Garment Ideation: Iterative View-Aware Sketch-Based Garment Modeling Supplemental

Pinaki Nath Chowdhury[1, 3] Tuanfeng Wang[3] Duygu Ceylan[3] Yi-Zhe Song[1]
Yulia Gryaditskaya[1,2]

[1]SketchX, CVSSP, University of Surrey, United Kingdom
[2]Surrey Institute for People Centred AI and CVSSP, University of Surrey, United Kingdom
[3]Adobe Research, London, UK

S1. Comparing implicit shape representations

In this section, we compare two implicit shape representations: Unsigned Distance Fields (UDFs) [2] and Generalized Winding Numbers (GWNs) [3]. The visual comparison is shown in Figure S2.

To perform this comparison, we first convert a 3D shape to each implicit shape representation, calculated on the points of the $256 \times 256 \times 256$ spatial grid. (Please note that during training, we evaluate the prediction results at the points of the $128 \times 128 \times 128$ spatial grid.) We then convert a shape in each implicit representation back to a 3D mesh. In particular, for UDFs, we obtain a 3D mesh using a marching cubes algorithm with a threshold value set to 0.005. For GWNs, we obtain a 3D mesh by *first computing GWNs gradients* followed by a marching cubes algorithm with a threshold set to 0.3 [1]. The threshold values are selected empirically. For UDFs, too large threshold values result in thick double surfaces, while too small threshold values result in appearance or visible holes in the reconstructed 3D geometry. For GWNs, too large threshold values result in appearance or visible holes in the reconstructed 3D geometry, and too small threshold values result in thick double surfaces.

From Figure S2, we observe that the reconstructed 3D shapes from UDFs often contain closed surfaces in open regions, such as the 'collar' region or 'sleeves' openings. While the reconstructed 3D shapes from GWNs miss some fine-details (e.g., less detailed reconstruction of the 'belt' region in Figure S2), they exhibit open surfaces like the 'collar' region. This was previously observed by Chi et al. [1].

S2. PyTorch-like pseudo-code for Binarisation and Feature Blending

We will release our code upon acceptance. In addition, Algorithm 1 presents a PyTorch code for binarisation of blending weights $\alpha^i \in \mathbb{R}^{512}$, implementing (1) Equation (1) with $\lambda = 1$, and (2) the feature blending step in Equation (3), that combines our binary activation α^i_* and the latest sketch feature f^i_{align} with the merged feature obtained in the previous step $\tilde{f}^{i-1}_{align}$.

S3. Qualitative comparison of iterative behavior

Table 4, in the main document, shows that our feature fusion strategy is by far the most efficient in updating the 3D geometry with respect to the most recent sketch view.

In Figure S4, we provide qualitative comparison with alternative feature fusion strategies, described in the main document: **B-RNN**, **B-Concat**, and **B-Cont-**α.

S4. Additional result on different datasets

We present qualitative results in Fig. S3 on Wang et al. [5] in the following figure. Although we evaluate on [5], a comparison of their proposed method with ours is unfair since: (a) [5] only works for a predefined garment category and lacks generalization (b) [5] does not utilize any feature aggregation strategy to handle multiple views (c) the dataset in [5] does not include part-information (i.e., annotation for "sleeves" etc.) to support the augmentation strategy used in our training to support our goal of iterative design evolution.

S5. Inference time or latency

Generating the underlying garment geometry from sketches includes two stages of latency: (a) Predicting the

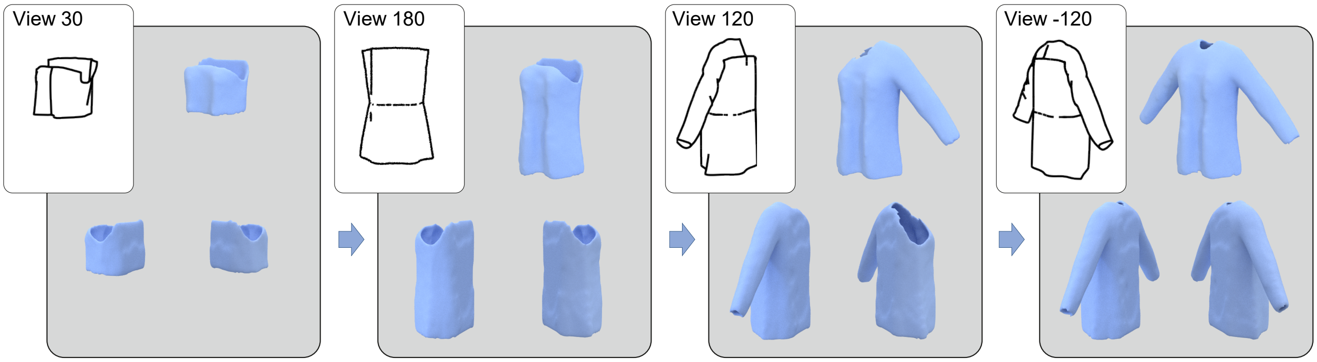

Figure S1. Alternate UI of our proposed system to support iterative multi-view garment design. In this example, the user starts by sketching in an arbitrary chosen view, then rotates the reconstructed garment to a back view and changes the design by augmenting it with a skirt. Exploring the design further, yet from different viewpoints, the user adds first one sleeve and then another. Our system efficiently updates the prediction, carefully matching it to the most recent view.

implicit representation of geometry via our model. This step takes 1.24s. (b) Following the *standard procedure* of extracting a mesh from a predicted implicit representation using Marching Cubes [4]. This takes 0.97s. Therefore, the total latency is 2.21s.

For real-time performance this latency can be further reduced by paralleling the computation into multiple GPUs. Exploring alternative garment representation may also help in the long term.

Algorithm 1: PyTorch code to compute strictly binary signal in Eq.2 and Feature Blending in Eq.3

```
import torch
import torch.nn.functional as F

# Compute strictly binary signal
def binarise(x):
    # x:  Continuous α shape [nbatch, m]
    x = F.hardshrink(x, lambda=1.0)
    x = F.hardtanh(x, min_val=0, max_val=1)
    return x

# Compute Feature Blending
def blending(aligned, alpha):
    # aligned:  List of [f^0_align, f^1_align, ..., f^N_align]
    # alpha:  List of [α^0, α^1, ..., α^N]

    N = len(aligned) # Length of List f^i_align
    combined = aligned[0]

    # Iterate 1 to N-1; uses 0-th indexing
    for idx in range(1, N):
        f = aligned[idx] # select f^i_align
        a = alpha[idx] # select α^i
        b = binarise(a) # Binarise to 0 or 1

        # Feature Blending Step
        combined = b*f + (1 - b)*combined
    return combined
```

References

[1] Cheng Chi and Shuran Song. Garmentnets: Category-level pose estimation for garments via canonical space shape completion. In *ICCV*, 2021. 1

[2] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *NeurIPS*, 2020. 1, 3

[3] Alec Jacobson, Ladislav Kavan, and Olga Sorkine. Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Graph.*, 32(4), 2013. 1, 3

[4] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 2

[5] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.*, 2018. 1, 3

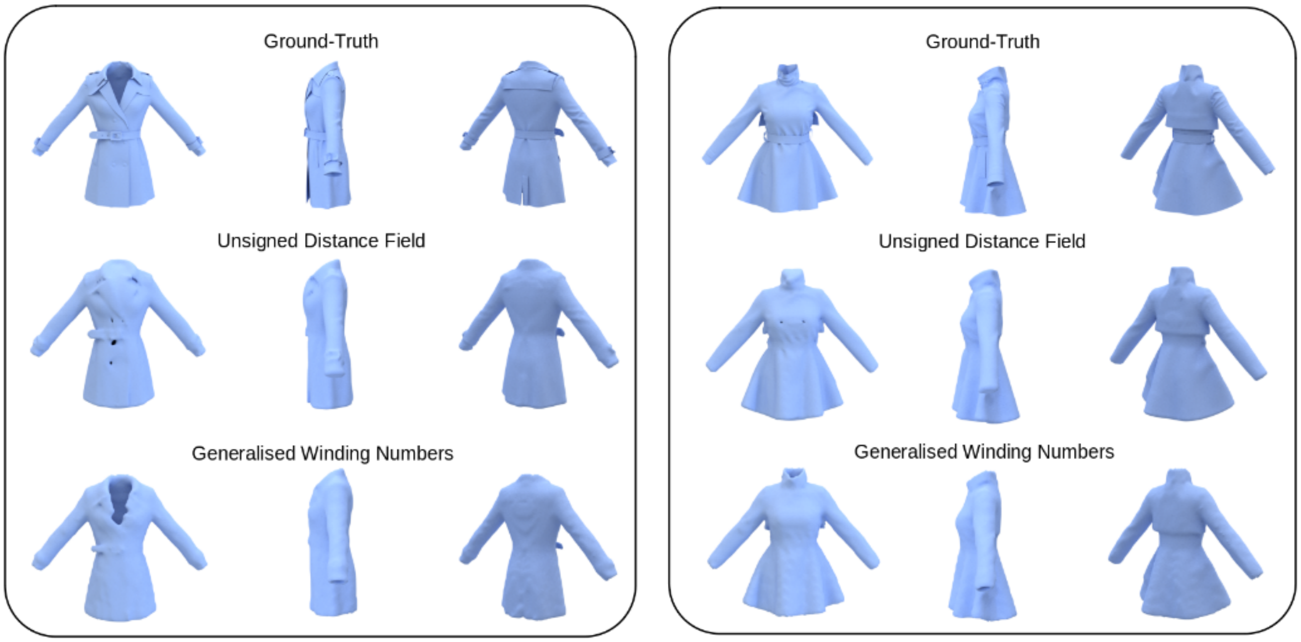

Figure S2. Qualitative comparison of the two implicit representations: Unsigned Distance Fields (UDFs) [2] and Generalized Winding Numbers (GWNs) [3]. For each rectangle: In the first row, we show the ground-truth geometry. In the second row, we show the reconstructed 3D shape from UDFs. In the third row, we show the reconstructed 3D shape from GWNs. See Section S1 for the details.

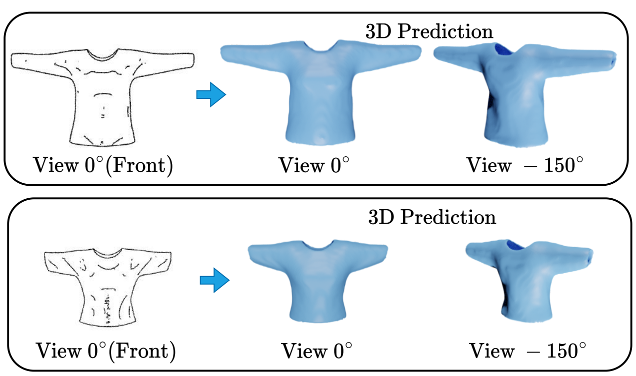

Figure S3. Qualitative results of our proposed method on Wang *et al*. [5]. We omit our data augmentation strategy described in Section 4 since [5] does not include part-information (i.e., annotation for "sleeves" etc.). This limits our goal of iterative design.

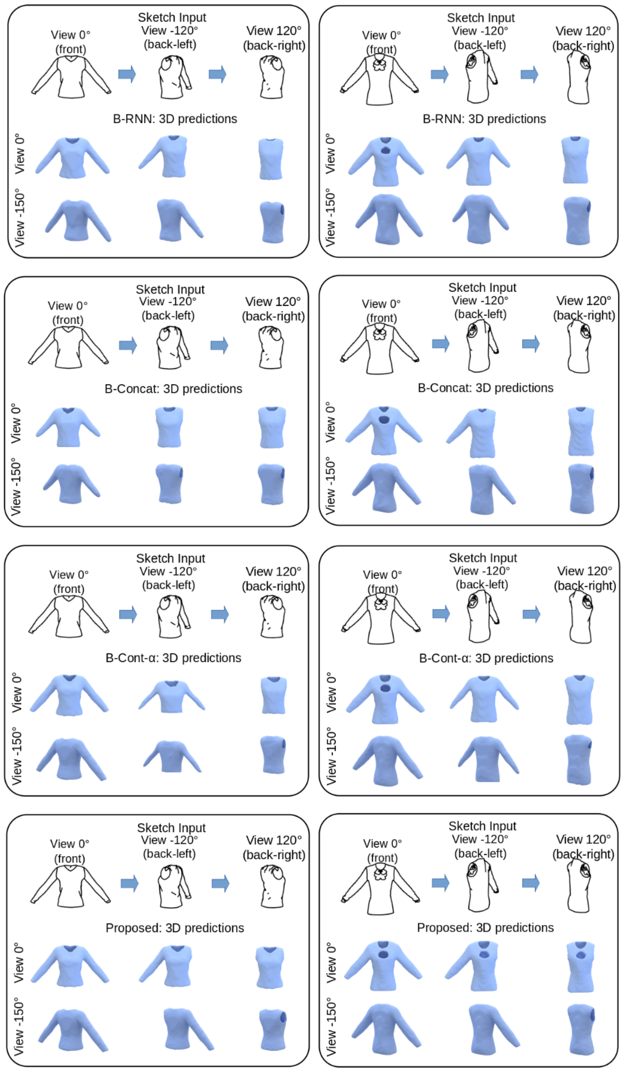

Figure S4. View-disentanglement for iterative design editing (see Section 4.2.3 in the main document) on 2 representative garments. We compare our method to **B-RNN**, **B-Concat**, and **B-Cont-α** baselines, described in the main document. It can be observed that (1) **B-RNN**, **B-Concat** and **B-Cont-α** 'forget' the details sketched in the earlier views, and not visible in the most recent views; (2) In addition, **B-Cont-α** can not accurately take into account edits to a garment design; (3) Our method not only updates 3D geometry with respect to the most recent sketch views (e.g. removing a sleeve), but also retains information from earlier sketches (the 'collar' region does not change when editing a 'sleeve' in the complementary views).