# Supplementary Material

## 1. Implementation Details

### 1.1. Pre-processing Details

This section introduces the details of generating our clothing-agnostic person representation. To remove the dependency on the clothing item originally worn by a person, regions that can provide any original clothing information, such as the arms that hint at the sleeve length, should be eliminated. Therefore, when generating a clothing-agnostic image $I_a$, we remove the arms from the reference image $I$. For the same reason, legs should be removed if the pants are the target clothing items. We mask the regions with a gray color, so that the masked pixels of the normalized image would have a value of 0. We add padding to the masks to thoroughly remove these regions, and the width of the padding is empirically determined.

### 1.2. Model Architectures

This section introduces the architectures of the segmentation generator, the geometric matching module, and ALIAS generator in detail.

**Segmentation Generator.** The segmentation generator has the structure of U-Net [6], which consists of convolutional layers, downsampling layers, and upsampling layers. Two multi-scale discriminators [9] are employed for the conditional adversarial loss. The details of the segmentation generator architecture are shown in Fig. 1.

**Geometric Matching Module.** The geometric matching module consists of two feature extractors and a regression network. A correlation matrix is calculated from the two extracted features, and the regression network predicts the TPS parameter $\theta$ with the correlation matrix. The feature extractor is composed of a series of convolutional layers, and the regression network consists of a series of convolutional layers followed by a fully connected layer. The details are shown in Fig. 2.

**ALIAS Generator.** The architecture of the ALIAS generator consists of a series of ALIAS ResBlks with nearest-neighbor upsampling layers. We employ two multi-scale discriminators with instance normalization. Spectral normalization [4] is applied to all the convolutional layers. Note that we separately standardize the activation based on the misalignment mask $M_{misalign}$ only in the first five ALIAS ResBlks. The details of the ALIAS generator architecture is shown in Fig. 3.

### 1.3. Training Details

This section introduces the losses and the hyperparameters for the segmentation generator, the geometric matching module, and the ALIAS generator.

**Segmentation Generator.** The segmentation generator $G_S$ uses the clothing-agnostic segmentation map $S_a$, the pose map $P$, and the clothing item $c$ as inputs ($\hat{S} = G_S(S_a, P, c)$) to predict the segmentation map $\hat{S}$ of the person in the reference image wearing the target clothing item. The segmentation generator is trained with the

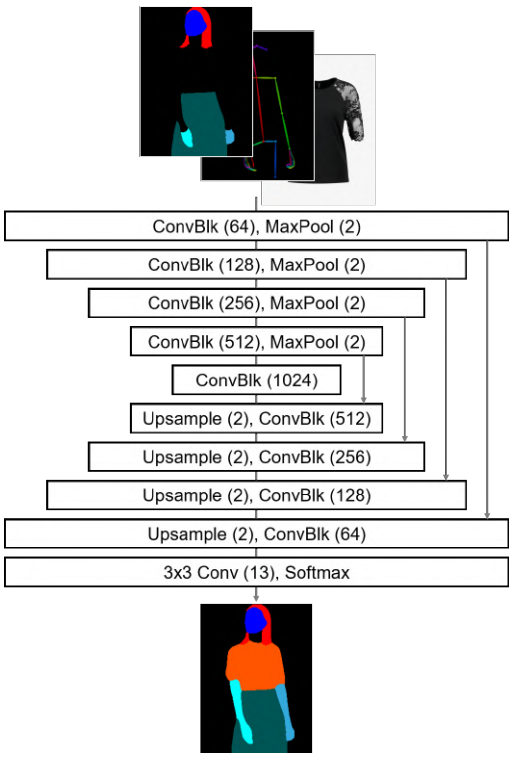


Figure 1: Segmentation Generator. $k \times k$ Conv ($x$) denotes a convolutional layer where the kernel size is $k$ and the output channel is $x$. Also, ConvBlk ($x$) denotes a block, which consists of two series of $3 \times 3$ convolutional layer, instance normalization, and ReLU activation.

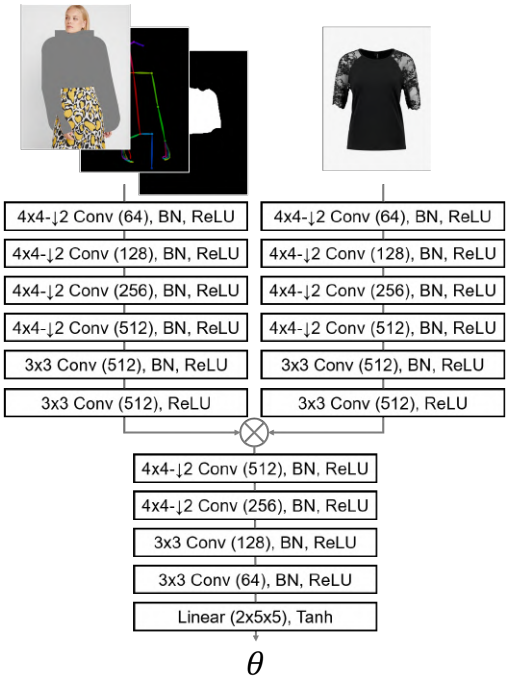


Figure 2: Geometric Matching Module. $k \times k$ $\downarrow 2$ Conv $(x)$ denotes a convolutional layer where the kernel size is $k$, the stride is 2, and the output channel is $x$.

cross-entropy loss $\mathcal{L}_{CE}$ and the conditional adversarial loss $\mathcal{L}_{cGAN}$, which is LSGAN loss [3]. The full loss $\mathcal{L}_S$ for the segmentation generator are written as

$$\mathcal{L}_S = \mathcal{L}_{cGAN} + \lambda_{CE}\mathcal{L}_{CE} \tag{1}$$

$$\mathcal{L}_{CE} = -\frac{1}{HW}\sum_{k\in C, y\in H, x\in W} S_{k,y,x}\log(\hat{S}_{k,y,x}) \tag{2}$$

$$\begin{aligned}\mathcal{L}_{cGAN} = \mathbb{E}_{(X,S)}[\log(D(X,S))] \\ + \mathbb{E}_X[1-\log(D(X,\hat{S}))],\end{aligned} \tag{3}$$

where $\lambda_{CE}$ is the hyperparameter for the cross-entropy loss. In the experiment, $\lambda_{CE}$ is set to 10. In Eq. (2), $S_{yxk}$ and $\hat{S}_{yxk}$ indicate the pixel values of the segmentation map of the reference image $S$ and $\hat{S}$ corresponding to the coordinates $(x, y)$ in channel $k$. The symbols $H$, $W$ and $C$ indicate the height, width, and the number of channels of $S$. In Eq. (3), the symbol $X$ indicates the inputs of the generator $(S_a, P, c)$, and $D$ denotes the discriminator.

The learning rate of the generator and the discriminator is 0.0004. We adopt the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train the segmentation generator for 200,000 iterations with the batch size of 8.

**Geometric Matching Module.** The inputs of the geometric matching module are $c$, $P$, clothing-agnostic image $I_a$, and $\hat{S}_c$, which is the clothing area of $\hat{S}$. The output is

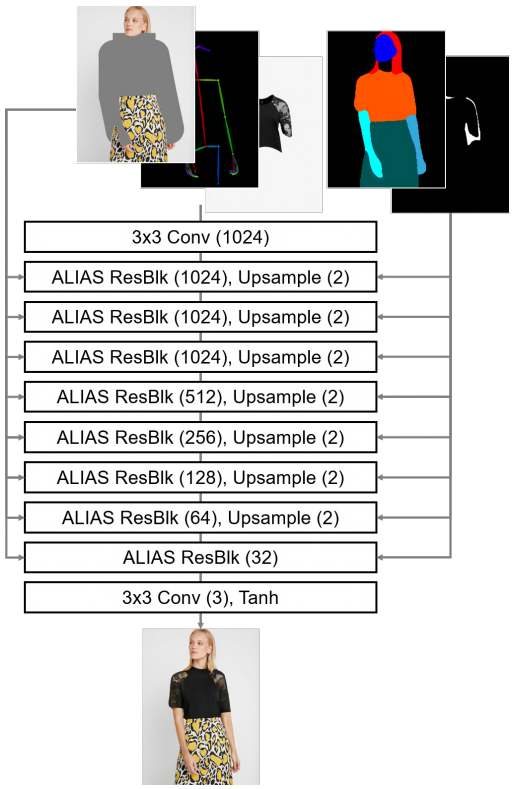


Figure 3: ALIAS Generator. The segmentation map $S$ and the misalignment mask $M_{misalign}$ are passed to the generator through the proposed ALIAS ResBlks.

the TPS transformation parameters $\theta$. The overall objective function is written as

$$\mathcal{L}_{warp} = ||I_c - \mathcal{W}(c,\theta)||_{1,1} + \lambda_{const}\mathcal{L}_{const} \tag{4}$$

$$\begin{aligned}\mathcal{L}_{const} = \sum_{p\in\mathbf{P}} |\,(|\,||pp_0||_2 - ||pp_1||_2| + |\,||pp_2||_2 - ||pp_3||_2|) \\ +(|\mathcal{S}(p,p_0) - \mathcal{S}(p,p_1)| + |\mathcal{S}(p,p_2) - \mathcal{S}(p,p_3)|),\end{aligned} \tag{5}$$

where $\mathcal{W}$ is the function that deforms $c$ using $\theta$, and $I_c$ is the clothing item extracted from the reference image $I$. $\mathcal{L}_{const}$ is a second-order difference constraint [10], and $\lambda_{const}$ is the hyperparameter for $\mathcal{L}_{const}$. In the experiment, we set $\lambda_{const}$ to 0.04. In Eq. (5), the symbol $p$ indicates a sampled TPS control point from the entire control points set $\mathbf{P}$, and $p_0$, $p_1$, $p_2$, and $p_3$ are top, bottom, left and right point of $p$, respectively. The function $\mathcal{S}(p, p_i)$ denotes the slope between $p$ and $p_i$.

The learning rate of the geometric matching module is 0.0002. We adopt the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We train the geometric matching module for 50,000 iterations with the batch size of 8.

**ALIAS Generator.** The loss function of ALIAS generator follows those of SPADE [5] and pix2pixHD [9], as it

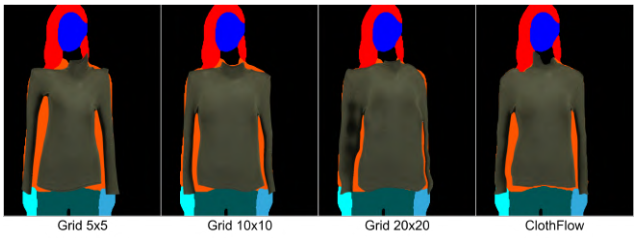


Figure 4: Qualitative comparisons of TPS transformation with various grid numbers and the flow estimation from ClothFlow.

| Method | Warp-SSIM$_\uparrow$ | MACs$_\downarrow$ | Mask-SSIM$_\uparrow$ |
|---|---|---|---|
| ClothFlow | **0.841**$^\star$ | 8.13G | 0.803$^\star$ |
| VITON-HD | 0.782 | **4.47G** | **0.852** |

Table 1: $\star$ denotes a score taken from the ClothFlow paper, and we train VITON-HD in the same setting (*e.g.*, dataset and resolution). We compute MACs of their warping modules at 256×192.

contains the conditional adversarial loss $\mathcal{L}_{cGAN}$, the feature matching loss $\mathcal{L}_{FM}$, and the perceptual loss $\mathcal{L}_{percept}$. Let $D_I$ be the discriminator, $I$ and $c$ be the given reference and target clothing images, and $\hat{I}$ be the synthetic image generated by the generator. $S_{div}$ is the modified version of the segmentation map $S$. The full loss $\mathcal{L}_I$ of our generator is written as

$$\mathcal{L}_I = \mathcal{L}_{cGAN} + \lambda_{FM}\mathcal{L}_{FM} + \lambda_{percept}\mathcal{L}_{percept} \quad (6)$$

$$\begin{aligned}\mathcal{L}_{cGAN} = \mathbb{E}_I[\log(D_I(S_{div}, I))] \\ + \mathbb{E}_{(I,c)}[1 - \log(D_I(S_{div}, \hat{I}))]\end{aligned} \quad (7)$$

$$\mathcal{L}_{FM} = \mathbb{E}_{(I,c)}\sum_{i=1}^{T}\frac{1}{K_i}[||D_I^{(i)}(S_{div}, I) - D_I^{(i)}(S_{div}, \hat{I})||_{1,1}] \quad (8)$$

$$\mathcal{L}_{percept} = \mathbb{E}_{(I,c)}\sum_{i=1}^{V}\frac{1}{R_i}[||F^{(i)}(I) - F^{(i)}(\hat{I})||_{1,1}], \quad (9)$$

where $\lambda_{FM}$ and $\lambda_{percept}$ are hyperparameters. In the experiment, both $\lambda_{FM}$ and $\lambda_{percept}$ are set to 10. $T$ is the number of layers in $D_I$, and $D_I^{(i)}$ and $K_i$ are the activation and the number of elements in the $i$-th layer of $D_I$, respectively. Similarly, $V$ is the number of layers used in the VGG network $F$ [7], and $F^{(i)}$ and $R_i$ are the activation and the number of elements in the $i$-th layer of $F$, respectively. We replace the standard adversarial loss with the Hinge loss [11].

The learning rate of the generator and the discriminator is 0.0001 and 0.0004, respectively. We adopt the Adam optimizer [2] with $\beta_1 = 0$ and $\beta_2 = 0.9$. We train the ALIAS generator for 200,000 iterations with the batch size of 4.

## 2. Additional Experiments

### 2.1. Comparison with ClothFlow

To demonstrate that the optical flow estimation does not solve the misalignment completely, we re-implement the flow estimation module of ClothFlow [1] based on the original paper. Fig. 4 shows that the misalignment still occurs, although both TPS with a higher grid number (e.g., a 10×10 or 20×20 grid) and the flow estimation module of ClothFlow can reduce the misaligned regions. The reason is that the regularization to avoid the artifacts (*e.g.*, TV loss) prevents the warped clothes from fitting perfectly into the target region. In addition, we evaluate the accuracy and the computational cost of warping modules in VITON-HD and ClothFlow with Warp-SSIM [1] and MACs, respectively. We also measure how well the models reconstruct the clothing using Mask-SSIM [1]. Table 1 shows that the ClothFlow warping module has the better accuracy than ours, whereas the higher Mask-SSIM in VITON-HD proves that ALIAS normalization is more effective at solving the misalignment problem than the improved warping method. We found that the ClothFlow warping module needs a huge computational cost (MACs: 130.03G) at 1024×768, but the cost could be reduced when predicting the optical flow map at 256×192. Table 1 demonstrates that the ClothFlow warping module still needs more computational cost than ours, yet it is a viable option to combine the flow estimation module with ALIAS generator.

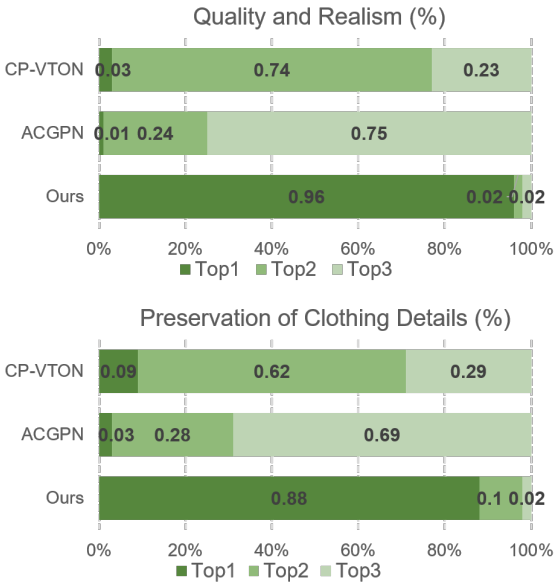


Figure 5: User study results. We compare our model with CP-VTON [8] and ACGPN [10].

Figure 6: Failure cases of VITON-HD.

### 2.2. User Study

We further evaluate our model and other baselines via a user study in the unpaired setting. We randomly select 30 sets of a reference image and a target clothing image from the test dataset. Given the reference images and the target clothes, the users are asked to rank the 1024×768 outputs of our model and baselines according to the following questions: (1) Which image is the most photo-realistic? (2) Which image preserves the details of the target clothing the most? As shown in Fig. 5, it can be observed that our approach achieves the rank 1 votes more than 88% for the both questions. The result demonstrates that our model generates more realistic images, and preserves the details of the clothing items compared to the baselines.

### 2.3. Qualitative Results

We provide additional qualitative results to demonstrate our model's capability of handling high quality image synthesis. Fig. 7, 8, 9, and 10 show the qualitative comparison of the baselines across different resolutions. Fig. 11, 12, 13, and 14 show additional results of VITON-HD at 1024×768 resolution.

## 3. Failure Cases and Limitations

Fig. 6 shows the failure cases of our model caused by the inaccurately predicted segmentation map or the inner collar region indistinguishable from the other clothing region. Also, the boundaries of the clothing textures occasionally fade away.

The limitations of our model are as follows. VITON-HD is trained to preserve the bottom clothing items, limiting the presentation of the target clothes (*e.g.*, whether they are tucked in). It can be a valuable future direction to generate multiple possible outputs from a single input pair. Next, our dataset mostly consists of slim women and top clothing images, which makes VITON-HD handle only a limited range of body shapes and clothing during the inference. We believe that VITON-HD has the capability to cover more diverse cases when the images of various body shapes and clothing types are provided. Finally, existing virtual try-on methods including VITON-HD do not provide robust performance for in-the-wild images. We think generating realistic try-on images for the in-the-wild images is an interesting topic for future work.

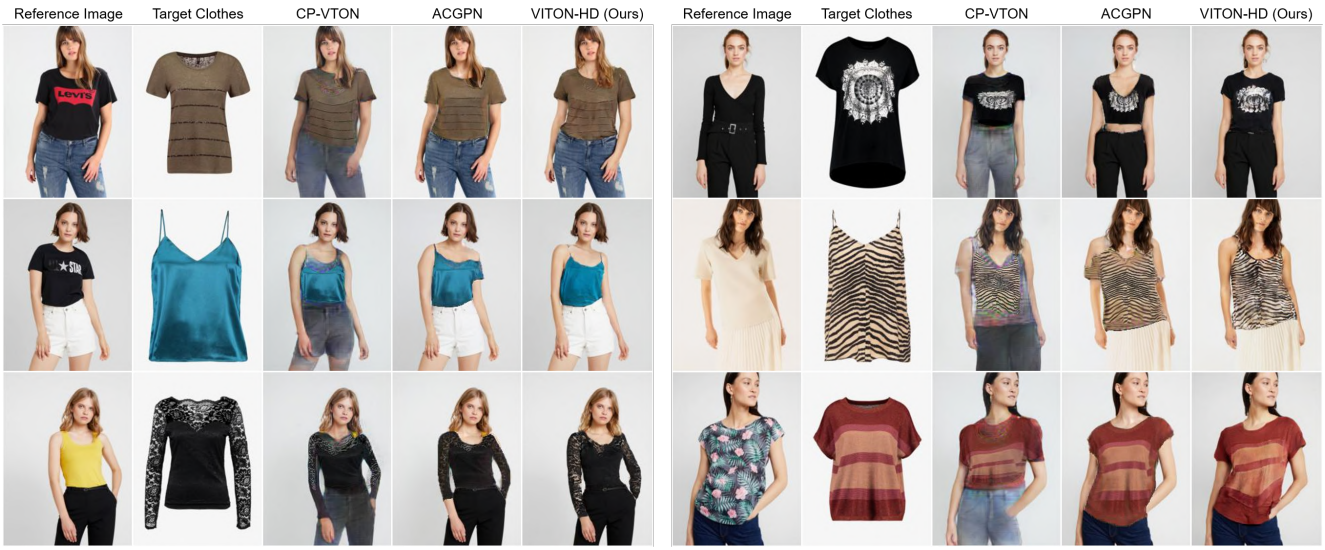


Figure 7: Qualitative comparison of the baselines (256×192).

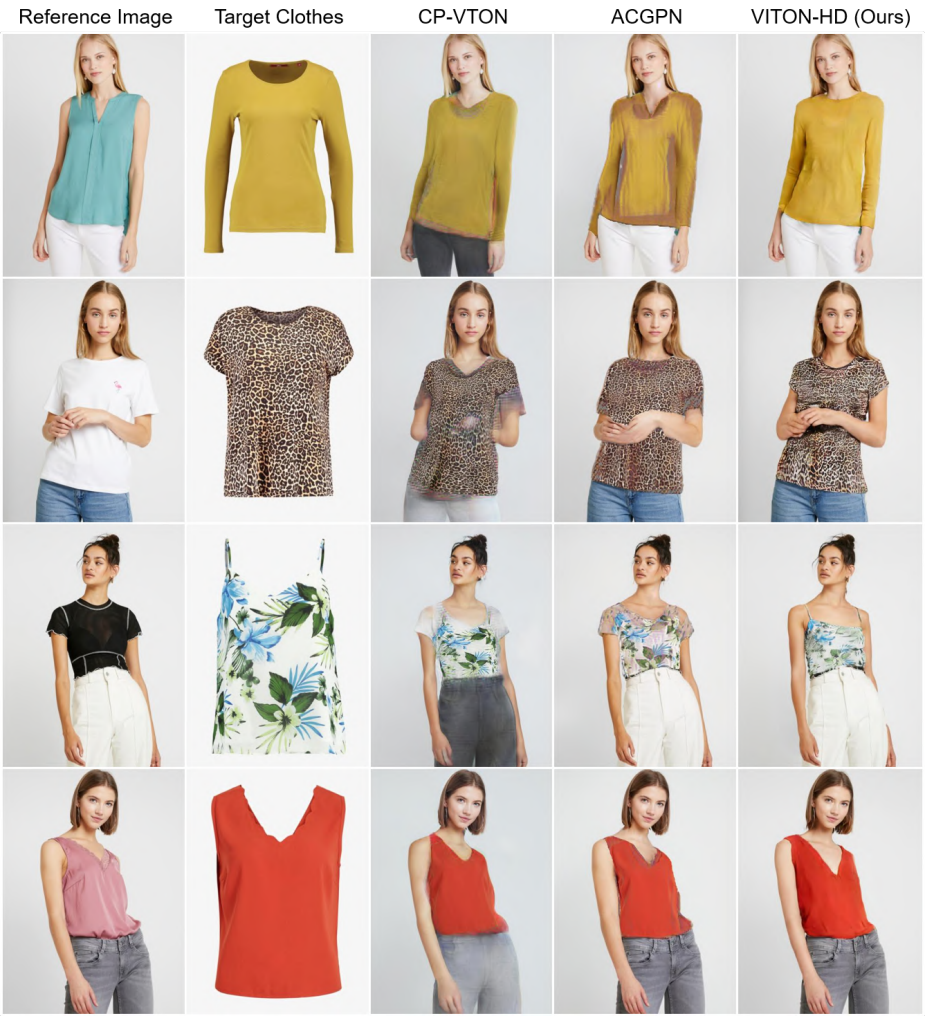


Figure 8: Qualitative comparison of the baselines (512×384).


Figure 9: Qualitative comparison of the baselines (1024×768).


Figure 10: Qualitative comparison of the baselines (1024×768).

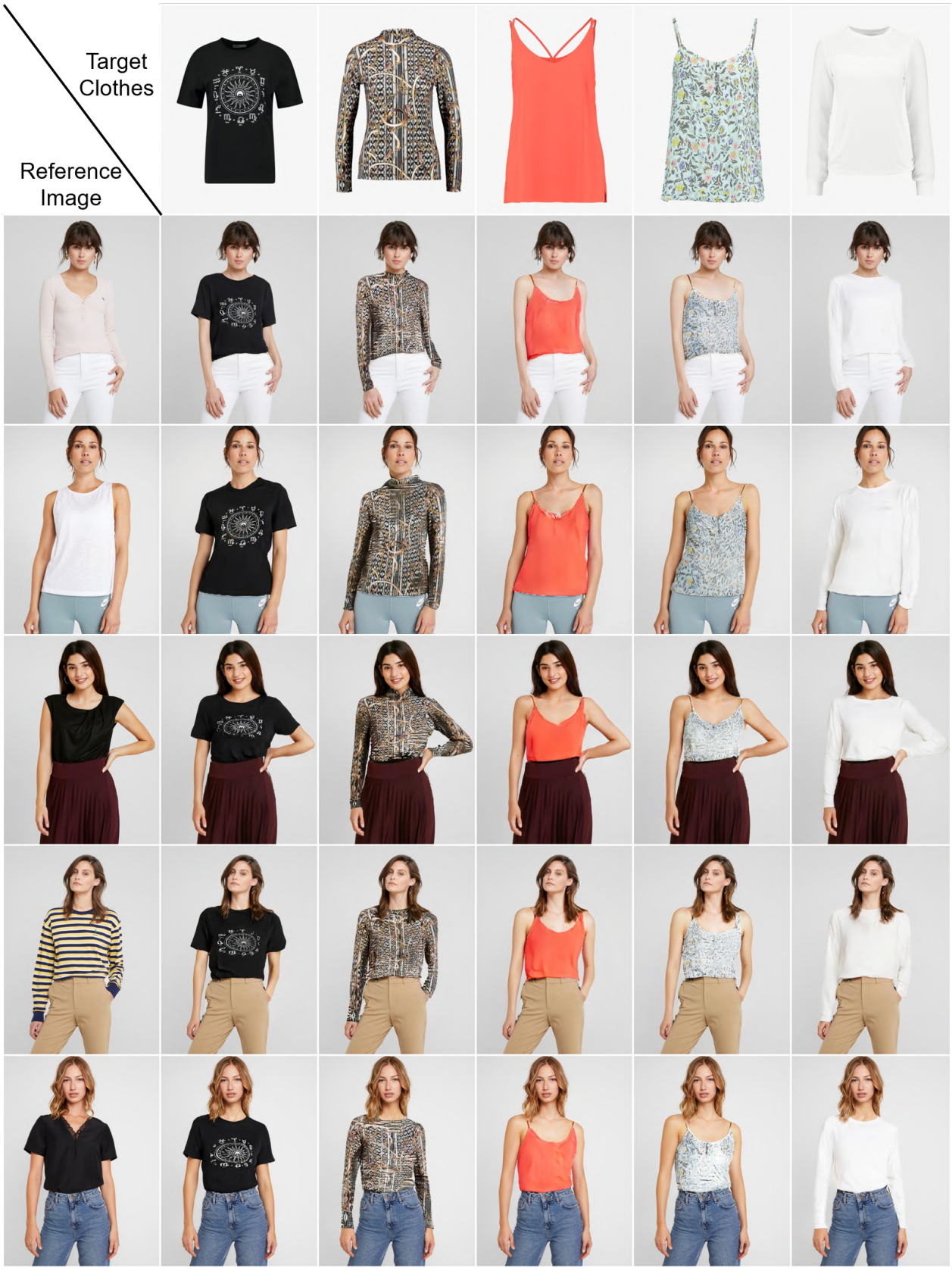


Figure 11: Additional qualitative results of VITON-HD.

Figure 12: Sample 1 of VITON-HD. (*Left*) The synthetic image. (*Right*) The reference image and the target clothing item.


Figure 13: Sample 2 of VITON-HD. (*Left*) The synthetic image. (*Right*) The reference image and the target clothing item.


Figure 14: Sample 3 of VITON-HD. (*Left*) The synthetic image. (*Right*) The reference image and the target clothing item.

## References


[1] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10471–10480, 2019. 3

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 3

[3] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 2

[4] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 1

[5] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[8] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 589–604, 2018. 3

[9] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 1, 2

[10] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7850–7859, 2020. 2, 3

[11] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019. 3