Supplementary Material

A. Implementation Details

In this section, we describe the detailed architectures, hyper-parameters, and objective functions of the try-on condition generator and the image generator.

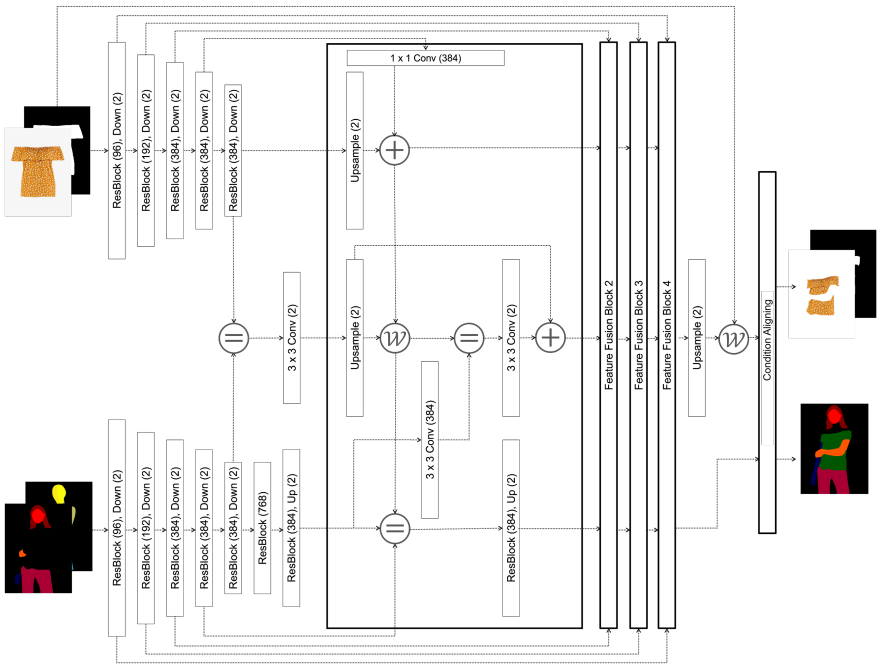

Fig. 1: The detailed architecture of the try-on condition generator. (ResBlock (n), Up/Down (f)) denotes a residual block where the scaling factor is f and the output channel is n. Conv (m) denotes a convolutional layer where the output channel is m.

Try-On Condition Generator. The try-on condition generator consists of two encoders and four feature fusion blocks, and each encoder is composed of five residual blocks. The features of the last residual blocks are concatenated and passed to a 3×3 convolutional layer, which generates the first flow map of the flow pathway. Also, the last feature of the segmentation encoder is used as the input of the segmentation pathway (*i.e.*, seg pathway) after passing through two residual blocks. We employ two multi-scale discriminators for the conditional adversarial loss. The visualization of the try-on condition generator architecture is in Fig. 1.

During the training of our try-on condition generator, the model predicts $\hat{I}_c, \hat{S}_c$, and $\hat{S}$ at 256×192 resolution. In the inference phase, before forwarding

our try-on image generator, the segmentation map and the appearance flow obtained from the try-on condition generator are upscaled to 1024×768. We down-sampled the inputs for the discriminator of our try-on condition generator by a factor of 2 to increase the receptive field. In addition, we apply a dropout [6] to the discriminator to stabilize the training. For hyper-parameters we used, $\lambda_{CE}, \lambda_{VGG}$, and λ_{TV} are set to 10, 10, and 2, respectively. The batch sizes for training our try-on condition generator and image generator are set to 8 and 4, respectively. We train each module for 100,000 iterations. The learning rates of the generator and the discriminator of the try-on condition generator are set to 0.0002.

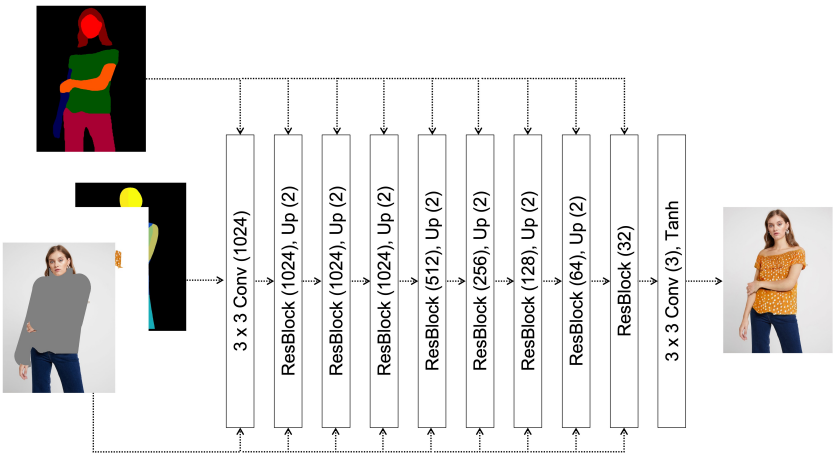

Fig. 2: The detailed architecture of the try-on image generator. (ResBlock (n), Up (f)) denotes a residual block, where the scaling factor is f, and the output channel is n. Conv (m) denotes a convolutional layer where the output channel is m.

Try-On Image Generator. We describe the detailed architecture of the try-on image generator as shown in Fig. 2. The generator is composed of a series of residual blocks with upsampling layers, and two multi-scale discriminators are employed for the conditional adversarial loss. Spectral normalization [4] is applied to all the convolutional layers.

To train the try-on image generator, we utilize the same losses used in SPADE [5] and pix2pixHD [7]. Specifically, our full objective function consists of the conditional adversarial loss, the perceptual loss, and the feature matching loss. Formally, our objective function is as follows:

$$\mathcal{L}_{TOIG} = \mathcal{L}_{cGAN}^{TOIG} + \lambda_{VGG}^{TOIG}\mathcal{L}_{VGG}^{TOIG} + \lambda_{FM}^{TOIG}\mathcal{L}_{FM}^{TOIG}, \tag{1}$$

where $\mathcal{L}_{cGAN}^{TOIG}$, $\mathcal{L}_{VGG}^{TOIG}$, and $\mathcal{L}_{FM}^{TOIG}$ denote the conditional adversarial loss, the perceptual loss, and the feature matching loss [7], respectively. We use λ_{VGG}^{TOIG} and λ_{FM}^{TOIG} for hyper-parameters controlling relative importance between different

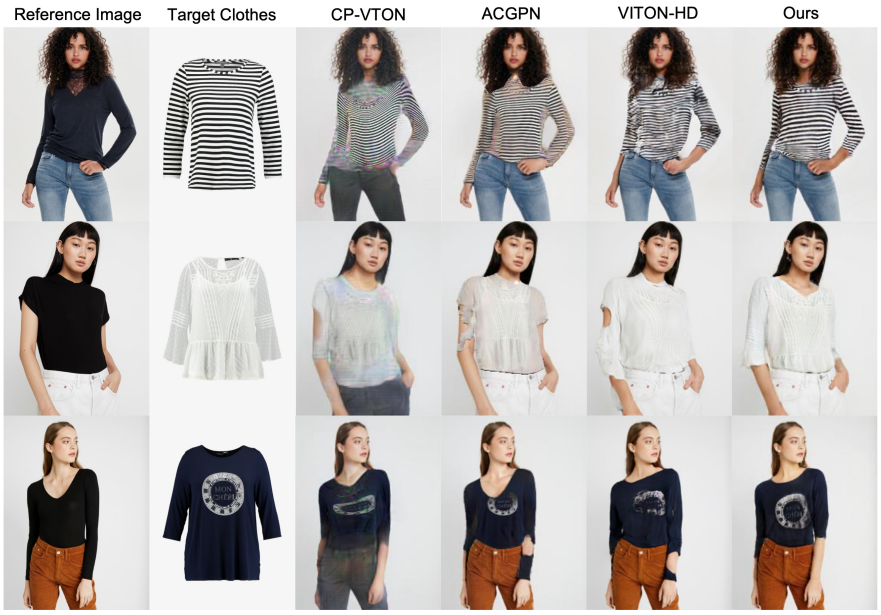

Fig. 3: Qualitative comparison of the baselines (256×192)

losses. For $\mathcal{L}_{GAN}^{TOIG}$, we employ the Hinge loss [3]. λ_{VGG}^{TOIG} and λ_{FM}^{TOIG} are set to 10. The learning rates of the generator and the discriminator of the try-on image generator are set to 0.0001 and 0.0004, respectively. We adopt the Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for both modules.

B. Additional Experiments

Results on Different Resolutions. We provide the additional qualitative results for comparison across different resolutions (Fig. 3, and Fig. 4).
Comparison with the Variant of VITON-HD. Previous studies [2,1] improve the performance of the geometric deformation for the target clothes by utilizing the appearance flow. However, simply increasing the degree of freedom of the warping module cannot perfectly remove the artifacts caused by misalignment and pixel-squeezing. To verify this, we further compare our method with VITON-HD*, the VITON-HD variant of which the clothes warping module is replaced by that of Clothflow [2]. Since Clothflow is superior to the warping module of VITON-HD, VITON-HD* can reduce the misalignment region.

Despite the improvement of the warping module in VITON-HD, our model consistently outperforms the VITON-HD* in all evaluation metrics, as seen in Table 1. Also, *2nd* column in Fig. 5 shows that VITON-HD* still suffers from the artifacts due to the misalignment. Furthermore, increasing the degree of freedom of the warping module exacerbates the pixel-squeezing artifact, indicating that

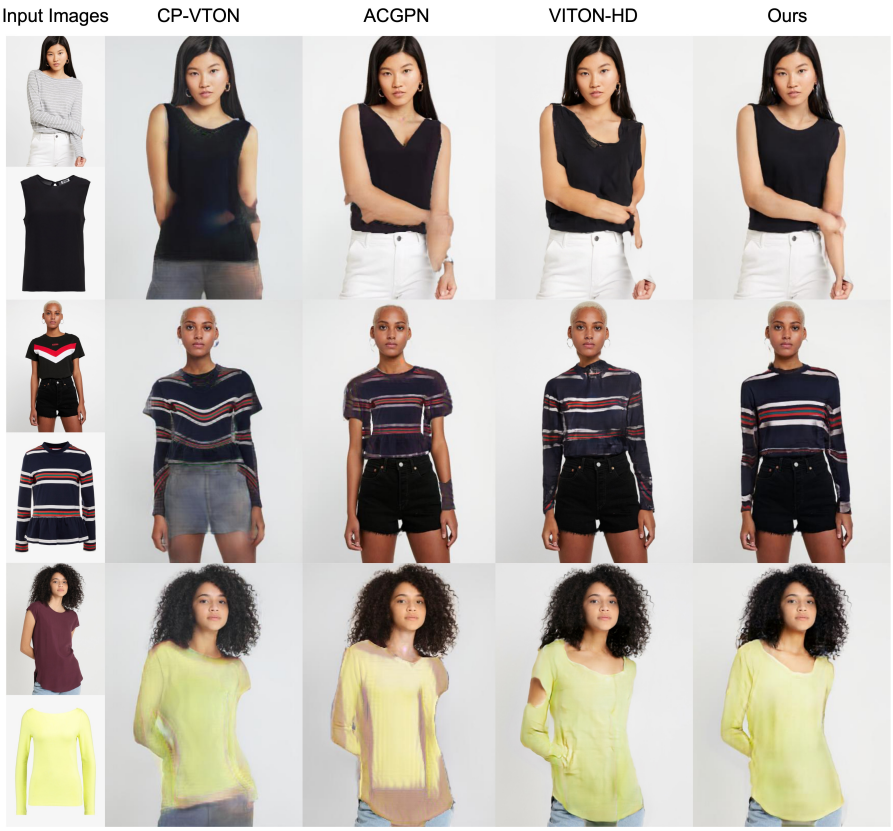

Fig. 4: Qualitative comparison of the baselines (512×384)

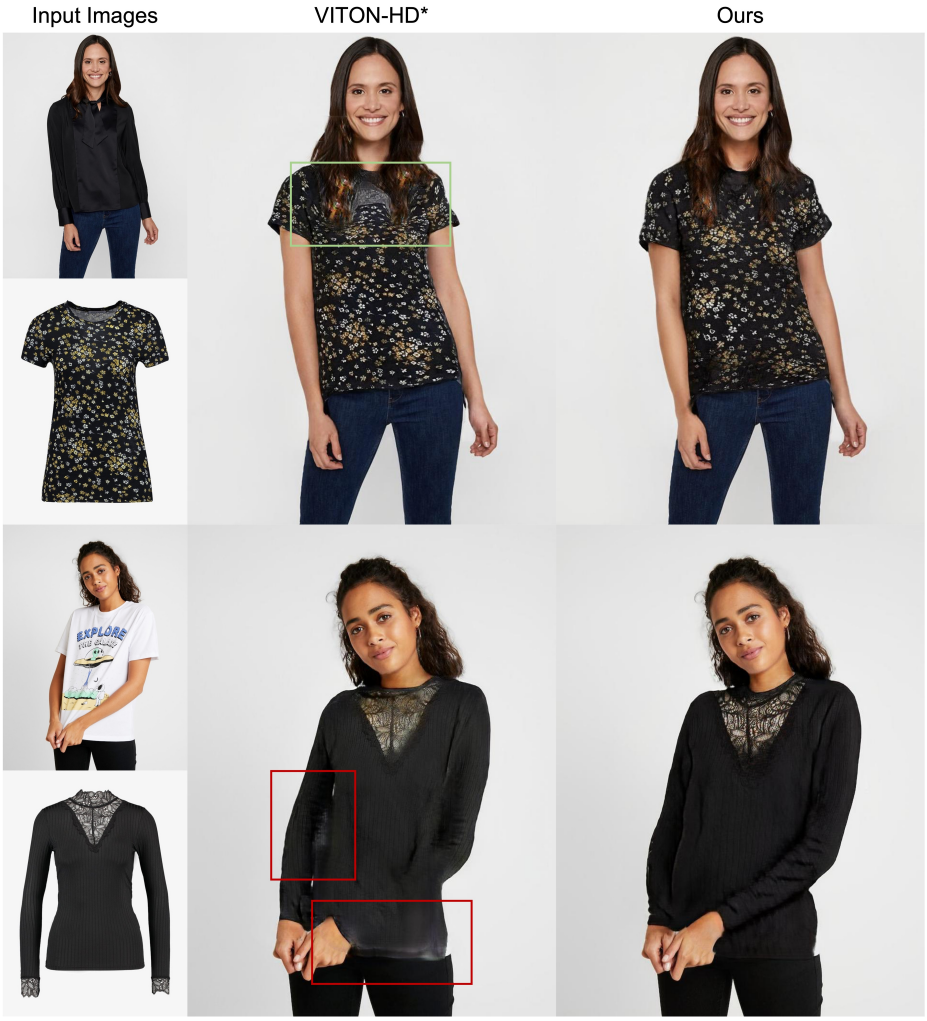

Fig. 5: Qualitative comparison with VITON-HD* (1024×768). VITON-HD* suffers from the misalignment and the pixel-squeezing artifacts indicated by green and red colored areas, respectively.

	LPIPS$_\downarrow$	SSIM$_\uparrow$	FID$_\downarrow$	KID$_\downarrow$
VITON-HD*	0.070	0.875	11.55	0.2993
Ours	**0.065**	**0.892**	**10.91**	**0.1794**

Table 1: Quantitative comparison with VITON-HD* at the 1024×768 resolution. We describe the KID as a value multiplied by 100.

the use of appearance flow without proper occlusion handling can be harmful. On the other hand, our model successfully solves both the misalignment and the pixel-squeezing problems, as shown in *3rd* column in Fig. 5.

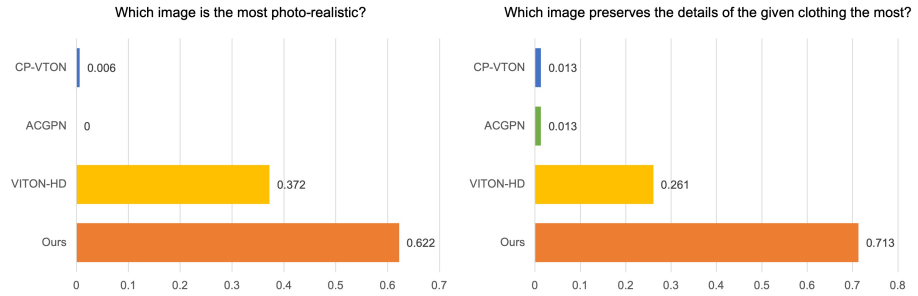

Fig. 6: User study results.

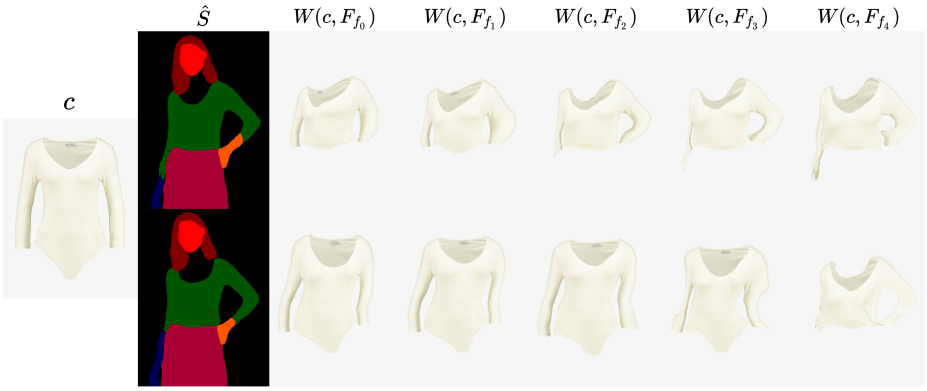

Fig. 7: Effects of the multi-scale $L1$/VGG losses. $1st$ row: w/ multi-scale losses. $2nd$ row: w/o multi-scale losses.

User Study. We conduct a user study to further assess our model and other baselines at the 1024×768 resolution. Given the 30 sets of a reference image and a target garment image from the test set, the users are asked to choose an image among the synthesized results 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 given clothing the most? In addition, a total of 21 participants participate in the user study. Fig. 6 shows that our model achieves the highest average selection rate for both questions, indicating that our model synthesizes more perceptually convincing results and preserves the detail of the clothing items better than other baselines.

Effectiveness of Multi-Scale L1/VGG Losses. During the training of the try-on condition generator, $\mathcal{L}_{L1}$ and $\mathcal{L}_{VGG}$ are directly applied to the intermediate flow estimations. As shown in *2nd* row of Fig. 7, the model without the multi-scale losses has difficulty learning flow estimation in a coarse scale.

Multi-scale losses enable the model to learn the meaningful intermediate flow estimation, which is crucial for the coarse-to-fine generation of appearance flow.

Additional Results. We present additional qualitative results of our model. Fig. 8 shows the combination of different clothes and different people, and Fig. 9-11 shows the high-resolution synthesis results (*i.e.*, 1024×768).

Fig. 8: Qualitative results of our model (1024×768).

Fig. 9: Qualitative results of our model (1024×768). The reference image and the target clothes (*left*), the synthesis image (*right*).

Fig. 10: Qualitative results of our model (1024×768). The reference image and the target clothes (*left*), the synthesis image (*right*).

Fig. 11: Qualitative results of our model (1024×768). The reference image and the target clothes (*left*), the synthesis image (*right*).

References

1. Chopra, A., Jain, R., Hemani, M., Krishnamurthy, B.: Zflow: Gated appearance flow-based virtual try-on with 3d priors. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5433–5442 (2021)
2. Han, X., Hu, X., Huang, W., Scott, M.R.: Clothflow: A flow-based model for clothed person generation. In: Proc. of the IEEE international conference on computer vision (ICCV). pp. 10471–10480 (2019)
3. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
4. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018)
5. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2337–2346 (2019)
6. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
7. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018)