Controlling Virtual Try-on Pipeline Through Rendering Policies - Supplementary Materials

Kedan Li kedan@revery.ai
Jeffrey Zhang jeff@revery.ai
Shao-Yu Chang shaoyuc3@illinois.edu
David Forsyth daf@illinois.edu

Figure 1. This figure plots the ground truth control points we used during training. The control points are obtained by running a pre-trained garment key points predictor on the fashion models in a try-on dataset. As shown, the control points mark anchors such as the corner of the sleeves and trouser legs and is ordered to embed semantic meaning. In another sense, the control points are ordered and mark the garment's silhouette on the person.

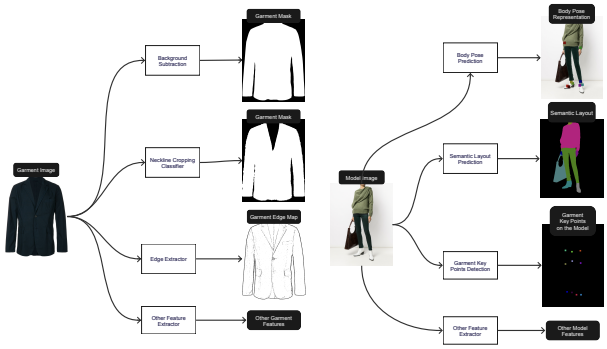

Figure 2. The figure shows the feature representations our system produces from the garment and person images.

1. Dataset

1.1. Dataset Preparation

To obtain the control points on the model images in the try-on dataset, we first apply a pre-trained garment key points prediction network trained on DeepFashion2 [8]. Examples of predicted key points are in Figure 1. Then, we convert the predicted key points into control points by merging the key points across different garment categories with the same semantic meaning. For example, we consolidated each of the shoulder control points for tops and dresses to have the same label (e.g., in Figure 1, the model's right shoulder control points will have the same label, indicated by cyan, and the model's left shoulder will have the same label, indicated by brown). To obtain style labels for Z, we use simple heuristics to label the styles: for closed vs. open outerwear, we check whether the outerwear has two large disjoint regions of similar size in the layout; for tuck vs. untuck, we threshold the height of waistline garment control points against the waist height in the body pose.

1.2. Garment Features

The neutral garment representation A consists of a neutral garment image a and many other extracted features shown in Figure 2. A neutral garment image a is an image taken in a flat-lay position or on a mannequin; many features are directly derived from a. Examples of these features include: (1) Garment Mask a_m is a binary mask separating the garment region and the background region. (2) Garment mask cropped a_c is a binary mask where the region of the garment that is supposed to be covered by the human body (e.g., the collar, the back of the dress) is cropped out. (3) Edge Map a_e is a binary edge map computed from the garment image that provides information to garment contour and shape. Finally, *other features* in Figure 2 refers to categorial metadata about the garment. Some things the metadata includes: (1) the type of the garment a_t (e.g., top, bottom, outerwear, full-body), (2) the dimensions of the garment (e.g., sleeve length, torso length, etc.), (3) garment attributes (e.g., has sleeves, has slings, etc.), etc.

Not all other features are required, and some may sometimes be unavailable. However, having more of these features available does help the network produce higher-quality outputs. Note that when applying the spatial transformation to the neural garment image, it is required to performed the same spatial transformation to the features that are directly derived from the neutral garment image.

1.3. Human Body Features

The human body representation B consists of a full body person image b (ideally) taken in a studio setting, the semantic layout (or human parsing) mask b_m, the body pose representation b_p, the garment key points computed on the person K, and other additional features.

The semantic layout mask b_m is a segmentation mask of the person wearing the garment. Like most of the other works [4–7, 9–11, 13, 15], the semantic layout mask is primarily used to occlud part of the body and guide the skin generation. The segmentation classes should at least distinguish the regions of the body, skin, different pieces of garments and shoes, and the background. In practice, we work with the following classes: background, hair, face, neckline, right arm, left arm, right shoe, left shoe, right leg, left leg, bottoms, full-body, tops, outerwear, bags, belly.

The body pose representation b_p can be different representations in the form of key points, 3d prior, or others described in [1–3, 12, 14]. Prior literature highlighted that certain pose representations are strongly influenced by the type of garment worn. As a result, we recommend simple pose representations which are less biased by the garment. In practice, we use OpenPose [14].

We use garment key points on the person K as the intermediate representation to guide garment warping and enable adjustments to the garment dimension and its position on the person. We compute the ground truth key points used during training by using a pre-trained network on DeepFashion2 [8]. During inference, the garment key points K are computed from the garment representation A and the body pose b_p.

Other categorical or numerical features include skin color, gender, person's body dimension, etc.. They are helpful but optional.

2. Virtual Try-on Pipleline

2.1. Warper

Our pipeline can work with different warper implementations (e.g. Thin-Spline Warper, Optical Flow Warper, Multiple Coordinated Affine Warper, etc.) [4–7, 9–11, 13, 15]. In our experiment, we adopt the flow warper formulation from [5] but use OpenPose [2] instead of DensePose [1] as the body presentation, because Li *et al.* [11] demonstrated that DensePose representations are often biased by the garments worn on the person. We use the warping loss of the prior work [5], which minimizes the difference in appearance between the warped garment and the region of the warp on the person.

2.2. Split Outerwear

Splitting outerwear requires cutting a neutral garment into two regions and warping each region separately. In this scenario, we divide the garment representation of outerwear into left garment A^l and right garment A^r. The control points are also divided into left component K^l and right component K^r. The warper W predicts the spatial transformation parameter for the left side as $\theta^l = W\left(b_p, A^l, K^l\right)$ and the right side as $\theta^r = W\left(b_p, A^r, K^r\right)$. Finally, both sides of the warps are merged into a single warped image and fed into the image generator G_I. The way each side of the outerwear drapes is guided by the corresponding control points, as shown in Figure 3.

Neural Networks:

R_c	Garment Key Points Regressor
G_L	Layout Completion Network
G_I	Image Generator Network
W	Warper Network

Feature Sets:

A	Feature representations for a garment
A^w	Feature representations for a warped garment
B	Feature representations for a person image
K	Garment key points on the person
Z	Control parameters for R_c
θ	Spatial transformation parameters

2D Tensor Attributes:

a	neutral garment image
a_m	garment foreground mask
a_c	garment mask with only visible regions
a_e	garment edge map
a^w	warped garment image
a^w_m	warped garment mask
a^w_c	warped garment mask with only visible regions
a^w_e	warped garment edge map
b	full-body image with person wearing garments
b_m	semantic layout mask of the person wearing garment
$\hat{b_m}$	occluded semantic layout mask
b_o	occluded person image
b_g	to try-garment layout mask
b^g_m	semantic layout mask without the to try-garment
b_p	body pose representation

Numerical and Categorical Features:

a_t	garment category
k_i	key point
x_i	horizontal coordinate of a key point
y_i	vertical coordinate of a key point
i	ith item in the set
n	total number of items in the set
λ_i	training loss hyper parameters
N	batch size
W	width of 2D tensor
H	height of 2D tensor

Functions or Logical Operations:

f_o	producing the occluded mask for semantic layout

Misc:

'	prediction made by a network from input data
"	prediction made by a network from predicted data
$\mathcal{L}$	training loss

Table 1. This table contains all the notations used to describe our method.

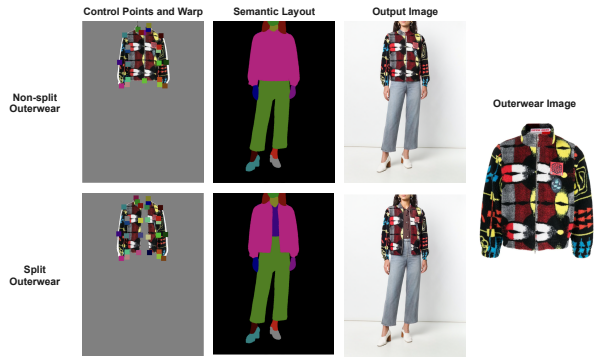

Figure 3. The figure shows examples of the same outerwear worn zipped or unzipped. Note that we use a slightly different set of control points to signal an open jacket. Both the computed warp and the predicted layout exactly follow the control points. The neutral garment is separated into two pieces and warped separately.

Figure 4. This figure shows the control points used for different tucks. We see that the shape of the shirt follows the silhouette of the control points.

Figure 5. This figure shows an interpolation of the control points. We move the control points by a small offset every step to gradually open the outerwear. The drape of the outerwear closely follows the control point in the entire process, demonstrating that the control points are highly effective in controlling the garment.

2.3. Obtaining the Occluded Person Image

The occluded person image b_o is created by applying the occluded semantic layout mask $\hat{b_m}$ produced by f_o on the person image b. We remove the garment mask from the semantic layout b_m because the garment warp may not exactly match the shape of the mask during inference. Removing the garment mask allows G_I to figure out the garment shape through the warped garment features A^w and yields better results. This procedure was also adopted by other VTON works [5, 10].

2.4. Control Parameters Z

In our current pipeline, $z_0 \in Z$ controls the garment category; $z_1 \in Z$ controls whether outerwear is open or closed; $z_2 \in Z$ controls whether a top is tuck or untuck. More control parameters can be introduced as needed.

3. User Study Analysis

The user study evaluates both the accuracy and consistency of the images generated using different rendering policies. In the first study, we show each participant 22 sets of examples. Each example contains two distinct generated images with a highlighted garment. Participants should determine if the highlighted garment from both images are the same garment. In the second study, we first prime the participants with a list of styles each corresponding to a different rendering policy. For example, the styles for tops-bottom outfits include untuck, full tuck, front tuck, side tuck and half tuck. For each style, we also show the user a real life example of the style as reference. There are 44 respondents in total. We attached the raw results in a separate folder.

Results show that users are able to identify the same garment worn in different styles quite consistently. This sug-

Figure 6. This figure shows a simple policy that shifts the waistline height of skirts. Note that the garment length remains the same.

gests that our method is able to preserve garment identity. When it comes to selecting styles, we noticed that users sometimes have difficulties distinguishing between certain styles. We break down the accuracy per style in Table 3. The accuracy per style is computed using all the questions in which the style is the correct answer. The results show that participants have difficulty recognizing certain styles (such as draping outerwear vs. one-side draping outerwear), but can consistently recognize other styles. This implies that our method is able to drape the garments in different and meaningful ways consistently, but people sometimes disagree on how certain styles should be named.

Style	Accuracy (%)
Untuck	93.67%
Full tuck	97.6%
Front tuck	83.35%
Side tuck	90.5%
Half tuck	90.5%
Unsplit outerwear	84.5%
Split outerwear	80.97%
Draping outerwear	45.2%
One-sided draping outerwear	65.87%

Figure 7. More outfit coordination comparisons.

References

[1] Rıza Alp Guler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *The IEEE Conference on Computer Vision and Pattern Recogni-*

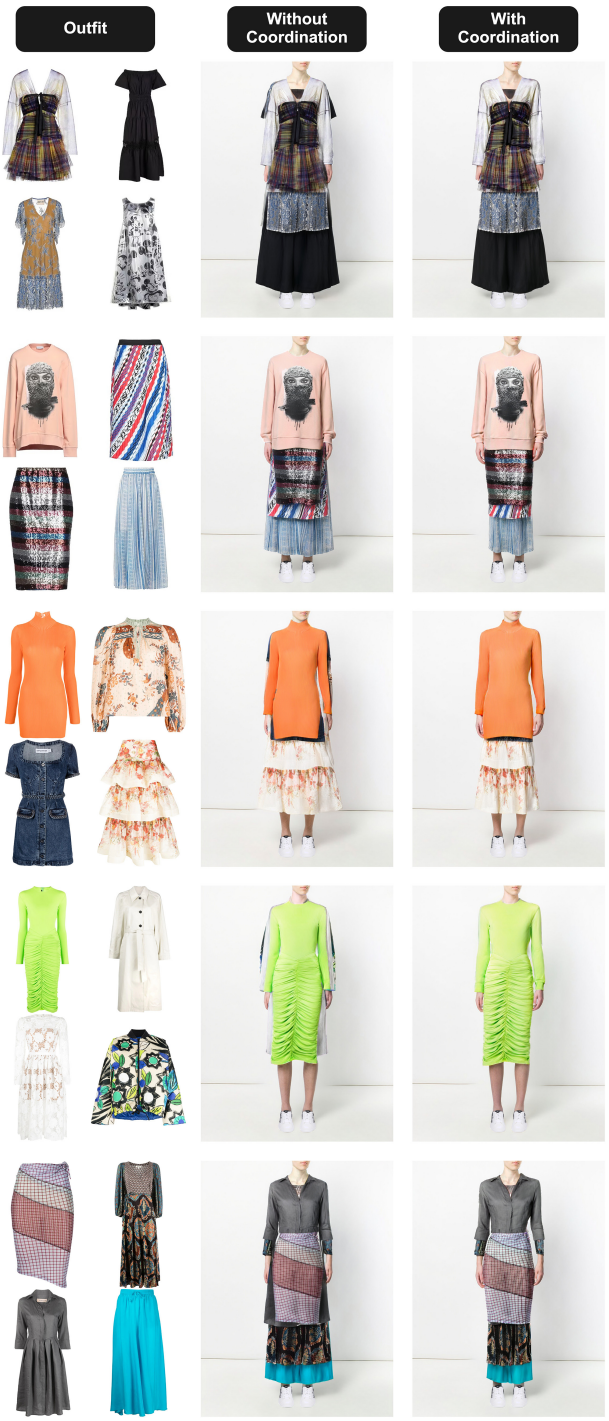

Figure 8. More outfit coordination comparisons.

Figure 9. More outfit coordination comparisons.

tion (CVPR), June 2018. 2

[2] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 2

[4] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *CVPR*, 2021. 2

[5] Ayush Chopra, Rishabh Jain, Mayur Hemani, and Balaji Krishnamurthy. Zflow: Gated appearance flow-based virtual try-on with 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3

Figure 10. More outfit coordination comparisons.

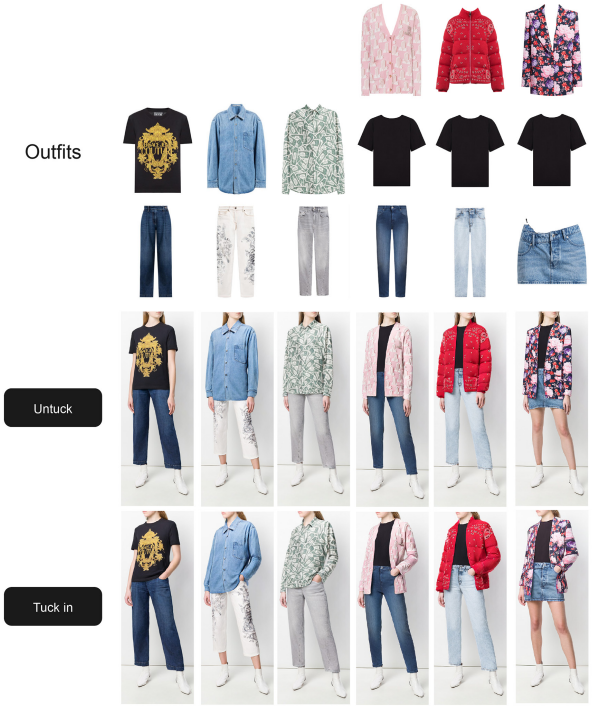

Figure 11. This figure shows the same outfit rendered in different hand postures. Note that in the second row, when the hand is supposed to be in the pocket, our method can reliably control the drape and add appropriate folds on the garment around the hand.

[6] Chongjian Ge, Yibing Song, Yuying Ge, Han Yang, Wei Liu, and Ping Luo. Disentangled cycle consistency for highly-realistic virtual try-on. In *CVPR*, 2021. 2

[7] Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. Parser-free virtual try-on via distilling appearance flows. In *CVPR*, 2021. 2

[8] Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *CVPR*, 2019. 1, 2

[9] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R. Scott. Clothflow: A flow-based model for clothed person generation. In *ICCV*, 2019. 2

[10] Sen He, Yi-Zhe Song, and Tao Xiang. Style-based global appearance flow for virtual try-on. In *CVPR*, 2022. 2, 3

[11] Kedan Li, Min Jin Chong, Jeffrey Zhang, and Jingen Liu. Toward accurate and realistic outfits visualization with attention to details. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[12] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 2

[13] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, and Liang Lin. Toward characteristic-preserving image-based virtual try-on network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[14] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 2

[15] 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 *CVPR*, 2020. 2