ClothFlow: A Flow-Based Model for Clothed Person Generation
Supplemental Material

1. Network Structures

We visualize our network structures for ClothFlow's three stages in Table 1, 2, 3, respectively. In each table, the left column indicates the spatial shape of the feature map output by the corresponding right column, where

• **ResBlock down** is a 2-strided convolutional layer with 3×3 kernel followed by a residual block [9].

• **ResBlock up** is a nearest-neighbor upsampling with a scale of 2, followed by a 3×3 convolutional layer and then a residual block.

• **Skip** is a skip connection that concatenates the feature maps of an encoding layer and decoding layer with the same spatial resolution.

• Numbers before and after $\rightarrow$ are the numbers of input and output channels, respectively.

• In stage 2 (Table 2), after each residual block, a 1×1 convolutional layer changes the pyramidal feature dimension to 256. Feature pyramids are merged by addition with nearest-neighbor upsampling adjusting the spatial resolution before addition. This lateral connection is the same as the original FPN paper [10] and not shown in the table.

Architectures shown in these tables are for pose-guided person image generation. For virtual try-on task, the input, output, and image resolution will be adjusted accordingly.

2. Pose and Semantic Layout Representation

Pose Heatmap. We follow recent methods on pose-pose guided approaches [11, 3, 12] to extract 18 person keypoints for a person image and convert them to an 18-channel heatmap. Each channel represents a keypoint by filling in with ones for a circle with a radius of 5 pixels at that keypoint and zeros elsewhere. Thus, $p_t \in \mathrm{R}^{\mathrm{H \times W \times 18}}$, where H, W denote the size of the image.

Semantic Layout. The original human parser [5] produces 20 semantic labels to represent different human parts. As human parsing is a challenging task, directly using this result as a pseudo ground truth to train our conditional layout generator may yield unrealistic conditional parsing results (*e.g.*, part of pant regions may be predicted as dress or skirt). To this end, we also apply a merging process (similar to [18, 7]) to the original human parsing result, leading to a 12-channel human semantic layout where each channel corresponds to *background*, *hair*, *face*, *hat*, *tops*, *bottoms*, *shoes*, *torso*, *left arm*, *right arm*, *left leg*, *right leg*. Hence, $s_s, s_t \in \mathrm{R}^{\mathrm{H \times W \times 12}}$.

256×256	Source Image I_s Source Segment s_s Target Pose p_t
128×128	ResBlock down $(3 + 12 + 18) \rightarrow 64$
64×64	ResBlock down $64 \rightarrow 128$
32×32	ResBlock down $128 \rightarrow 256$
16×16	ResBlock down $256 \rightarrow 512$
8×8	ResBlock down $512 \rightarrow 512$
4×4	ResBlock down $512 \rightarrow 512$
8×8	ResBlock up $512 \rightarrow 512$
16×16	Skip + ResBlock up $(512 + 512) \rightarrow 512$
32×32	Skip + ResBlock up $(512 + 512) \rightarrow 256$
64×64	Skip + ResBlock up $(256 + 256) \rightarrow 128$
128×128	Skip + ResBlock up $(128 + 128) \rightarrow 64$
256×256	Skip + ResBlock up $(64 + 64) \rightarrow 12$
256×256	Softmax

Table 1: Architecture of our conditional layout generator.

In our clothing flow estimation stage, we models the flow for three clothing regions, namely hat, tops, bottoms, *i.e.*, $i = 1, 2, 3$ in Eqn. (4) and (5) in the main paper. We do not include shoes in the clothing flow since shoes have distinct appearance in different views, making it is difficult to warp them from one view to another. So, we let the network generate shoes without warping guidance.

	Source Cloth c_s
256×256	Source Cloth Seg s_s
128×128	ResBlock down $(3 + 3) \to 64$
64×64	ResBlock down $64 \to 128$
32×32	ResBlock down $128 \to 256$
16×16	ResBlock down $256 \to 256$
8×8	ResBlock down $256 \to 256$

256×256	Target Cloth Seg s_s
128×128	ResBlcok down $3 \to 64$
64×64	ResBlock down $64 \to 128$
32×32	ResBlock down $128 \to 256$
16×16	ResBlock down $256 \to 256$
8×8	ResBlock down $256 \to 256$

Table 2: Architectures of our source (top) and target (bottom) feature pyramid networks in the cascaded flow estimation network.

	Warped Cloth c_s'	Source Image I_s
256×256	Target Segment s_t	Target Pose p_t
128×128	ResBlcok down $(3 + 3 + 12 + 18) \to 64$	
64×64	ResBlock down $64 \to 128$	
32×32	ResBlock down $128 \to 256$	
16×16	ResBlock down $256 \to 512$	
8×8	ResBlock down $512 \to 512$	
4×4	ResBlock down $512 \to 512$	
8×8	ResBlock up $512 \to 512$	
16×16	Skip + ResBlock up $(512 + 512) \to 512$	
32×32	Skip + ResBlock up $(512 + 512) \to 256$	
64×64	Skip + ResBlock up $(256 + 256) \to 128$	
128×128	Skip + ResBlock up $(128 + 128) \to 64$	
256×256	Skip + ResBlock up $(64 + 64) \to 3$	
256×256	Tanh	

Table 3: Architecture of our rendering network.

3. Pose-guided Person Image Generation

3.1. Comparison with State-of-the-art

We compare ClothFlow with DSC [14], VUNET [4], DPT [13] and CBI [6] in Figure 2, Figure 3 and Figure 4. The authors of [6] kindly provided us with 176 test results of all these compared methods, and we use these results for visualization. From these figures, we can see that ClothFlow achieves better results in most of the time. In particular, ClothFlow generates structurally coherent results with texture details well preserved.

Since Soft-Gated [3] uses its unique training/test split that merely overlaps with these 176 pairs, we compare Soft-Gated [3] with ClothFlow separately in Figure 5. Soft-Gated [3] synthesizes more photorealistic faces and hairs due to its use of adversarial loss and feature matching loss [16]. ClothFlow, without adversarial training, struggles to generate these regions. However, ClothFlow is better at transferring the source image details to the target and does not suffer from the artifacts that are introduced by utilizing adversarial loss.

3.2. Ablation Study

In Figure 6, we show four more examples to verify the effectiveness of ClothFlow's key components. In the first example, our full ClothFlow warps the clothing region naturally and fills in the invisible region to address partial observability, while *w/o Layout* and *w/o Cascade* observe holes in the generated results and *w/o Flow + TPS* presents weird artifacts with textures unpreserved. In the second example, only our method successfully deforms the pant regions and other warping-based methods render pants with wrong color. In the third and forth examples, ClothFlow generates more realistic tops and pants thanks to the accurate cascaded warping.

w/o Flow does not model the geometric changes and usually generates inconsistent appearance with the source image. We find that style loss is vital for keep the original textures—*w/o Style* loses the desired style information as observed in the first two examples.

3.3. DensePose

In the main paper, we report the quantitative improvement when replacing 2D keypoints with DensePose descriptor [1]. Here, we also study the performance boost qualitatively as shown in Figure 7. DensePose has more abundant information about the body shape and pose, thus synthesizing a more accurate conditional target layout (*e.g.*, arms in the first three examples). Also, when estimation of the 2D keypoint is inaccurate or ambiguous (*e.g.*, the last four examples), DensePose generates person with realistic poses. Consequently, ClothFlowDense can obtain a higher SSIM score.

One interesting future research direction would be combining the DensePose-based texture warping [13, 6] instead of just serving DensePose as network input.

4. Virtual Try-on

It is worth noting that most of pose-guided image generation methods need pose information of the source image. For example, [14, 2] need to compute the transformation for each keypoint given its coordinates in the source and target. And DensePose-based methods [6, 13, 17] need source and target DensePose to obtain the correspondence and warp the textures. This makes them implausible when facing virtual try-on problem because in this case the source image is a product image whose pose information is not available. Differently, ClothFlow does not rely on the source pose and can address pose-guided image generation and virtual try-on in a unified framework.

Figure 8 provides more comparison against VITON [8], CP-VTON [15] and our *w/o Cascade* for virtual try-on tasks, from which we find that ClothFlow is favorable when modeling large clothing deformation. ClothFlow deforms the clothing product image more naturally (logos and graphics are more realistic, long sleeves align with arms, inner collars are warped to be invisible, *etc*.) and seamlessly renders it on the target person.

5. Limitations

ClothFlow has the following limitations:

(1) Relying on a third-party human parser, ClothFlow fails to generate realistic synthesized results when the parsing results are inaccurate (the 6th and last examples in Figure 7). Also, for complicated or less seen poses, similar unsatisfactory results are generated (the 1st and 8th examples in Figure 7. This limitation is shared by most pose-guided person image generation approaches.

(2) Without adversarial training, many non-clothing regions do not look realistic such as faces, shoes, *etc*. Some examples can be found in Figure 2, 3 and 4.

(3) Unaware of the 3D structure of human body, ClothFlow is still essentially a 2D warp-based image generation system. When the source and target images have a large view discrepancy (*e.g*., from side view to front view, or from front view to back view), it will try to directly warp the visible clothing regions of the source image to the target view, creating some unnatural generated images. See the last two rows in Figure 5.

(4) When the target image has some regions not visible in the source image, the problem becomes ill-posed. ClothFlow cannot benefit from the estimated spatial transformation for these regions, and will directly hallucinate them without rich details as shown in Figure 1.

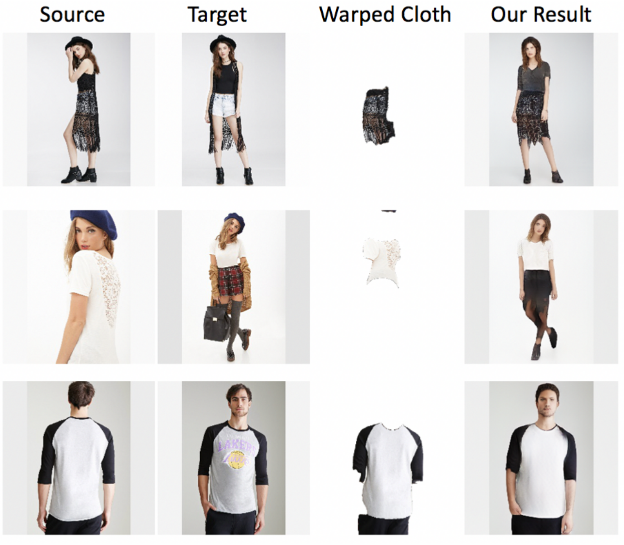

Figure 1: Failure cases when target view has some invisible regions in the source view.

References

[1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 2

[2] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018. 3

[3] Haoye Dong, Xiaodan Liang, Ke Gong, Hanjiang Lai, Jia Zhu, and Jian Yin. Soft-gated warping-gan for pose-guided person image synthesis. In *NeurIPS*, 2018. 1, 2, 7

[4] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018. 2, 4, 5, 6

[5] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *ECCV*, 2018. 1

[6] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Lempitsky Victor. Coordinate-based texture inpainting for pose-guided image generation. In *CVPR*, 2018. 2, 3, 4, 5, 6

[7] Xintong Han, Zuxuan Wu, Weilin Huang, Matthew R Scott, and Larry S Davis. Compatible and diverse fashion image inpainting. In *ICCV*, 2019. 1

[8] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *CVPR*, 2018. 3, 10

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[10] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1

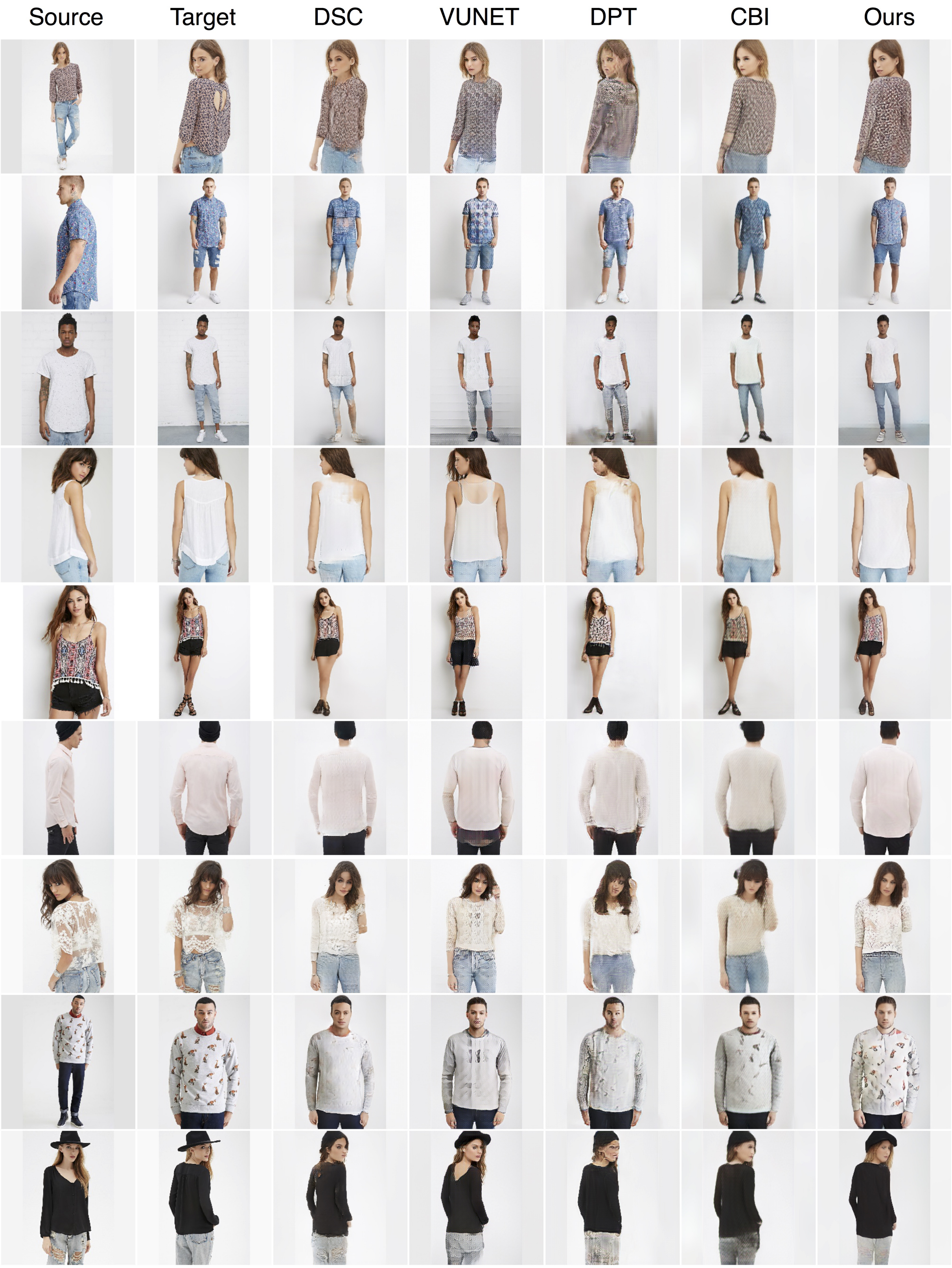

Figure 2: Comparison with DSC [14], VUNET [4], DPT [13], CBI [6] on pose-guided person image generation.

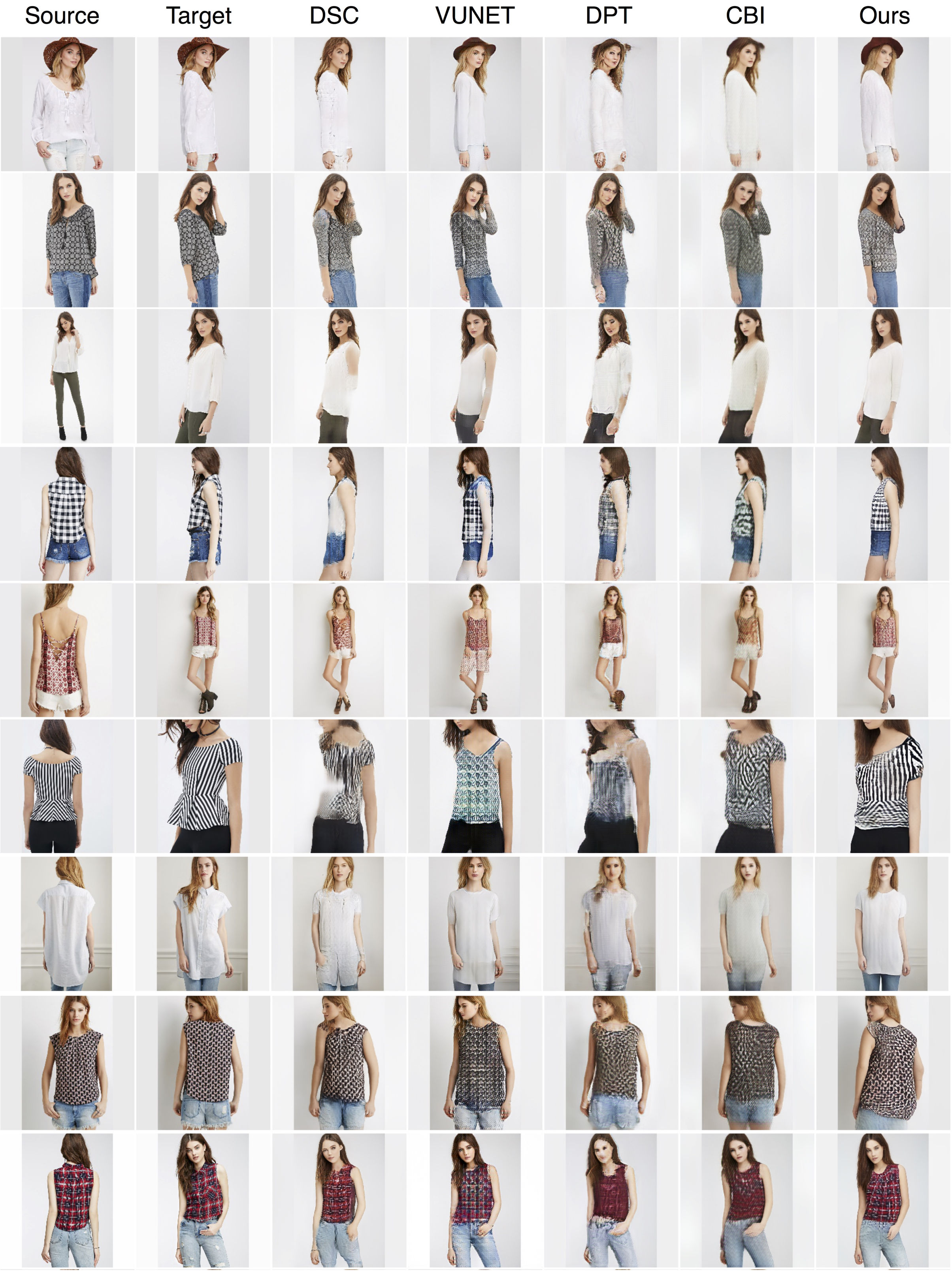

Figure 3: Comparison with DSC [14], VUNET [4], DPT [13], CBI [6] on pose-guided person image generation.

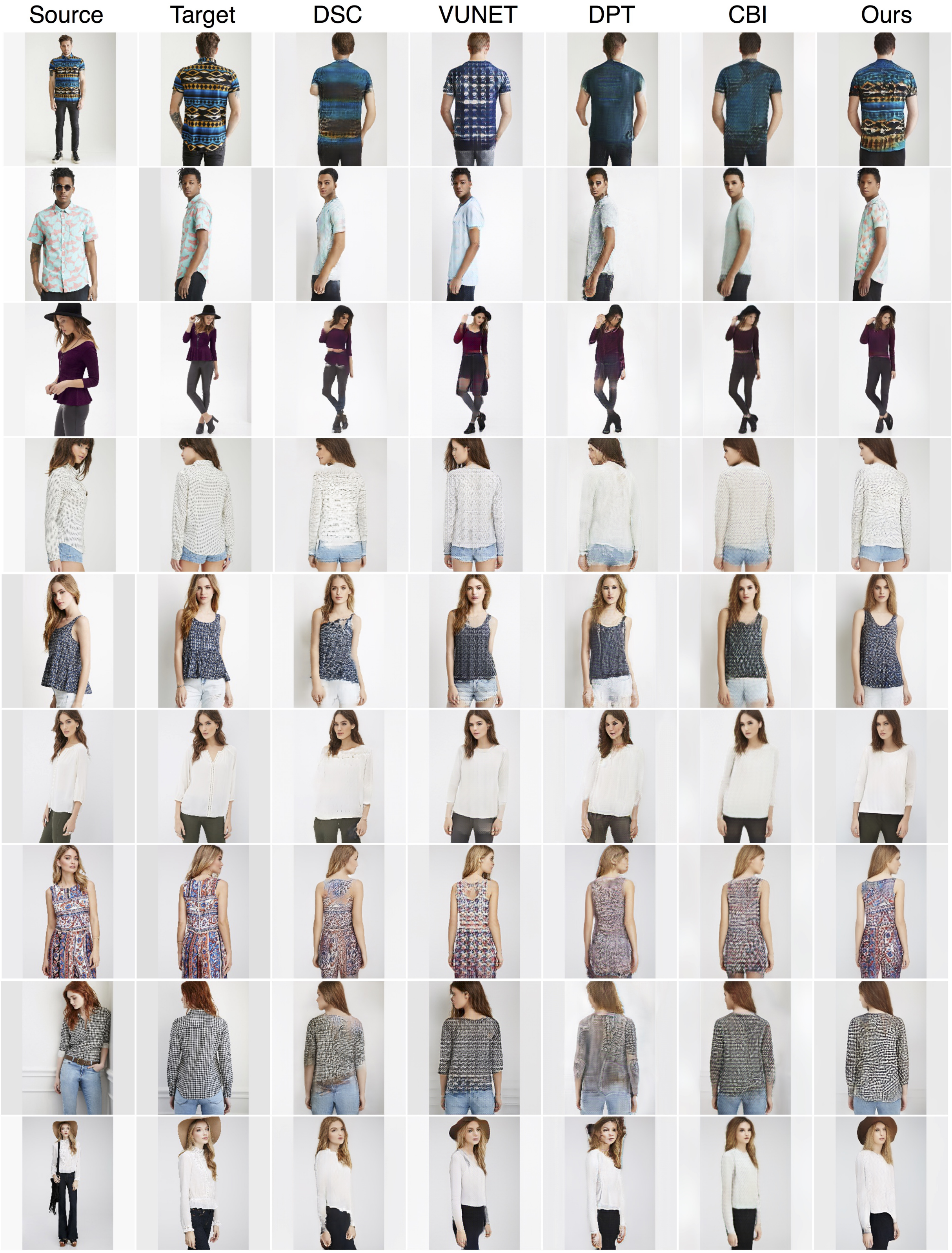

Figure 4: Comparison with DSC [14], VUNET [4], DPT [13], CBI [6] on pose-guided person image generation.

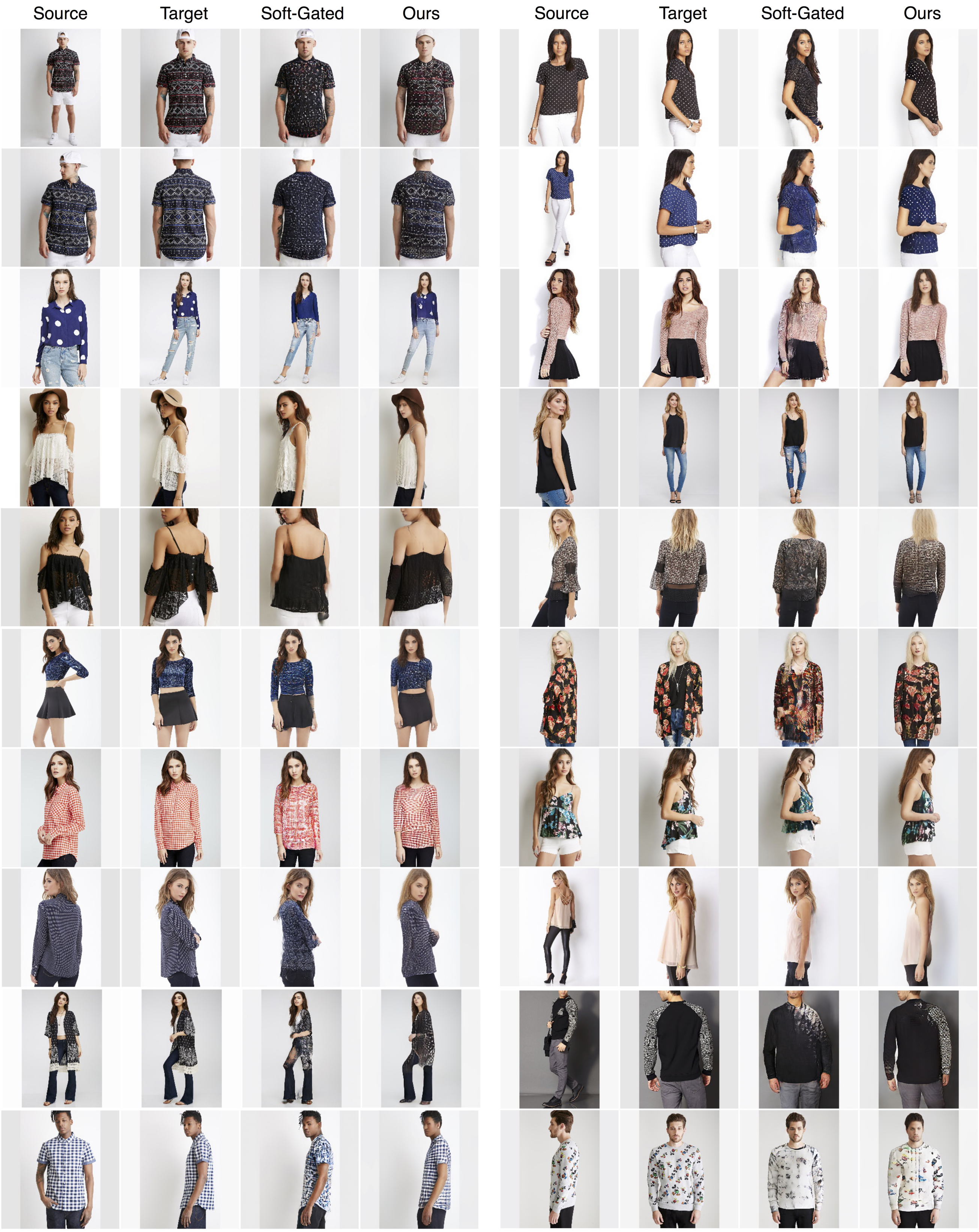

Figure 5: Comparison between ClothFlow and Soft-Gated [3].

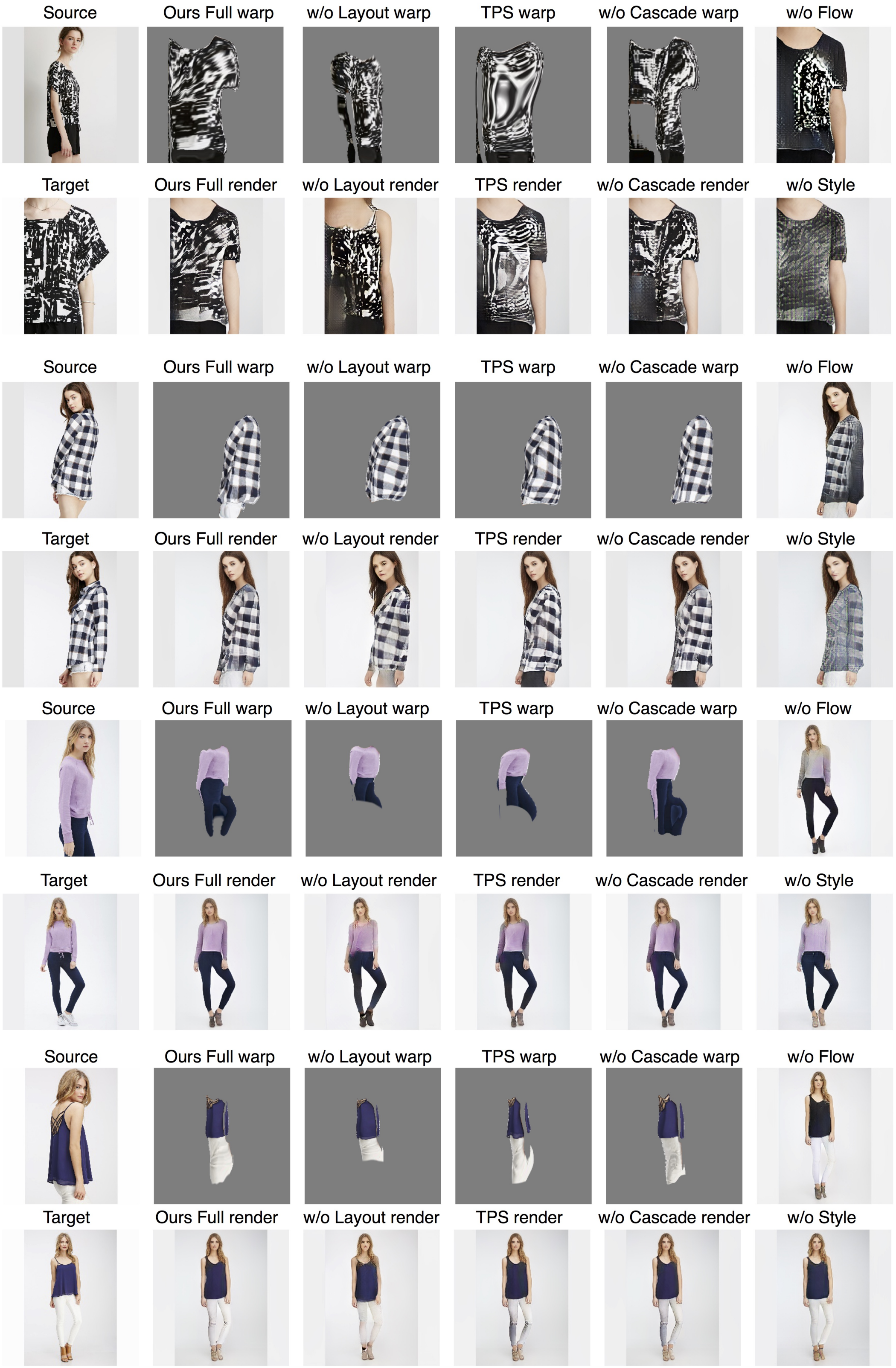

Figure 6: More visual ablation results on pose-guided person generation.

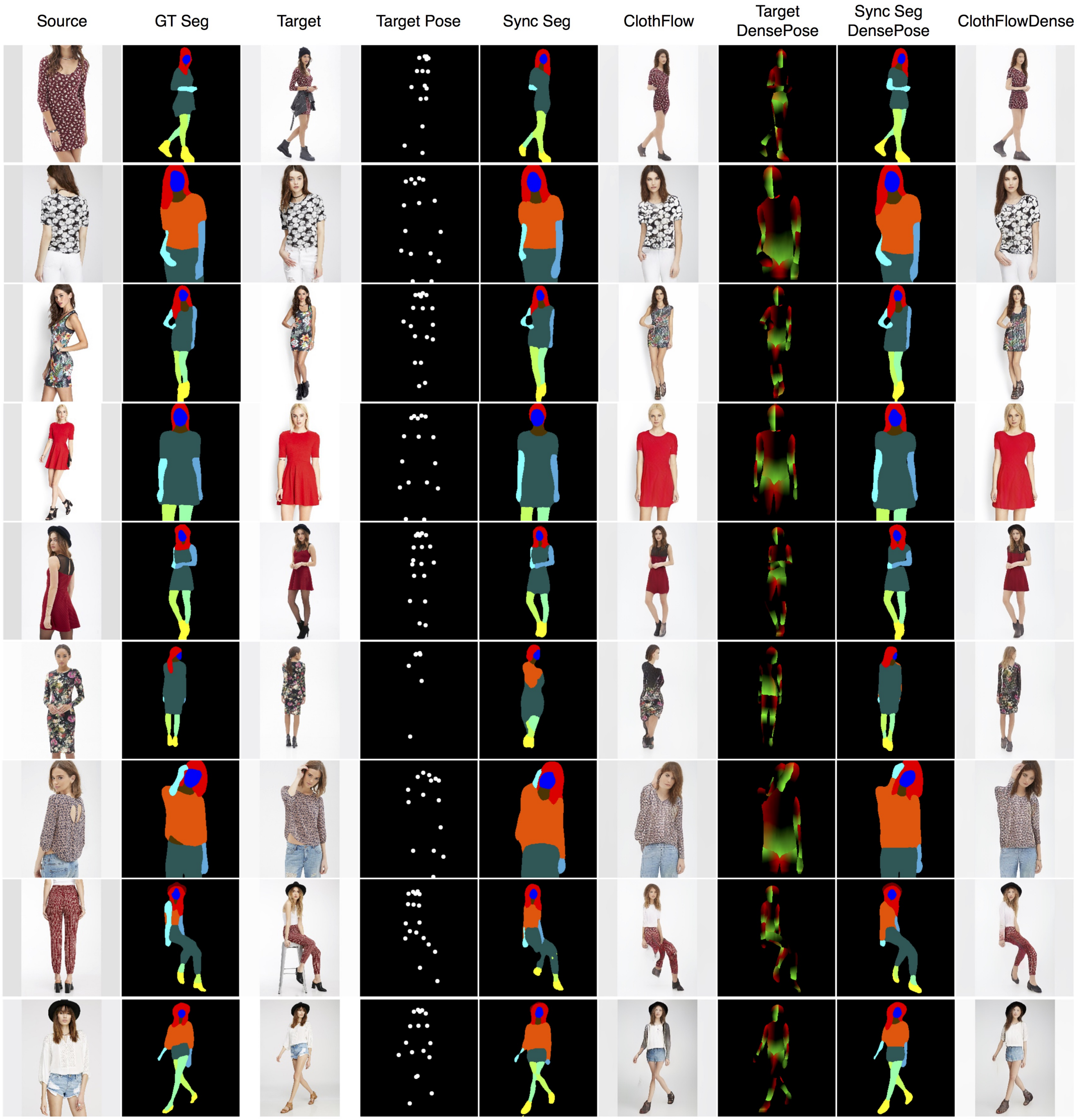

Figure 7: Comparison between 2D pose keypoints and DensePose. DensePose contains more information about the body shape and pose, leading to a better estimation of the synthetic target segmentation layout (*e.g.*, arm regions in the first three rows). DensePose generates more realistic results when estimation of the keypoint is inaccurate or ambiguous as in the last four rows.

[11] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017. 1

[12] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *CVPR*, 2018. 1

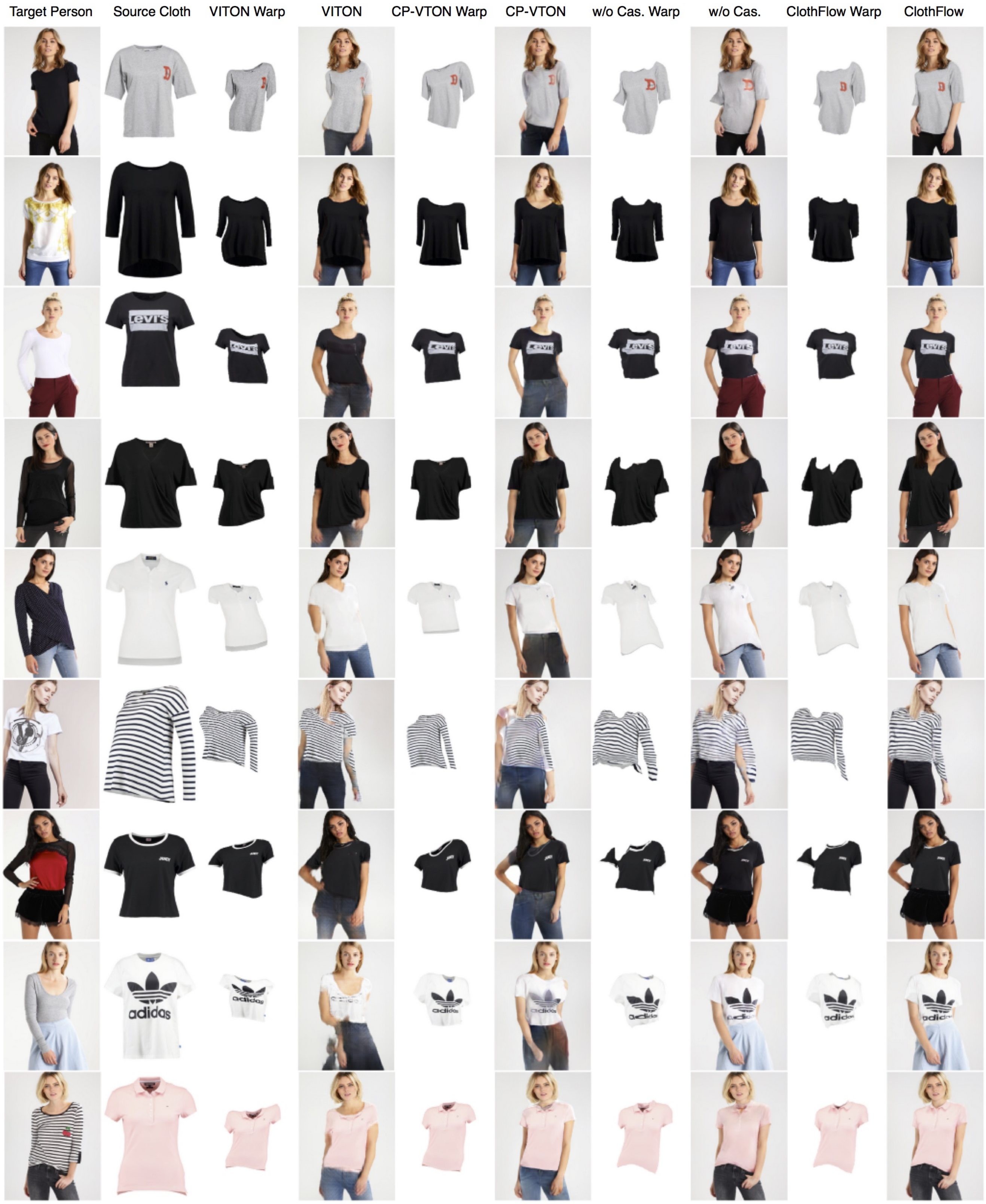

Figure 8: Virtual try-on results of VITON [8], CP-VTON[15], our *w/o Cascade* and our ClothFlow.

[13] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *ECCV*, 2018. 2, 3, 4, 5, 6

[14] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR*, 2018. 2, 3, 4, 5, 6

[15] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In *ECCV*, 2018. 3, 10

[16] 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 *CVPR*, 2018. 2

[17] Zhonghua Wu, Guosheng Lin, Qingyi Tao, and Jianfei Cai. M2e-try on net: Fashion from model to everyone. *arXiv preprint arXiv:1811.08599*, 2018. 3

[18] Shizhan Zhu, Sanja Fidler, Raquel Urtasun, Dahua Lin, and Change Loy Chen. Be your own prada: Fashion synthesis with structural coherence. In *ICCV*, 2017. 1