In Proceedings of the International Conference on Computer Vision (ICCV), 2019

Fashion++: Minimal Edits for Outfit Improvement (Supplementary File)

Wei-Lin Hsiao[1,4] Isay Katsman[2,4*] Chao-Yuan Wu[1,4*] Devi Parikh[2,4] Kristen Grauman[1,4]
[1]UT Austin [2]Cornell Tech [3]Georgia Tech [4]Facebook AI Research

This supplementary file consists of:

- Implementation details of the complete Fashion++ system presented in Section 4 of the main paper
- Ablation study on our outfit's representation (referenced in Section 3.2 of the main paper)
- Details on shape generation
- More details on the automatic evaluation metric defined in Section 4.1 of the main paper
- More examples of Fashion++ edits
- MTurk interfaces for the three human subject studies provided in Section 4.2 of the main paper
- Full results and Turkers' verbal rationales (as a word-cloud) for user study A (Section 4.2 of the main paper)
- Examples of Turkers' verbal descriptions of what actions to perform in user study C (Section 4.2 of the main paper)

I. Implementation details

Training. We have two generators, a GAN for texture and a VAE for shape, and a classifier for editing operations. All generation networks are trained from scratch, using the Adam solver [1] and a learning rate of 0.0002. For VAE, we keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 200 epochs. For GAN, we keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs. For the fashionability classifier, we train from scratch with the Adam solver with weight decay 0.0001 and a learning rate of 0.001. We keep the same learning rate for the first 60 epochs and decay it 10 times every 20 epochs until epoch 120.

For the GAN, we adopt the architecture from [3]. For the VAE, our architecture is defined as follows: Let `c7s1-k` denote a 7×7 convolutional block with k filters and stride 1. `dk` denotes a 3×3 convolutional block with k filters and stride 2. `Rk` denotes a residual block that contains two 3×3 convolutional blocks with k filters. `pk` denotes a layer reflection padding 3 on all boundaries. `fck` denotes a fully connected layer with k filters. We use Instance Normalization (IN) [2] and ReLU activations. The VAE consists of:

- Encoder: `p3`, `c7s1-16`, `d32`, `d64`, `d128`, `d128`, `d128`, `d128`, `d128`, `R128`, `R128`, `R128`, `R128`, `R128`, `R128`, `R128`, `R128`, `R128`, `fc8`
- Decoder: `d128`, `d128`, `d128`, `d128`, `d128`, `d64`, `d32`, `d16`, `p3`, `c7s1-18`

where the encoder is adapted from [3] and decoder from [4]. Our MLP for the fashionability classifier is defined as: `fc256`, `fc256`, `fc128`, `fc2`. For shape and texture features, both d_s and d_t are 8. For the fashionability classifier to perform edits, we use an SGD solver with step size 0.1.

Baselines. Since the encodings' distribution of inventory garments is not necessarily Gaussian, the RANDOM baseline samples from inventory garments for automatic evaluation, and from a standard Gaussian for human subject study B.

Post-processing. As our system did not alter clothing-irrelevant regions, and to encourage viewers to focus on clothing itself, we automatically replace the generated hair/face region with the original, using their segmentation maps.

II. Ablation study

We use $d_s, d_t = 8$ throughout our paper. Here, we show the effect of texture and shape feature on their own, and how the dimension of the feature affects our model. We measure the feature's effect by the fashionability classifier (MLP)'s validation accuracy. We compare just using texture, just using shape, and using the concatenation of the two in Tab. 1(a): we found that shape is a more discriminative feature than texture. We tried $d_t = 3, 8$, and found that $d_t = 8$ gives qualitatively more detailed images than $d_t = 3$, but continuing increasing d_t beyond 8 does not give qualitatively better result. Tab. 1(b) shows the feature dimension's effect on the quantitative results, where left is

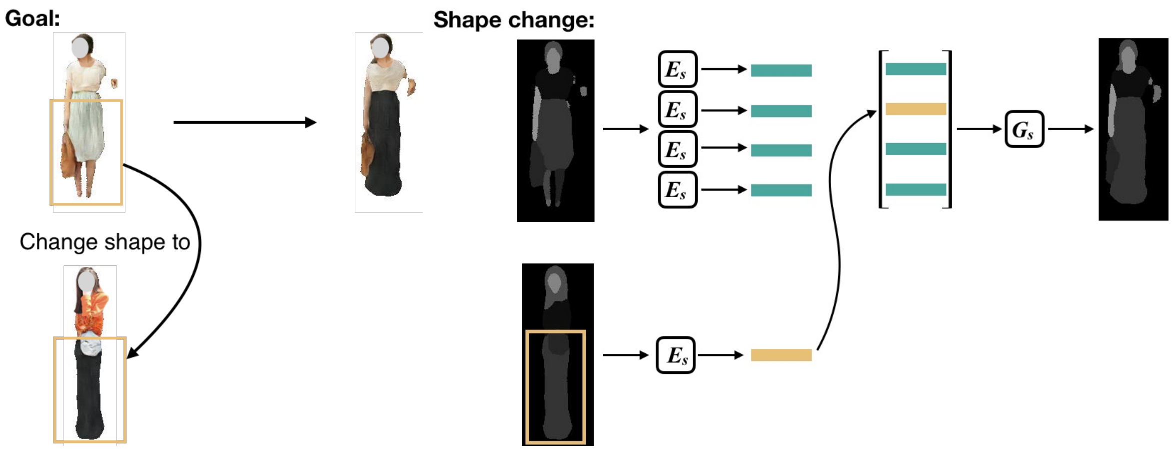

Figure 1: Shape generation using our shape-VAE. In this example, the goal is to change the girl's midi skirt to a long skirt. We encode each garment separately, overwrite the skirt's code with the code from the long skirt, and generate the final changed silhouette for the outfit.

texture	shape	texture + shape
0.663	0.741	0.751

(a) **Feature selection.**

texture		texture + shape	
3	8	3	8
0.576	0.663	0.717	0.751

(b) **Feature dimension.**

Table 1: Ablation study on how the outfit's features affect the accuracy of the fashionability classifier.

just using the texture as the feature and right is concatenating both texture and shape feature. In both cases, increasing d_t makes our features more discriminative.

III. More details about shape generation

Here, we walk through the process of how our shape generator controls the silhouette of each garment. If our goal is to change an outfit's skirt, as in Fig. 1 left, our shape encoder E_s first encodes each garment separately, and then overwrites the skirt's code with the skirt we intend to change to. Finally, we concatenate each garment's code into $\mathbf{s} = [\mathbf{s}_0; \ldots; \mathbf{s}_{n-1}]$, and our shape generator G_s decodes it back to a region map. This process is shown in Fig. 1 right.

IV. Automatic evaluation metric

To automatically evaluate fashionability improvement, we need ground-truth (GT) garments to evaluate against. To capture multiple ways to improve an outfit, we form *a set of GT garments* per outfit, as noted in Section 4.1 of the main paper. Our insight is that the garments that go well with a given blouse appear in outfits that also have blouses similar to this one. As a result, we take the corresponding region's garments, that is the pants or skirts worn with these similar blouses, to form this set. To do so, we first find the M nearest neighbors of the unfashionable outfit *excluding the swapped out piece* (Fig. 2 left), and then take the corresponding pieces in these M neighbors (Fig. 2 right) as M

Figure 2: Formulating the set of GT garments per negative outfit.

possible ways to make this outfit fashionable. We use the minimal distance of the original piece to all K pieces in GT set as the original piece's distance to GT. Using median or mean gives similar results.

V. More qualitative examples

Due to the sake of space, we show one Fashion++ edit for each example in Section 4.3 of the main paper. In Fig. 3, we show more editing examples by Fashion++, and for each one we display the editing spectrum from $K = 1$ to 6. Fig. 3(a) is the full spectrum for one of the examples in Fig. 6 of the main paper. The outfit starts changing by becoming sleeveless and tucked in, and then colors become even brighter as more edits are allowed. (b) changes

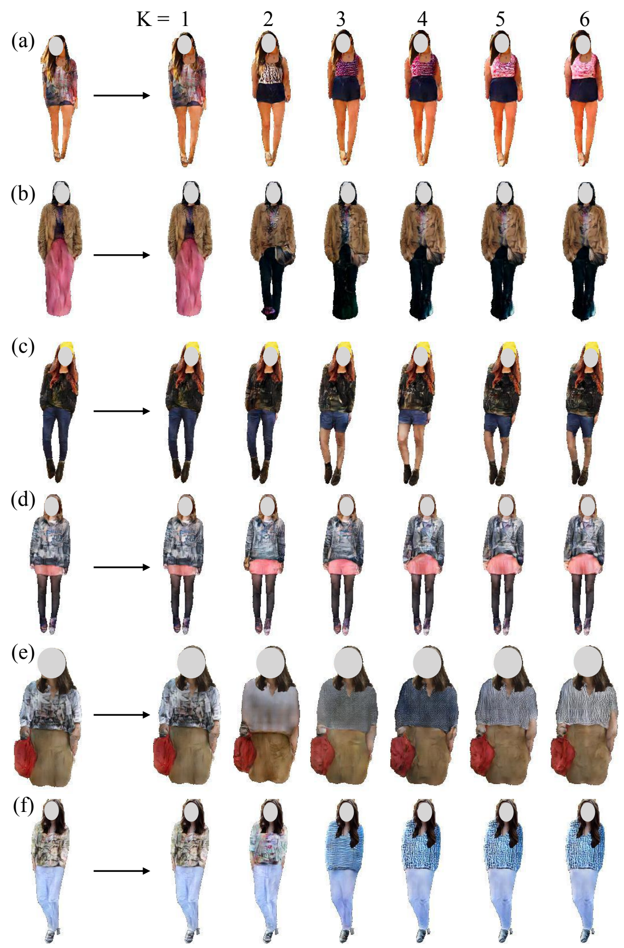

Figure 3: Spectrum of edits ($K = 1$ to 6) by Fashion++: the first column are the original outfits, and starting from the second are gradually editing the outfits more by taking more gradient steps, from 1 to 6.

the pink long skirt to black flared pants, which actually are not too different in shape, but makes the outfit more energetic and better color matching. (c) gradually shortens the length of the jeans to shorts. (d) tucks in more amount of the sweater. Both (e) and (f) change the pattern of the blouses to match the bottom better. In most examples, edits typically start saturating after $K = 4$, and changes are less obvious after $K = 6$.

VI. Mechanical Turk Interface

Fig. 8, Fig. 9, and Fig. 10 show our MTurk interfaces for the three human subject studies presented in the main paper. We give them the definition of minimal editing and good/bad examples of edits, and tell them to ignore artifacts in synthesized images. For A, we ask them to (i) choose whether any of the changed outfits become more fashionable, and (ii) which is the best minimal edited outfit and (iii) why. For B, we ask them two questions comparing the changed outfit to the original: (i) whether the changed outfit remains similar, and (ii) whether the changed outfit is more fashionable. For C, we ask them if (i) they understand what to change given the original and changed outfit, and (ii) describe it verbally.

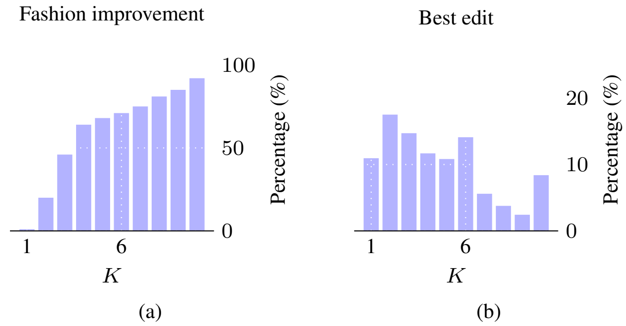

Figure 4: Results breaking down K for user study A: (a) which of the changed outfits become more fashionable, and (b) which edited outfit makes the best minimal edit to the original outfit.

VII. Detailed result for user study A

For question (i) in user study A, since there should be a consensus on fashionability improvement, we aggregate the responses over all subjects for each example. Each of the 100 testing examples will be judged as either improved or not improved for every K. The result is summarized in Fig. 4a. As more changes are made (increasing K), more examples are rated as improving fashionability, with 92% of them improved when $K = 10$.

Question (ii) is subjective in nature: different people prefer a different trade-off (between the amount of change versus the amount of fashionability added), so we treat response from each subject individually. The result is summarized in Fig. 4b. No specific K dominates, and a tendency of preferring $K \leq 6$ is observed, in total 80% of the time.

For question (iii), we ask users their reasons to selecting a specific K in question (ii). Examples of Turkers' responses are in Fig. 6. From phrases such as *add contrast*, *offer focus*, *pop*, or *catchy* in these examples, and a word cloud made from all responses (Fig. 5), we can tell that a common reason a user prefer an outfit is it being more attractive/interesting.

VIII. Verbal descriptions of actionable edits for user study C.

In the experiment presented as user study C in the main paper, we asked Turkers to rate how actionable the suggested edit is, and briefly describe the edit in words. Fig. 7 shows example descriptions from human judges. Each ex-

Figure 5: Summary in word cloud of why a changed outfit is preferred in user study A.

ample has 6 to 7 different descriptions from different people. For example, despite mild artifacts in Fig. 7(a), humans still reach consensus on the actionable information. Note that in Fig. 7(b)(c)(d), most people described the edit as changing color/pattern, while in Fig. 7(e)(f) more descriptions are about changing to/adding another garment, because Fig. 7(e)(f) changes garments in a more drastic way. *Tweaking* the color/pattern of a garment is essentially changing to another garment, yet humans perceived this differently. When the overall style of the outfit remains similar, changing to a garment with different colors/patterns seems like a slight change to humans.

References

[1] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[2] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*, 2017. 1

[3] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 1

[4] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NIPS*, 2017. 1

Figure 6: Examples of Turkers' responses to user study A: pairs of images on the left show the original outfits and the changed outfits generated by Fashion++ preferred by the user. Corresponding sentences on the right are their verbal explanation for why they make such selection.

(a)

- Different color for the shirt.
- Change shirt pattern.
- Color change.
- Color of shirt.
- Change the shirt color.
- Change the color of the shirt.
- Made the shirt slightly darker.

(b)

- Skirt is different.
- Different skirt color.
- Changed skirt.
- Change skirt pattern.
- Color is changed.
- Changed the skirt to a different style.
- Changed skirt pattern.

(c)

- Change the pants color.
- Change the pants color.
- Change trousers - different color.
- Change pants color.
- Change pants.
- Change pant color.
- Changed to another color pant.

(d)

- Different blouse underneath and added purse.
- Change bottom of shirt color.
- Change to another shirt: changed color to white.
- Changed the skirts color.
- Change color of long shirt under jacket.
- Change the color skirt.

(e)

- Different blouse.
- Different color shirt.
- Change shirt color.
- Changed blouse.
- Different blouse.
- Changed the type of blouse.

(f)

- Garment added.
- Change the shirt's color.
- Pattern is changed.
- Changed jacket pattern.
- Change top pattern.
- Different jacket.

Figure 7: Examples of Turkers' verbal descriptions about what is changed in a Fashion++ edit. Despite mild artifacts in the edits, note how humans reach consensus about what change is being recommended by the system.

Figure 8: Interface for human subject study A: understanding to what extent a given degree of change is preferred and why.

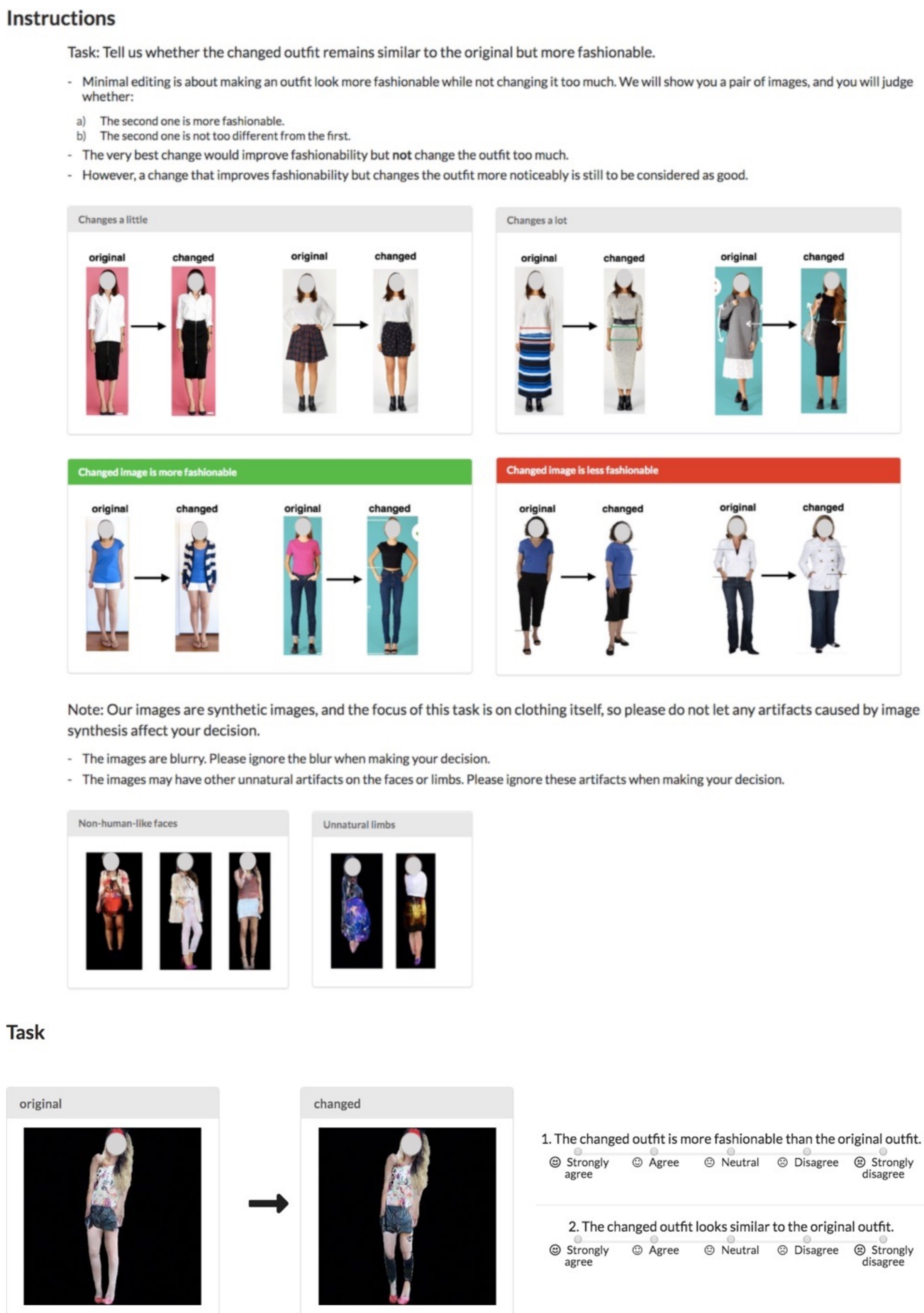

Figure 9: Interface for human subject study B: understanding how Fashion++ compares to the baselines.

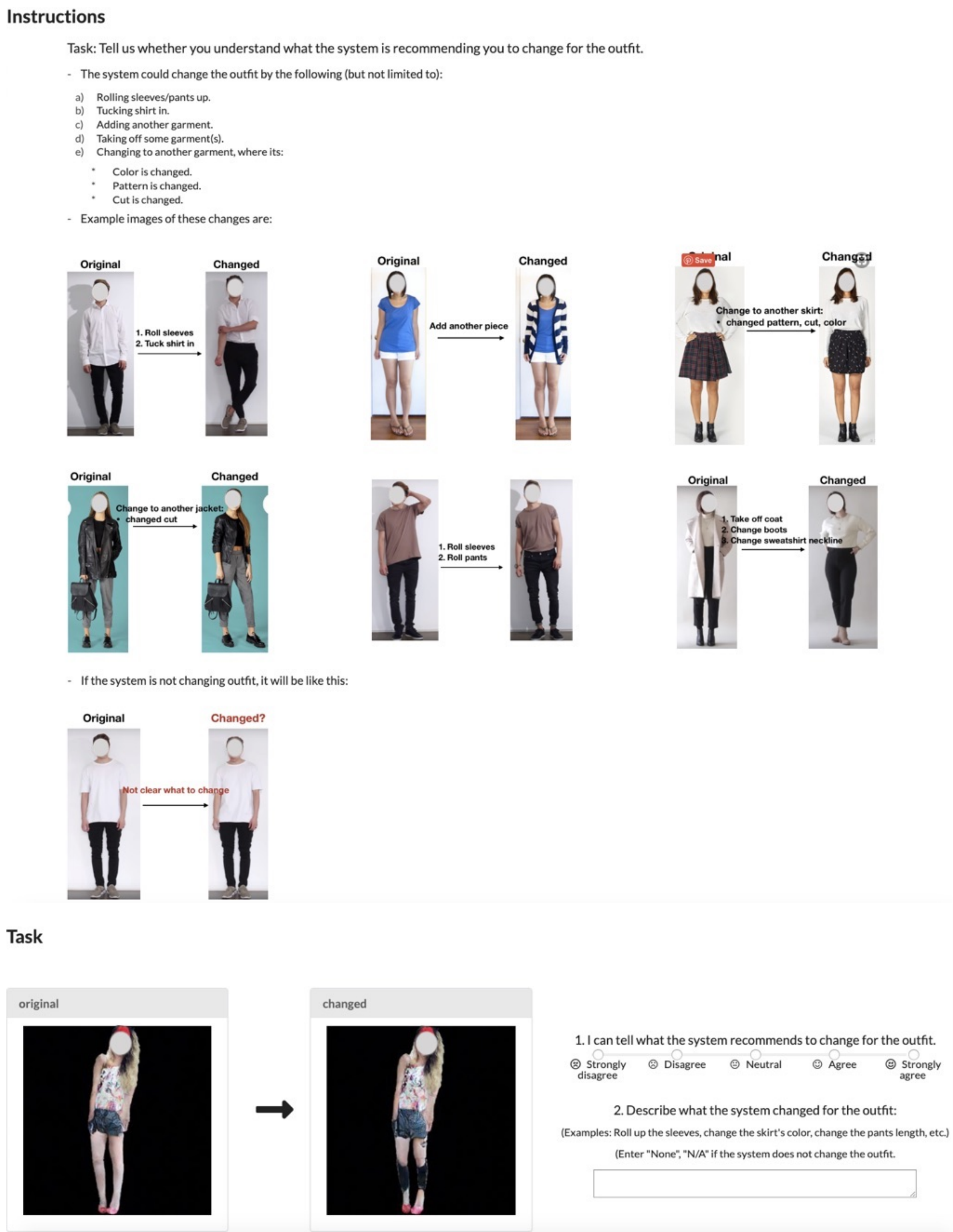

Figure 10: Interface for human subject study C: understanding whether humans can get *actionable information* from the suggested edits.