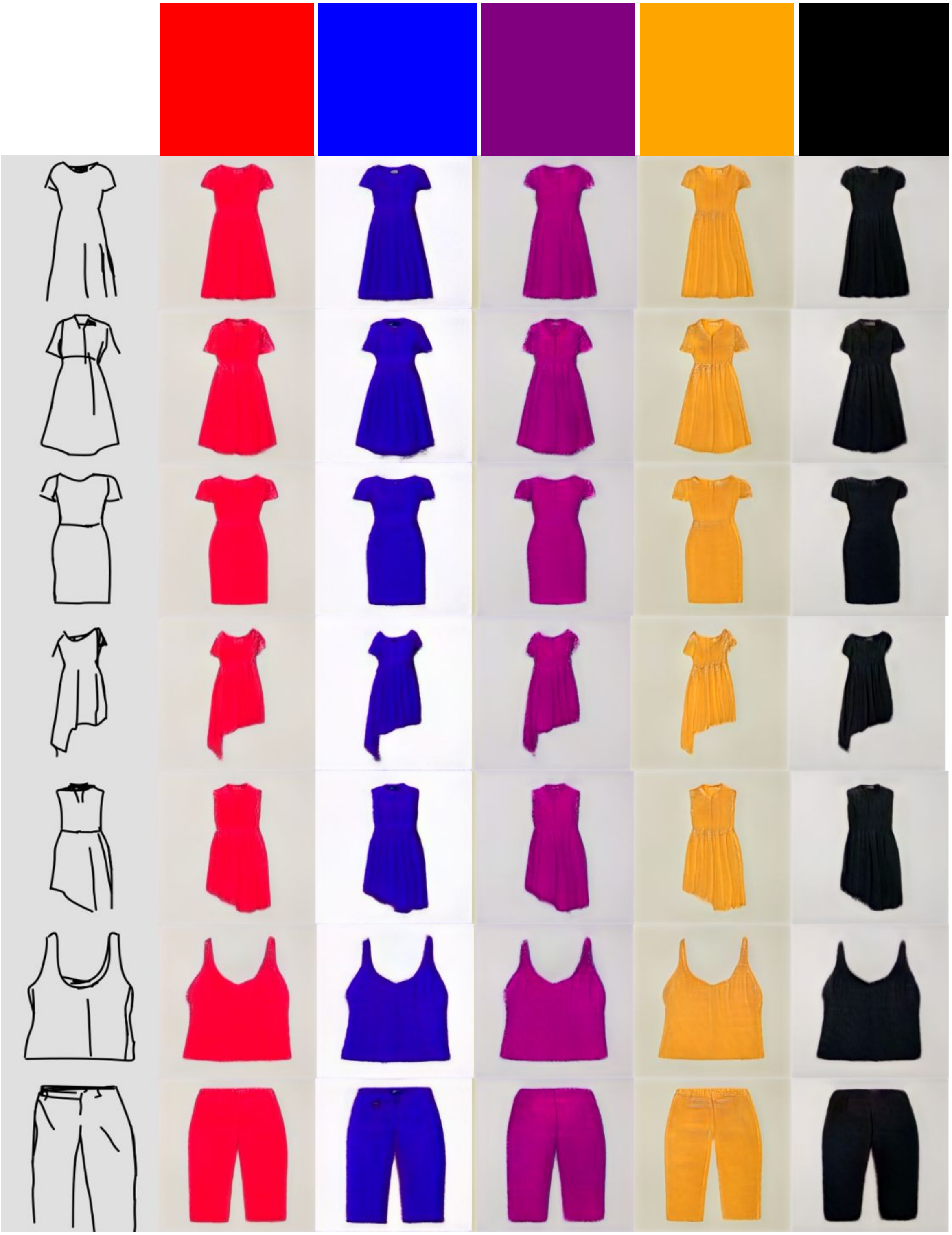

Figure 3. We demonstrate the geometry invariance of our color encoder by demonstrating samples from Sketch2Stitch-DressCode subset. The top row depicts the inputs to the color encoder, while the first column depicts the sketch which is processed by the sketch mapper. The following columns showcase the results for different colors.

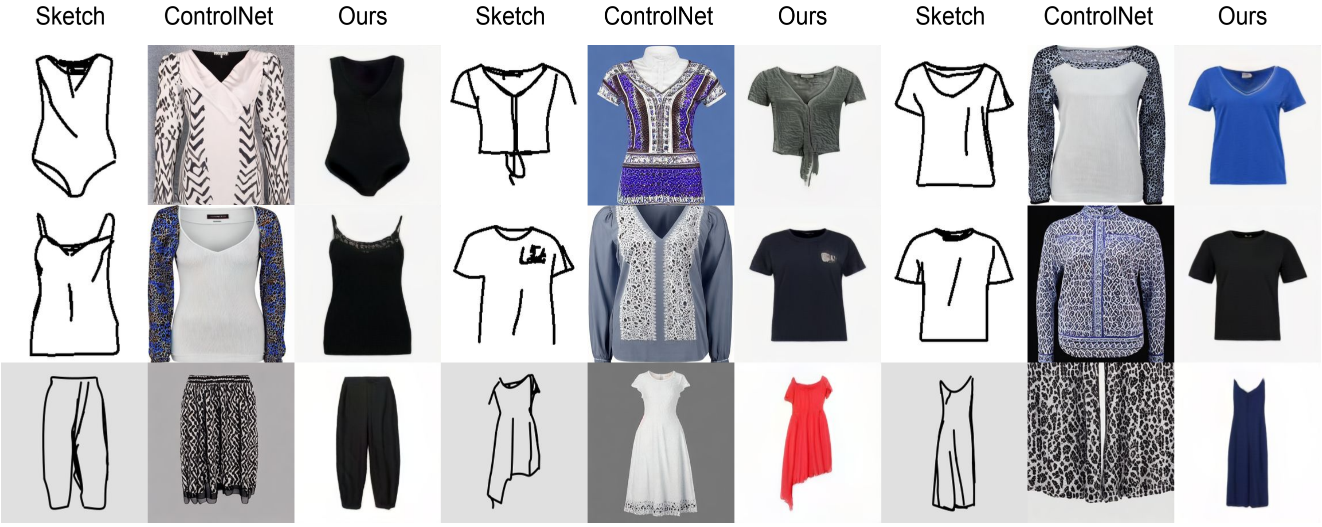

Figure 4. Qualitative Comparison of our sketch encoder with ControlNet.

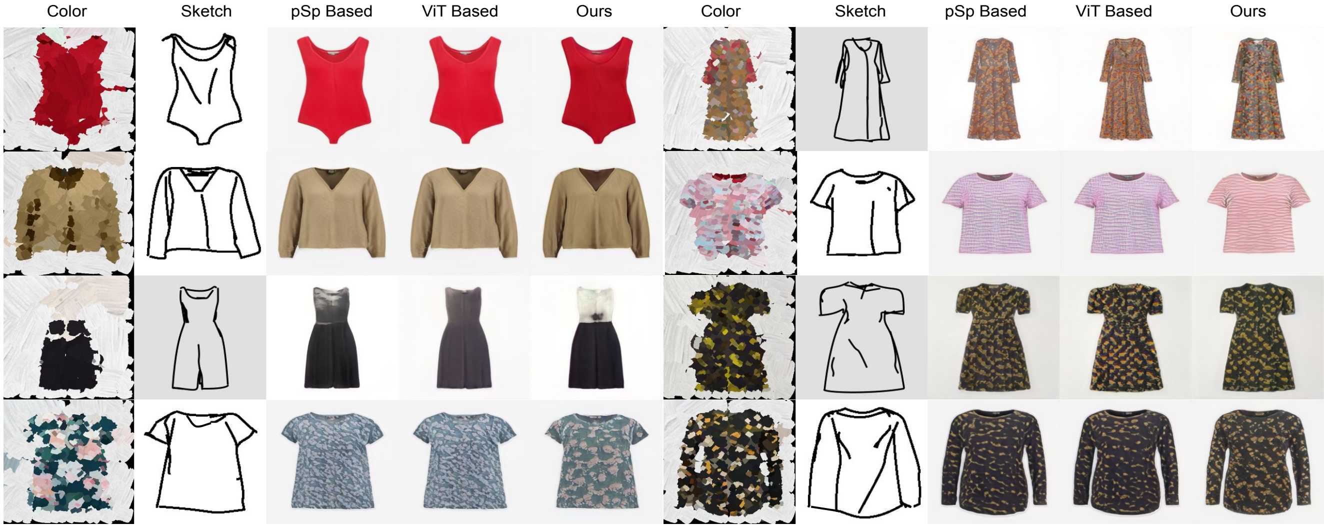

Figure 5. Qualitative Comparison with two strong color encoder baselines that we develop. We can see that our ResNet-based encoder is better at recognizing the fine-grained details from the input color strokes.

8. Future Work

In terms of future directions, a promising avenue for enhancement lies in addressing the current limitation of texture awareness within our garment generation model. Introducing a dedicated focus on texture generation or texture could significantly elevate the model's capacity to capture the intricate textures found in various fabrics faithfully. However, collecting textural information is a hard and laborious task. Recently, [1] collected a dataset with textural information for 20 pieces of clothing. The textural information also helps the visual quality of the generations. So, a large-scale texture-aware dataset can pave the way for more realistic and visually appealing garment designs.

Another critical aspect for future exploration is the incorporation of logo generation capabilities. A dedicated effort towards advancing logo generation, potentially through the integration of text-to-image synthesis techniques or improved handling of textual information within the model architecture, could substantially broaden the applicability of our garment generation system.

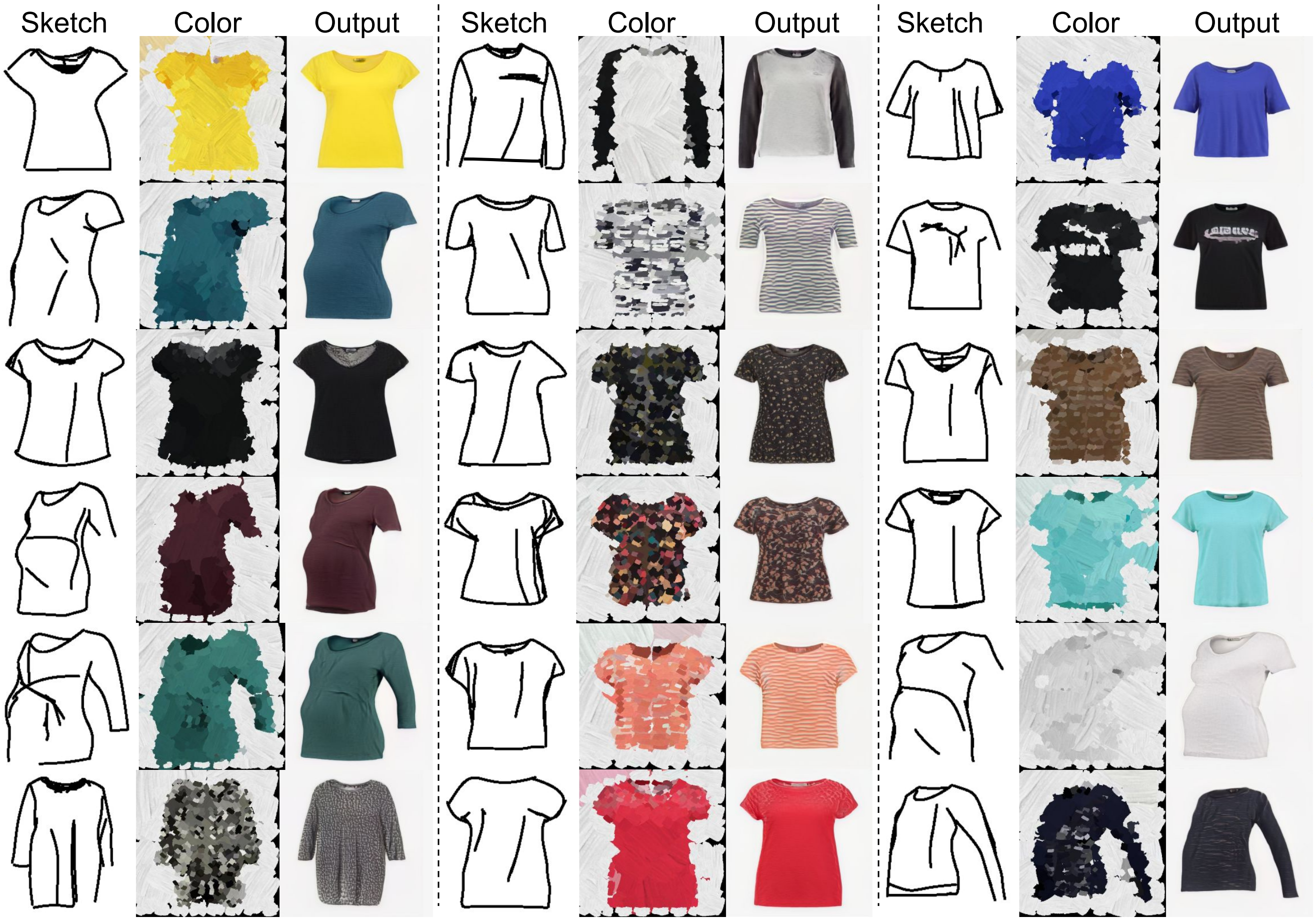

Figure 6. Similar to Figure 2, the above figure represents usage of our network for Sketch2Stitch-VITON-HD subset with color condition.

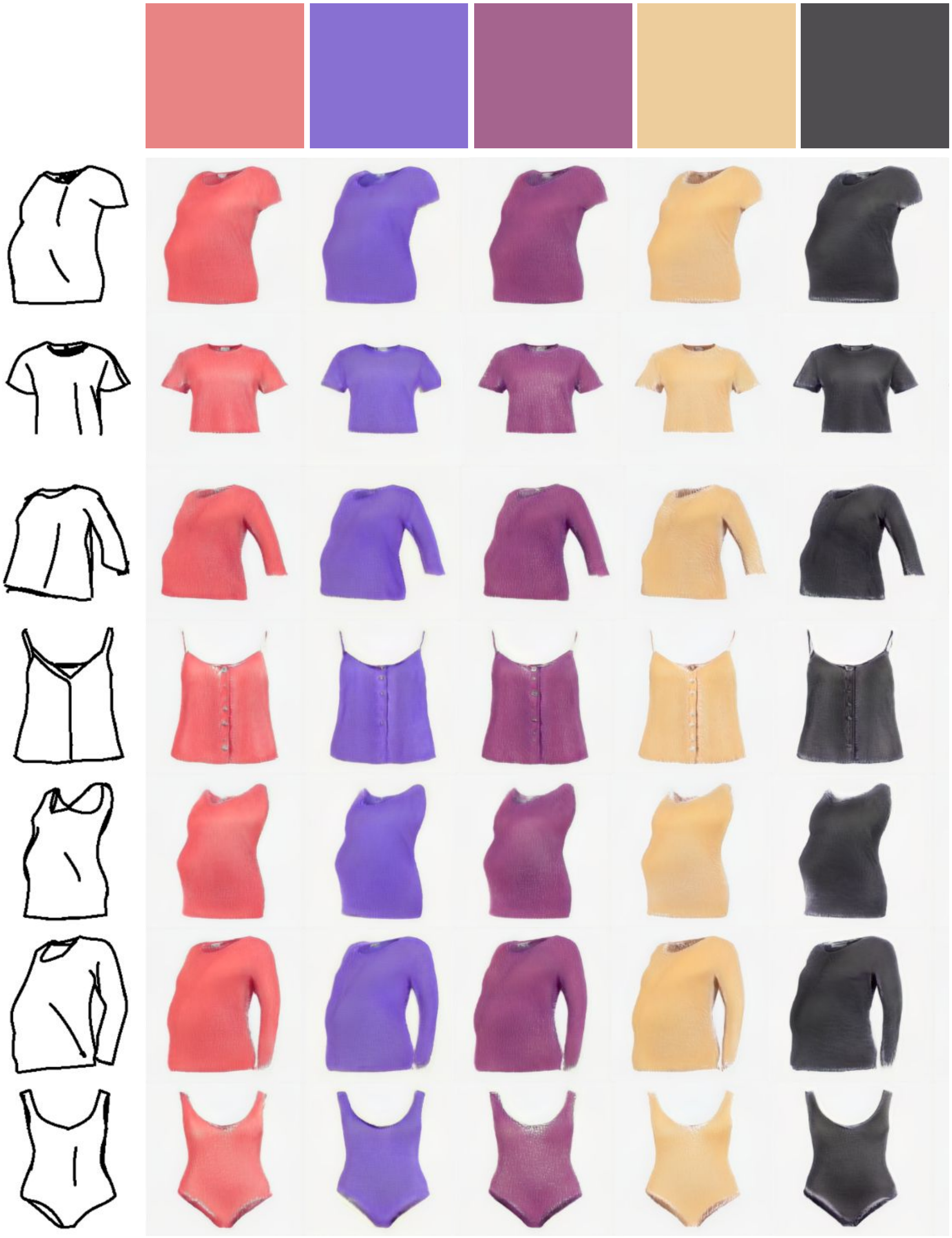

Figure 7. Similar to Figure 3, the above figure represents usage of our color encoder to show its geometry invariance for Sketch2Stitch-VITON-HD subset.

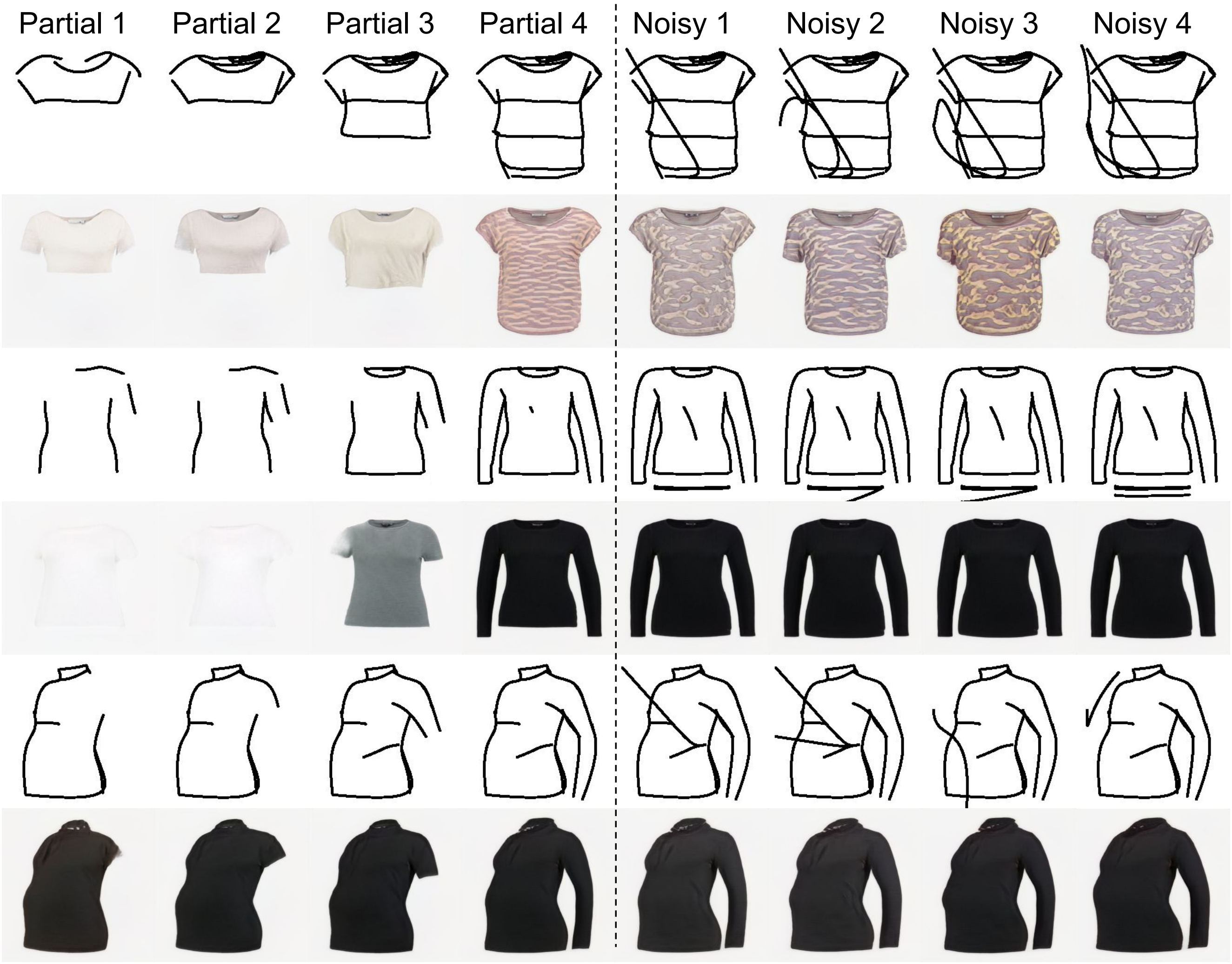

Figure 8. Left: Examples showing the generation from partial sketches. Right: Examples showing results when noisy strokes are added to the input. (For the second set of sketches in the partial category, the partial generations appear in white color, potentially making it challenging to see them clearly.)

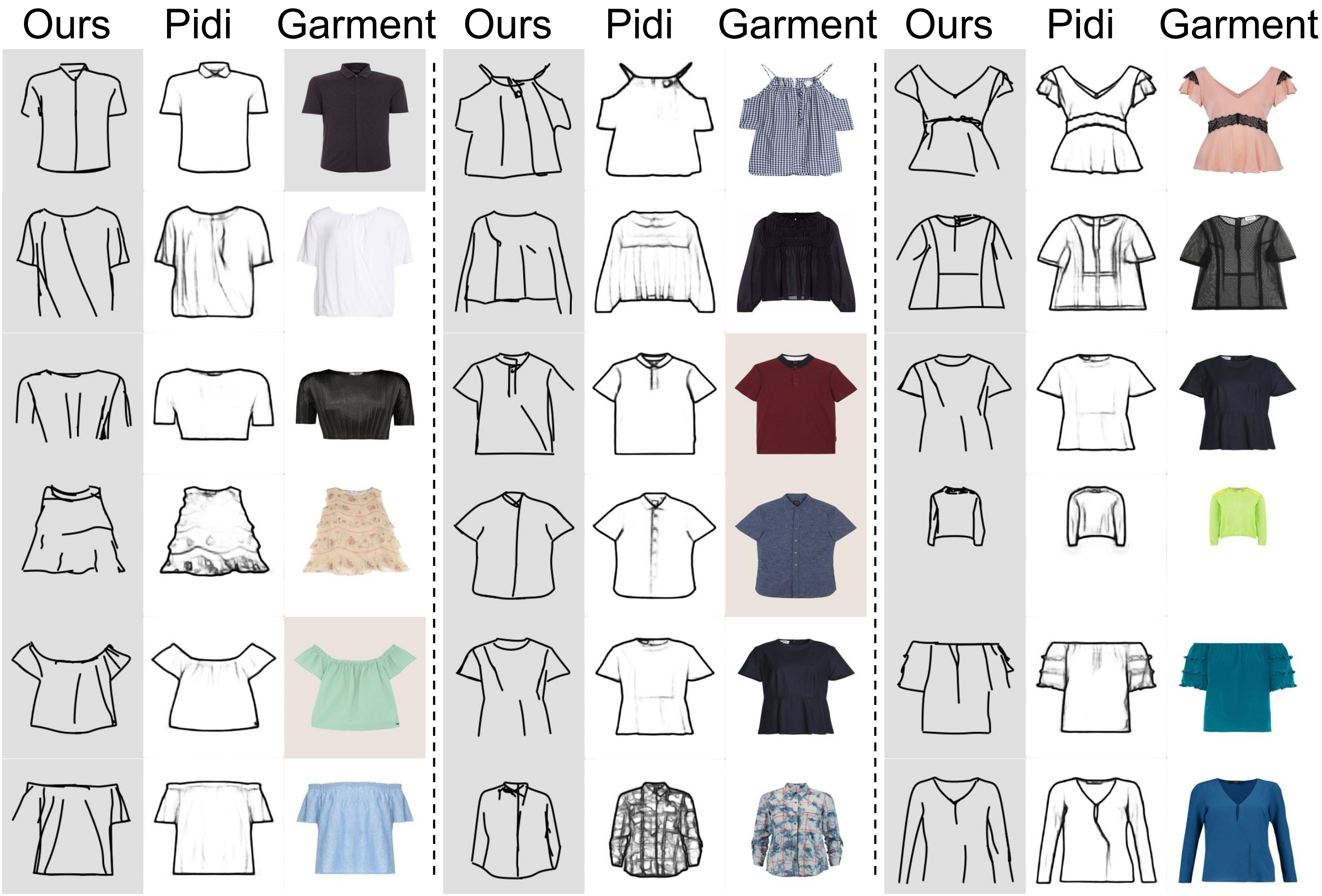

Figure 9. We show some examples of sketches from our Sketch2Stitch dataset and edge-maps generated from Pidinet [5].

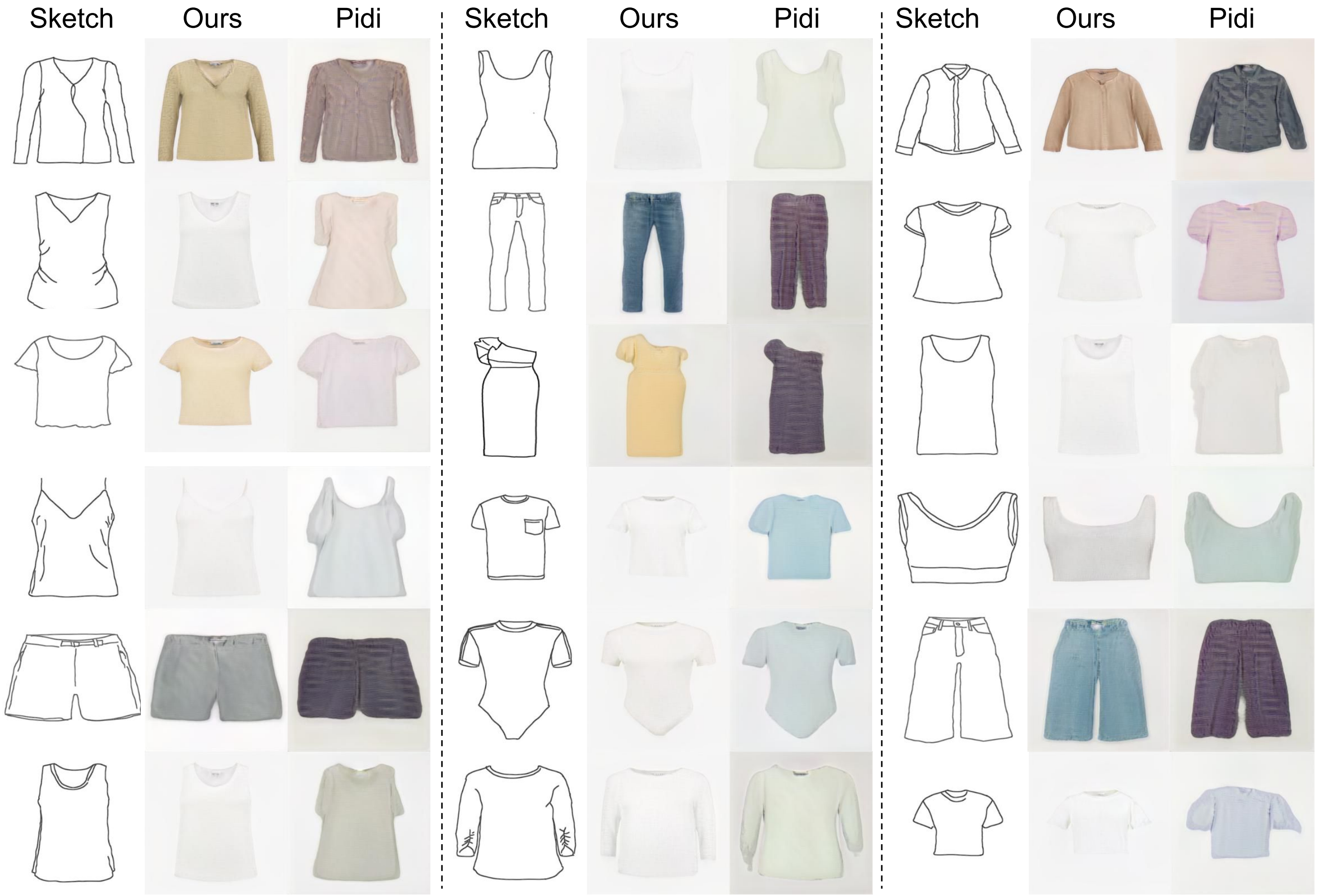

Figure 10. Examples showing the generations from our model, which is trained on sketches from Sketch2Stitch dataset and from the same model but trained on edge maps obtained from Pidinet [5].

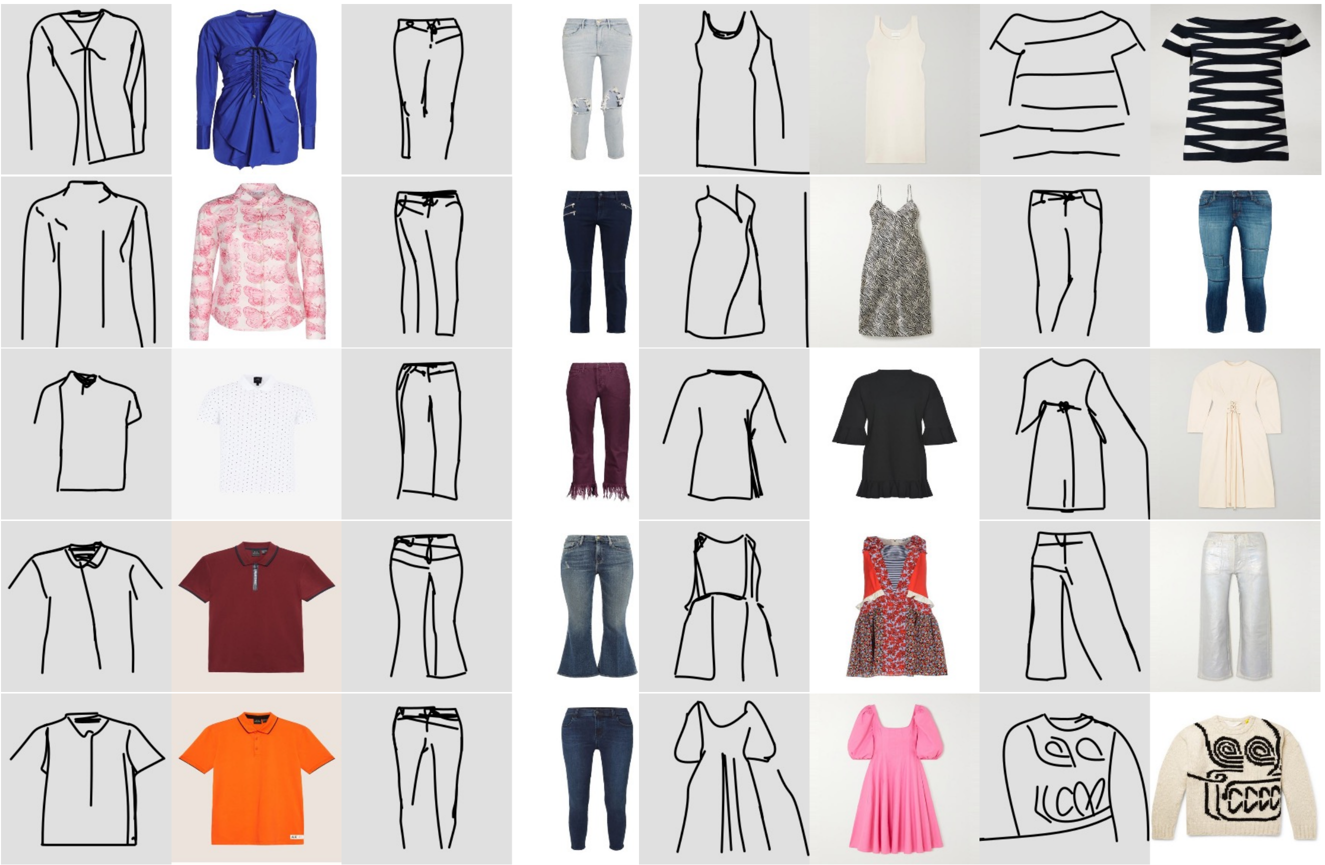

Figure 11. Examples of failed sketch-garment pairs from our generated dataset. These sketches are poorly aligned with the corresponding garment, often due to missing strokes or wrong stroke placement.

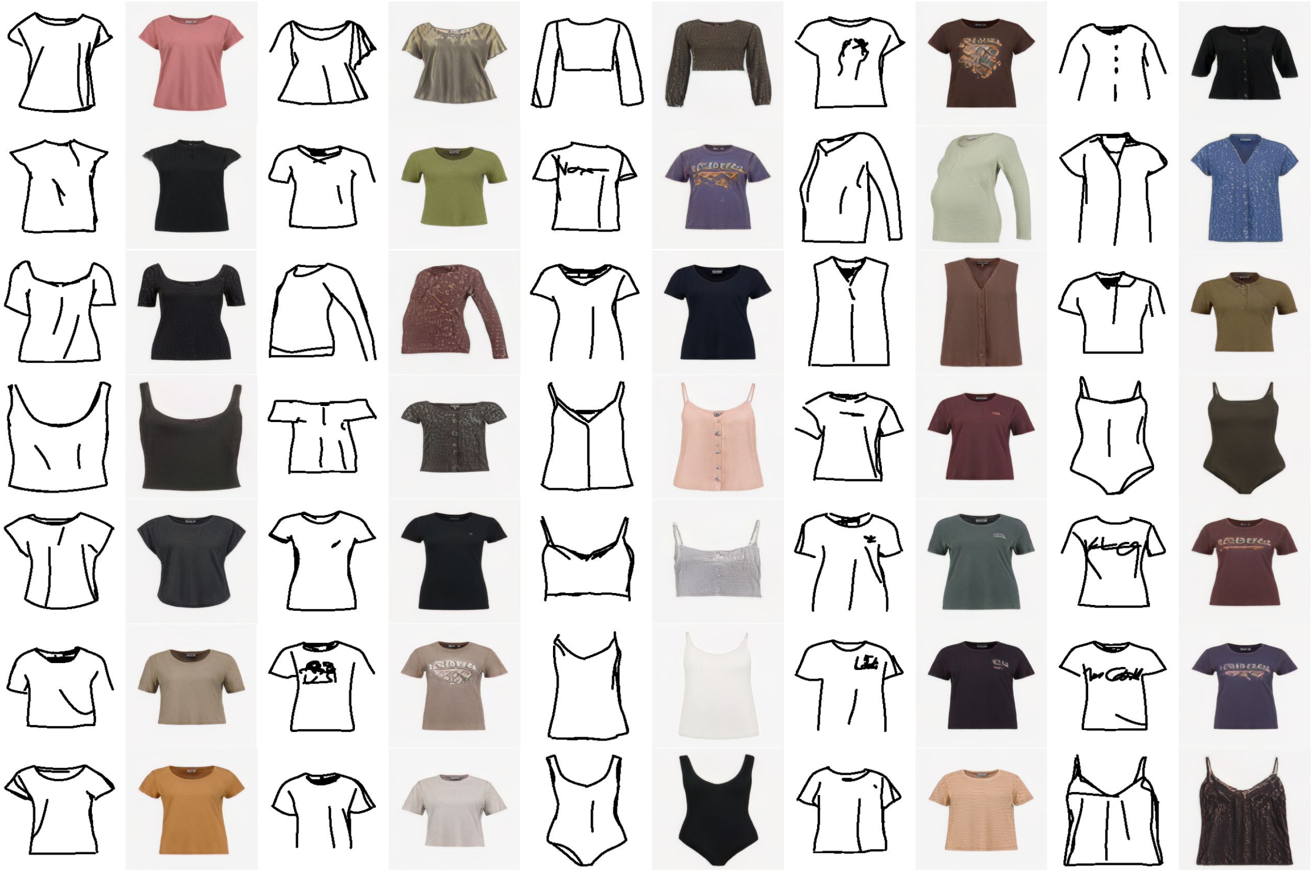

Figure 12. Additional qualitative examples from Sketch2Stitch-VITON-HD subset for sketch-to-garment task.

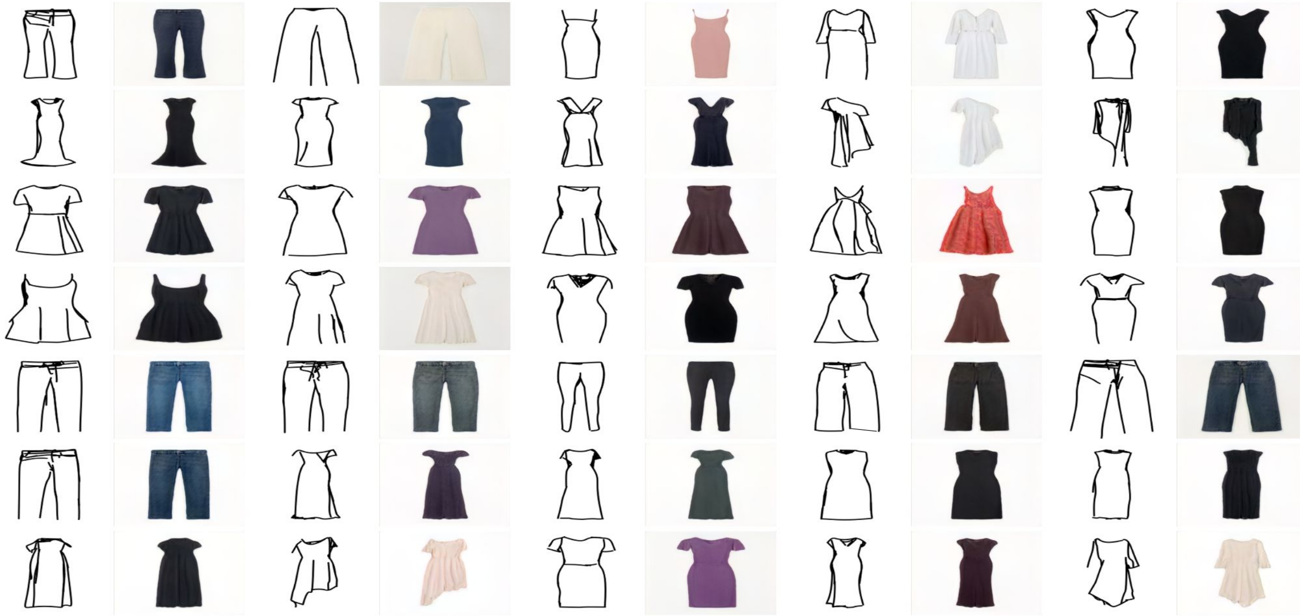

Figure 13. Additional qualitative examples from Sketch2Stitch-DressCode subset for sketch-to-garment task.

References

[1] Ruihan Gao, Wenzhen Yuan, and Jun-Yan Zhu. Controllable visual-tactile synthesis, 2023. 5

[2] Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Multimodal conditional image synthesis with product-of-experts gans, 2021. 2

[3] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint transformer: Feed forward neural painting with stroke prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2

[4] 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. 3

[5] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikainen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5117–5127, 2021. 2, 3, 9, 10