1 Appendix

1.1 Implementation details

Convolutional geometric matcher. To extract the feature maps, we apply five times one standard convolution layer followed by a 2-strided convolution layer which downsamples the maps. The depth of the feature maps at each scale is (16,32,64,128,256). The correlation map is then computed and feeds a regression network composed of two 2-strided convolution layers, two standard convolution layers and one final fully connected layer predicting a vector $\theta \in \mathbb{R}^{50}$. We use batch normalization [3] and relu activation. The parameters of the two feature maps extractors are not shared.

Siamese U-net generator. We use the same encoder architecture as in the convolutional geometric matcher, but we store the feature maps at each scale. The decoder has an architecture symmetric to the encoder. There are five times one standard convolution layer followed by a 2-strided deconvolution layer which upsamples the feature maps. After a deconvolution, the feature maps are concatenated with the feature maps passed through the skip connections. In the generator, we use instance normalization, which shows better results for image and texture generation [7], with relu activation.

Discriminator. We adopt the fully convolutional discriminator from Pix-2-Pix [4], but with five downsampling layers instead of three in the original version. Each of it is composed of: 2-strided convolution, batch normalization, leaky relu, 1-strided convolution, batch normalization, leaky relu.

Adversarial loss. We use the relativistic formulation of the adversarial loss [5]. In this formulation, the discriminator is trained to predict that real images are more real than synthesized ones, rather than trained to predict that real images are real and synthesized images are synthesized. We also use gradient penalty on the discriminator.

Optimization. We use the Adam optimizer [6] with $\beta_1 = 0.5$, $\beta_2 = 0.999$, a learning rate of $10e^{-3}$ and a batch size of 8. Also, we use $\lambda_p = \lambda_{L1} = \lambda_w = \lambda_{adv} = 1$.

Hardware. We use a NVIDIA Tesla V100 with 16GB of RAM. The training takes around 2 days for T-WUTON, and around 3 days for S-WUTON. For inference, S-WUTON processes $\sim$77 frames per second.

1.2 More results from S-WUTON

We show more results from S-WUTON model in Fig. 1. It shows the abilities of our model to preserve complex cloth patterns (stripes, text or textures) and body details. It is robust across a wide range of human pose. On the antepenultimate column, we show a common failure case of our method, when sleeves of the source person are too large.

Fig. 1: More visual results from S-WUTON.

1.3 More visual examples on the importance of distillation

In Fig. 2, we show more visual results proving the soundness of our teacher-student approach.

Visually, our student model solves two kinds of problems: it is robust to human parser errors; it preserves important information that is masked to the standard virtual try-ons (hands, skin, handbags).

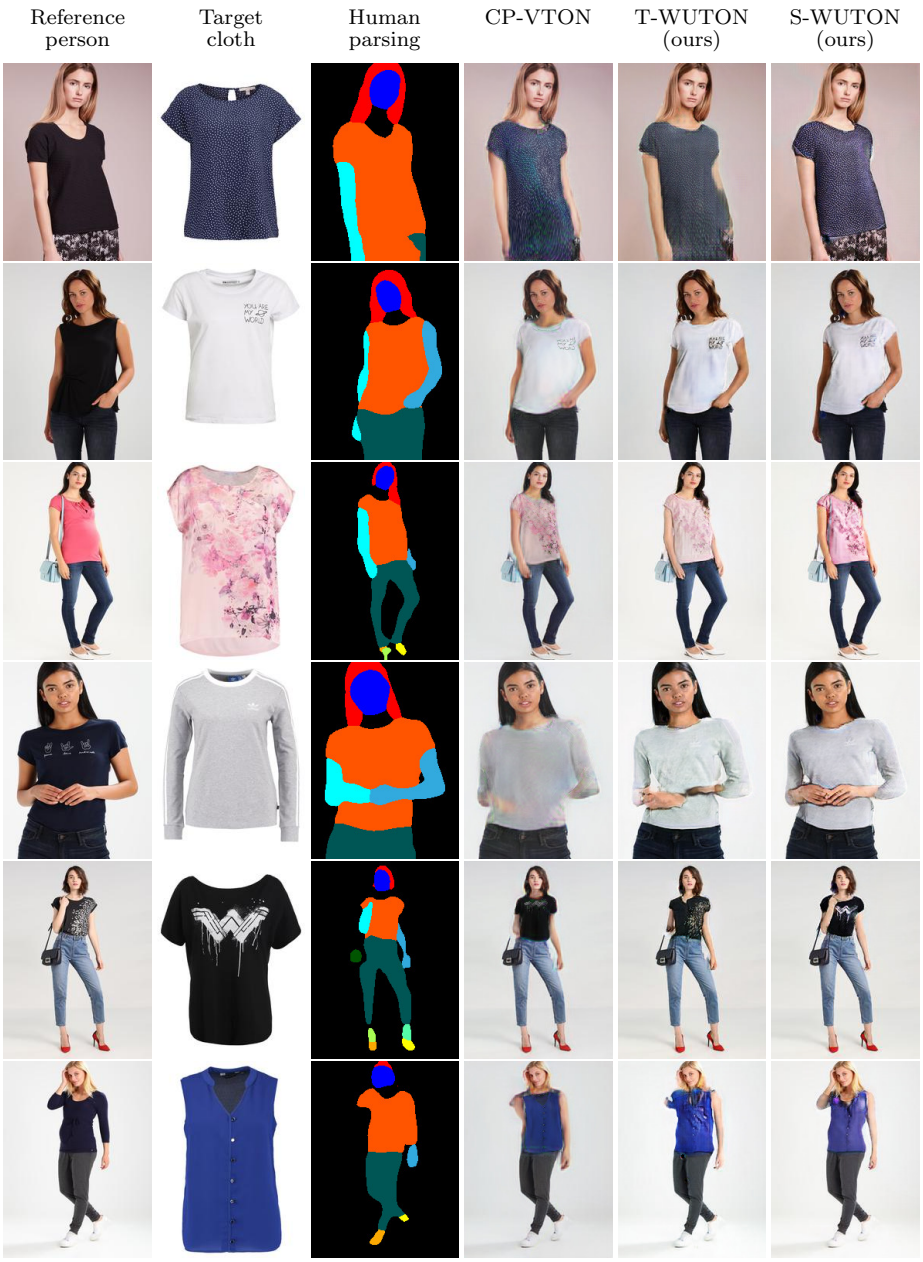


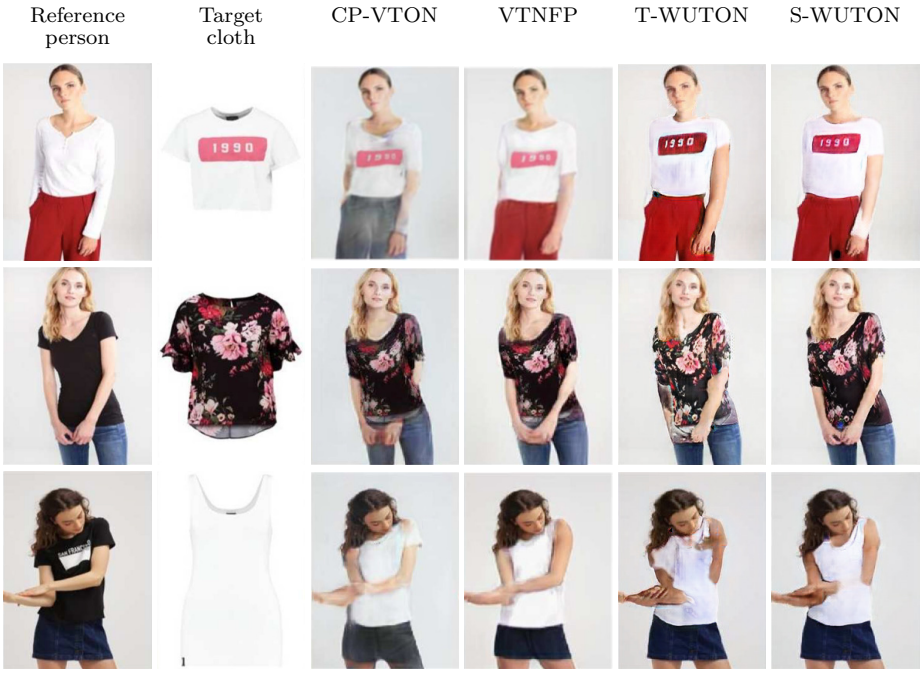
Fig. 2: Visual results proving the importance of the student-teacher approach. It is robust to parsing errors and preserves person's attributes such as arms, hands, and handbags.

1.4 Comparisons with VTNFP [8]

In Fig. 2, we show visual comparisons between CP-VTON, VTNFP, T-WUTON and S-WUTON. Images from all columns except T-WUTON and S-WUTON are taken from their paper, which explains the low resolution. Note that they trained their model on the original VITON dataset, that is now forbidden due to copyright issues. As mentioned in the paper, our model is trained on the dataset released in MG-VTON [1].

Results show that S-WUTON produces sharper images, with body details better preserved (especially the hands). Cloth patterns are also better rendered with S-WUTON (e.g. row 2). However, their model handles better difficult poses, when models are crossing arms (e.g rows 5,6,8). Note that our model performs well on persons crossing arms on MG-VTON dataset (see paper and Fig. 2 of Appendix).

In terms of computational cost at inference time, our S-WUTON is at least 13 times faster than their model. Moreover, since their pipeline relies on human parsing and pose estimation, it is also sensitive to the errors exhibited in our paper and in Fig. 2 of the Appendix.



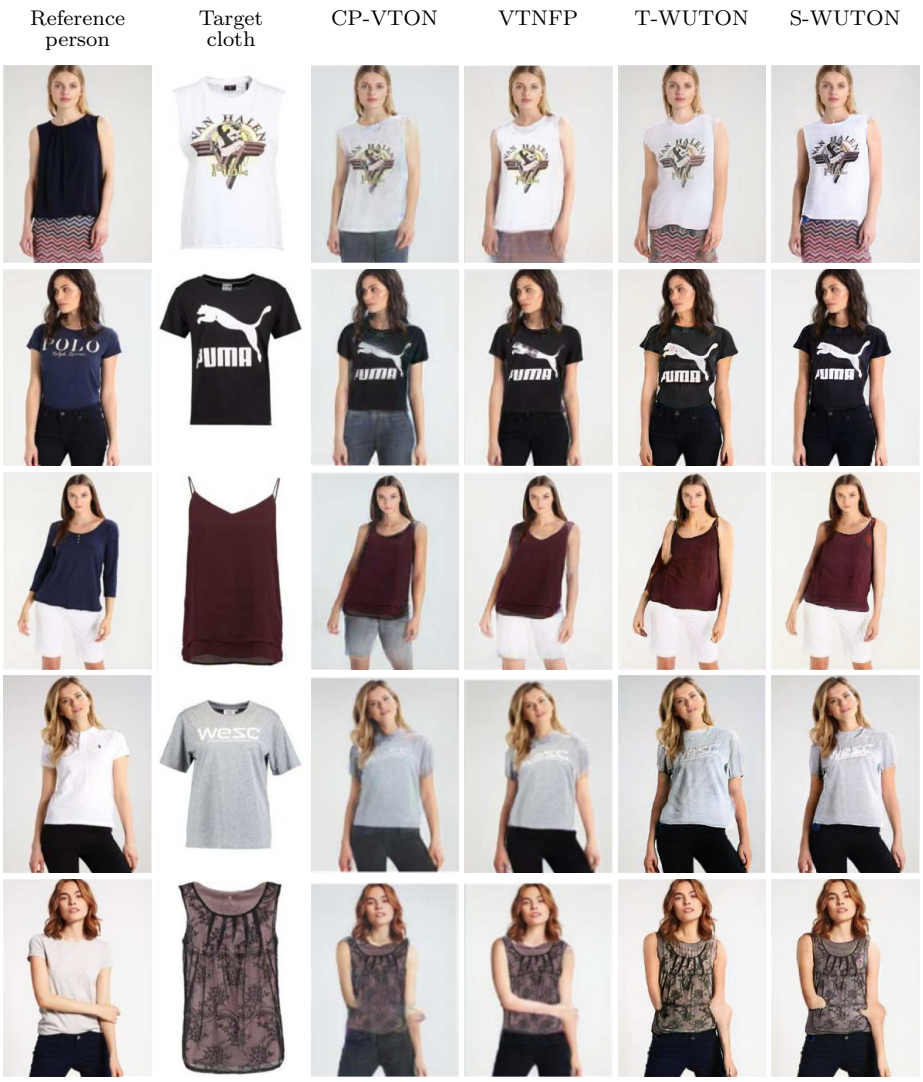


Fig. 2: Comparisons of S-WUTON and VTNFP. Images are taken from VTNFP's paper, except for T-WUTON's and S-WUTON's columns.

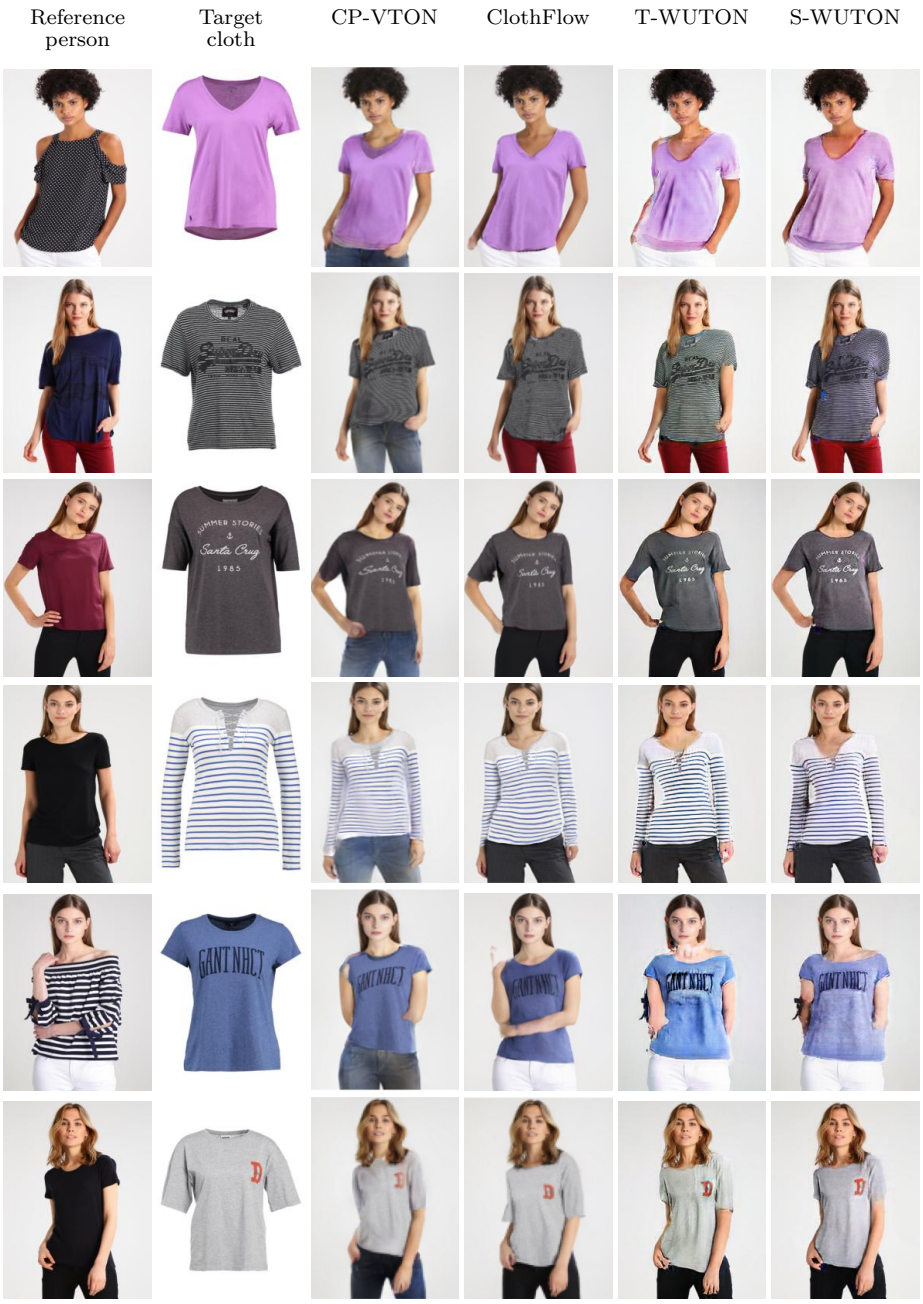
1.5 Comparisons with images from ClothFlow [2] paper

In Fig. 3, we show visual comparisons between CP-VTON, ClothFlow, T-WUTON and S-WUTON. Images from all columns except T-WUTON and S-WUTON are taken from their paper. Note that they trained their model on the original VITON dataset, that is now forbidden due to copyright issues. As mentioned in the paper, our model is trained on the dataset released in MG-VTON [1].

We can observe that S-WUTON preserves better the shape of the hands of the person (rows 6,7,12).

Compared to ClothFlow, S-WUTON handles as well complex geometric deformations. S-WUTON seems to be slightly better on stripes (rows 4 and 11). Indeed, ClothFlow uses a dense flow to warp clothes, which means the warping module warps from the source to the target pixel-by-pixel. It has thus difficulties to keep the stripes straight.

In terms of computational cost at inference time, our S-WUTON is at least 13 times faster than their model. Moreover, since their pipeline relies on human parsing and pose estimation, it is also sensitive to the errors exhibited in our paper and in Fig. 3 of the Appendix.


Reference person
Target cloth
CP-VTON
ClothFlow
T-WUTON
S-WUTON

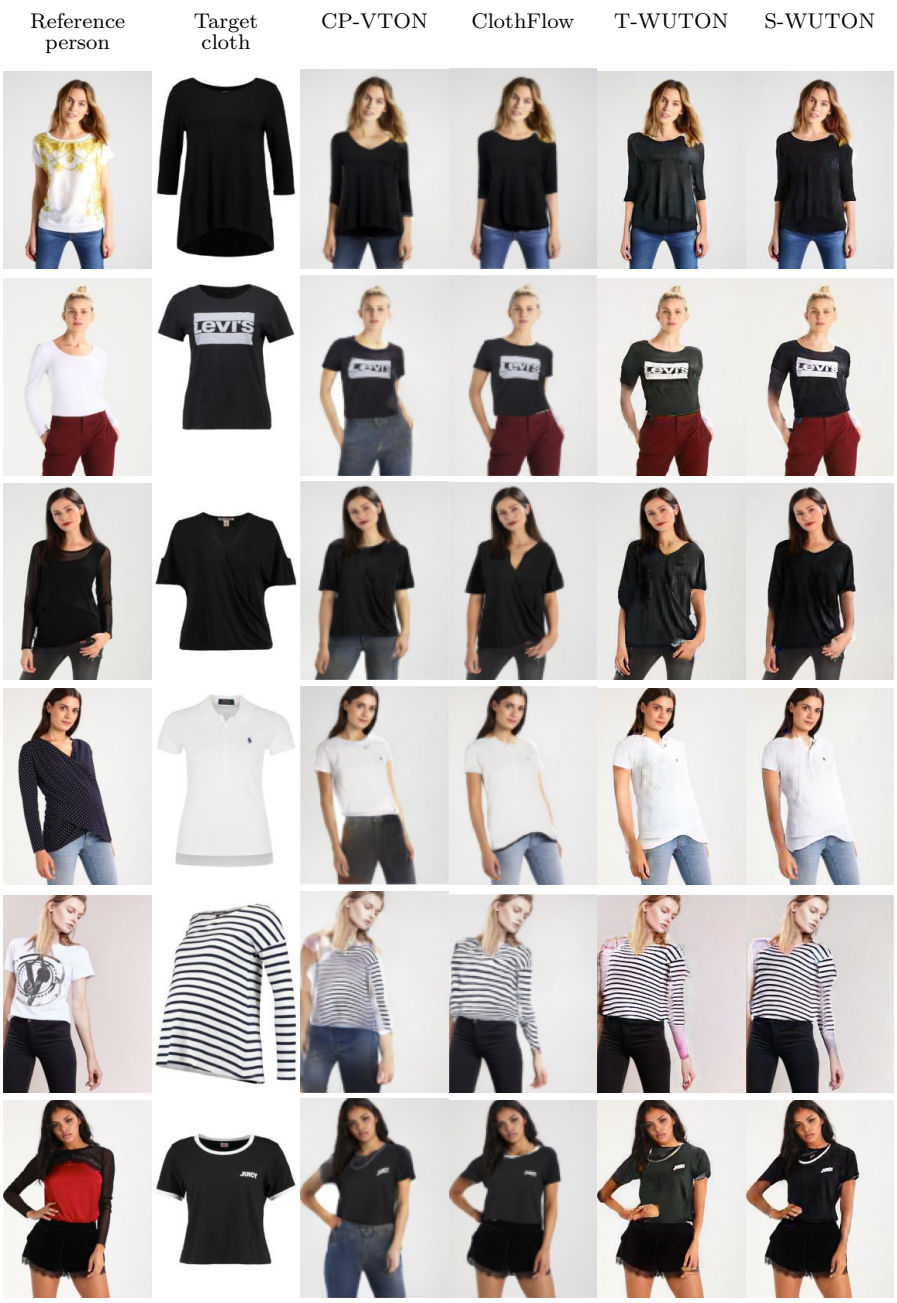


Fig. 3: Comparisons of S-WUTON and ClothFlow. Images are taken from ClothFlow's paper, except for T-WUTON's and S-WUTON's columns.

1.6 Ablation studies on T-WUTON

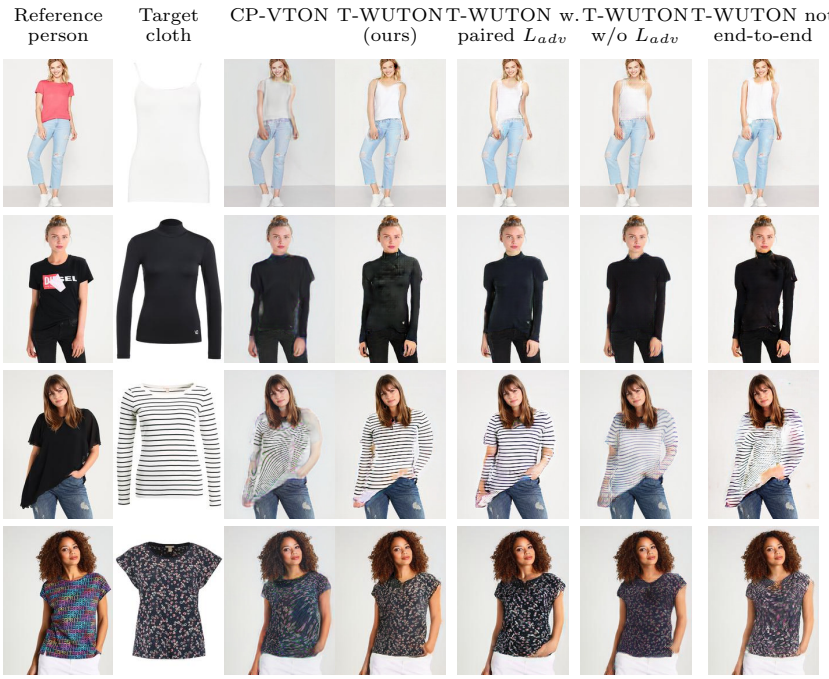


Fig. 4: Impact of loss functions on T-WUTON: The unpaired adversarial loss function improves the performance of T-WUTON in the case of significant shape changes from the source cloth to the target cloth. Specifically, when going from short sleeves to long sleeves, it tends to gum the shape of the short sleeves. With the paired adversarial loss, we do not observe this phenomenon since the case never happens during training. Without the adversarial loss, images are blurry and less sharp. Finally, the end-to-end training is key to realistic geometric deformations (see last column).

To investigate the effectiveness of T-WUTON's components, we perform several ablation studies. In Fig. 4, we show visual comparisons of CP-VTON and different variants of our approach: T-WUTON; T-WUTON with an adversarial loss on paired data (*i.e.* the adversarial loss is computed with the same synthesized image as the L1 and VGG losses); T-WUTON without the adversarial loss; T-WUTON without back-propagating the loss of the synthesized images ($L_1, L_{perceptual}, L_{adv}$) to the geometric matcher.

The results in Fig. 4 as well as FID and LPIPS metrics in Table 1 show the importance of our end-to-end learning of geometric deformations. When the geometric matcher only benefits from L_{warp}, it only learns to align c with the masked area in $p^\star$. However, it does not preserve the inner structure of the cloth. Back-propagating the loss computed on the synthesized images $\tilde{p}$ alleviates this

Method	T-WUTON	W/o L_{adv}	Paired L_{adv}	Not end-to-end
Paired L_{adv}			✓	
Unpaired L_{adv}	✓			✓
End-to-end	✓	✓	✓	
LPIPS	0.101 ± 0.047	0.107 ± 0.049	**0.099** ± 0.046	0.112 ± 0.053
SSIM	0.799 ± 0.089	0.799 ± 0.088	**0.800** ± 0.089	0.799 ± 0.089
IS	3.114 ± 0.118	2.729 ± 0.091	3.004 ± 0.091	3.102 ± 0.077
FID	9.877	13.020	**8.298**	11.125

Table 1: Ablation studies on T-WUTON. Quantitative metrics on paired setting (LPIPS and SSIM) and on unpaired setting (IS and FID). For LPIPS and FID, the lower is the better. For SSIM and IS, the higher is the better. ± reports std. dev.

issue. The quantitative results of IS and SSIM scores on the not end-to-end variant show that these metrics are less suited to the virtual try-on task than LPIPS.

The adversarial loss generates sharper images and improves the contrast. This is confirmed by the LPIPS, IS and FID metrics in Table 1 and with visual results in Fig. 4. With the unpaired adversarial setting, the system better handles large variations between the shape of the cloth worn by the person and the shape of the new cloth. On metrics in the paired setting (LPIPS and SSIM), the best model is the variant using adversarial loss on paired data, which is logical. However, visual investigation suggests that the unpaired adversarial loss is better in the real use case of our work (see Fig. 4).

References

1. Haoye Dong, Xiaodan Liang, Bochao Wang, Hanjiang Lai, Jia Zhu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
2. Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R. Scott. Clothflow: A flow-based model for clothed person generation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
3. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
4. Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
5. Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations*, 2019.
6. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
7. Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017.
8. Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.