Toward Accurate and Realistic Outfits Visualization with Attention to Details

Kedan Li[1,2] Min Jin Chong[1,2] Jeffrey Zhang[1,2] Jingen Liu[3]
{kedan, minjin, jeff}@revery.ai, jingenliu@gmail.com
[1]University of Illinois, Urbana Champaign. [2]Revery AI Inc. [3]JD AI Research.

Abstract

Virtual try-on methods aim to generate images of fashion models wearing arbitrary combinations of garments. This is a challenging task because the generated image must appear realistic and accurately display the interaction between garments. Prior works produce images that are filled with artifacts and fail to capture important visual details necessary for commercial applications. We propose Outfit Visualization Net (OVNet) to capture these important details (e.g. buttons, shading, textures, realistic hemlines, and interactions between garments) and produce high quality multiple-garment virtual try-on images. OVNet consists of 1) a semantic layout generator and 2) an image generation pipeline using multiple coordinated warps. We train the warper to output multiple warps using a cascade loss, which refines each successive warp to focus on poorly generated regions of a previous warp and yields consistent improvements in detail. In addition, we introduce a method for matching outfits with the most suitable model and produce significant improvements for both our and other previous try-on methods. Through quantitative and qualitative analysis, we demonstrate our method generates substantially higher-quality studio images compared to prior works for multi-garment outfits. An interactive interface powered by this method has been deployed on fashion e-commerce websites and received overwhelmingly positive feedback.

1. Introduction

While e-commerce has brought convenience to many aspects of our lives, shopping online is difficult for fashion consumers who want to try-on garments and outfits before deciding to buy them [52]. In most online shopping experiences, we are only given a neutral product image of a garment or a single example of a model wearing the garment, and users have to imagine how the garment would look in different settings (e.g. with different garments, on different models etc.). As a result, there has been a considerable amount of literature on synthesizing people wearing garments [18, 53, 46, 11, 16, 58, 41, 25, 23].

Figure 1. Our method takes in a model image and multiple neutral garments images as inputs, and generates a high quality image of the selected model wearing the garments. **Pay careful attention to details** of the garment properties that are accurately portrayed (e.g., the patterns on the dress (A-1), the unicorn and the string (C-2), the hemline (C-2), buttons (B-1, D-2), and the lengths of the garments); the interaction between multiple garments has been captured (e.g., the collar and sleeve coming out of the sweater (A-1), the open outerwear cast shading (B-1, C-2) to the garment beneath); the interaction between the garment and the person is natural (e.g., the loose sleeves, the folds by the arm (D-2), and the shadows casted on the leg by the dresses); and skin is generated realistically (B-1). See image without bounding box in Appendix.

Three natural cases arise when shopping online. A user may want to see **(a) any image of a model** wearing a chosen set of garments (outfit) to visualize a combination; **(b) any image of themselves** wearing the outfit to see how the garments interact; and **(c) an image of themselves** wearing the outfit (the VITON case [18, 53, 46, 11, 16, 58, 25, 23]). In all cases, users expect the image to capture the visual features of the garments and the physical interactions between

them. However, current methods have problems capturing details of shading, texture, drape and folds. Getting these right is crucial for shoppers to make purchase decisions.

In this work, we introduce a variety of innovations that substantially improve upon the synthesis of details (Figure 1). Our proposed method not only produces accurate textures, necklines, and hemlines, but also can drape multiple garments with realistic overlay and shading. The drape can adapt to the body pose and generate natural creases, folds, and shading. Skin and background are also generated, with appropriate shadows casted from the garments (Figure 1). Our method significantly outperforms prior work in multi-garment image synthesis as shown in Figure 9.

While other virtual try-on (VITON) methods [18, 53, 46, 11, 16, 58, 23] focused on **single** garment try-on, Neuberger *et al*. proposed O-VITON [41], which transfers **multiple** garments from model to model. In comparison, our system takes garments from neutral garment photographs and transfers them to a model. This distinction is commercially important because it is easier and cheaper to obtain neutral pictures. The formatting is also consistent across different sites, meaning no extra work is required for the merchants. Also, O-VITON [41] encodes garments into **feature vectors** and broadcasts the vectors onto a layout to produce the image. Such a formulation can handle complex garment shapes (a major difficulty for multi-garment try-on) but results in a loss of spatial patterns (e.g., logos, prints, buttons), making it hard to synthesize texture details accurately. In contrast, other VITON literature [18, 53, 16, 58, 23] uses **warping**, which faithfully perseveres details. However, they only demonstrate success with warping single garments of simple shapes (mostly). Warping multiple garments with complicated shapes has not yet been achieved.

In this work, we directly address the challenge of warping multiple garments, while also being able to accurately transfer textures between complicated garment shapes (Figure 1). Our procedure uses multiple warps, which can handle (say) open jackets, and can generate buttons, zippers, logos, and collars correctly (Figure 2). The warpers are trained end-to-end with the generator and learn to coordinate through a cascading loss, which encourages subsequent warps to address errors made by earlier warps. Using multiple coordinated warps produces substantial quantitative and qualitative improvements over prior single-warp methods [18, 53, 11, 16, 58, 25].

Finally, because publicly available try-on datasets do not contain rich garment categories, we test on a dataset with all available garment categories from multiple fashion e-commerce websites. Evaluation on this new dataset shows that using multiple warps consistently outperforms single warp baselines in this new setting, demonstrated both quantitatively (Table 3) and qualitatively (Figure 8). Our try-on system also produces higher quality images compared

Figure 2. We show a sequence of visualizations for the same outfit generated on different reference models. Our generation method is able to adapt to a diverse set of poses, skin-tones, and hand positions. When the hand is in the pocket, the jeans plump up and connect seamlessly with the jacket (Pose 2 & 5).

to prior works on both single and multi-garment generation (Table 1 and 2, and Figure 9). Furthermore, we introduce a procedure for matching garment-pose pairs, which yields significant improvement for both our and previous image generation pipelines in scenarios **(a) and (b)** (Table 2). Lastly, we conduct a user study comparing our generated images with real commercial photos, simulating the effectiveness of e-commerce sites replacing real photographs of models with our synthesized images. Results show over 50% of our synthesized images were thought to be real even with references to real images (Table 4). Furthermore, our method is fast enough to integrate with interactive user-interfaces, where users can select garments and see generated visualizations in real-time. A live demo of an virtual try-on shopping interface powered by our method is publicly available [10].

As a summary of our contributions:

- We introduce OVNet - the first multi-garment try-on framework that generates high quality images at latencies low enough to integrate with interactive software.
- We are the first warping-based try-on method that supports multi-garment synthesis on all garment types.
- We introduce a garment-pose matching procedure that significantly enhances our method and prior methods.
- Our results strongly outperform prior works, both quantitatively and qualitatively.
- We evaluate on a dataset with all available garment categories from multiple fashion e-commerce sites, and show that our method works with all categories.

2. Related Work

There are multiple ways to tackle virtual try-on. One solution is to use 3D modeling and rendering [8, 15, 43], but obtaining 3D measurements of the garments and users

[10] https://demo.revery.ai

Figure 3. The figure shows a sequence of outfit visualizations produced by our method on two different models. Our method can modify one garment at a time, leaving the rest of the image untouched. The details of the garments' shape are accurately represented (e.g., neckline shape, skirt length, pant width, etc.) and consistent on both models. The garment interactions of the top (hanging or tucked-in) also vary between poses.

is difficult and costly. A more economic and scalable approach is to synthesize images without 3D measurements. We discuss the variations of this approach in detail.

Image synthesis: Spatial transformer networks estimate geometric transformations using neural networks [24]. Subsequent work [31, 46] shows how to warp one object onto another. Warping works with images of rigid objects [28, 35] and non-rigid objects (e.g., clothing) [18, 13, 53].

In contrast to using a single warp with high degree of freedom, our work coordinates multiple spatial warps to support garments of complex shape. We use U-Net to combine multiple warps into a single image. U-Net is commonly used for inpainting methods, which tackle filling in missing portions of an image [57, 36, 60, 59]). Han *et al*. [17, 61] also show inpainting methods can complete missing clothing items on people.

Generating clothed people: Zhu *et al*. [66] uses a conditional GAN to generate images based on pose skeletons and text descriptions of garments. SwapNet [45] learns to transfer clothes from person A to person B by disentangling clothing and pose features. Hsiao *et al*. [21] learns a fashion model synthesis network using per-garment encodings to enable minimal edits to specific items. Recently, Men *et al*. [38] proposed a person image synthesis method, controllable through interpolating style and pose representations. These methods use feature vectors as visual representations, and thus cannot preserve geometric patterns (e.g, logo, prints). Our method warps garments and directly uses the warped images to generate our result.

Garment & body matching underlie our method to match garments to models. Tsiao *et al*. [20] learns a shape embedding to enable matching between human bodies and well-fitting clothing items. Prior work estimates the shape of the human body [3, 30], clothing items [10, 27] and both [40, 47], through 2D images. The DensePose [1] descriptor helps model the deformation and shading of clothes and has been adopted by recent work [42, 14, 56, 62, 7, 61].

Virtual try-on (VITON) maps a single garment onto a model image. VITON [18] first proposed using TPS transformation to create a warp, followed by a generation network to synthesize the final output. CP-VTON [53] improves this method by using a differentiable component for TPS transformation. Han *et al*. [16] uses a flow estimation network to enable more degrees of freedom for the warp. Issenhuth *et al*. [23] proposed a teacher-student training paradigm to warp without relying on masks. To enable shape changes (e.g., short sleeve to long sleeve), a common procedure has been to predict a semantic layout of body segments and clothes to assist with image generation [58, 25, 63, 44, 16]. More recent works proposed architectural improvements toward better preservation of details [54, 44] and adding adversarial training during the refinement phase to improve image realism [11, 63, 58, 44]. Others followed similar procedures [51, 22, 2]. The virtual try-on task has also been extended to multi-view scenarios and videos [13, 12]. In summary, recent work in VITON managed to preserve garment details, but only for **single garment**, with **simple shapes** (mostly tops).

Outfit try-on: Neuberger *et al*. [41] proposed a virtual try-on method that works for multiple garments. The method relies on visual feature vector encoding rather than warping, which falls short in preserving textures comparing to other VITON methods. To make up for deficiencies, they proposed an online optimization step that requires fine-tuning a generator using a discriminator for every query. Performing such an operation is massively expensive (requires multiple rounds of gradient computation and back-propagation), making it unrealistic to respond to user queries in real-time. In comparison, our method produces images of significantly better quality (Figure 9) and requires much less computation (<2s latency on a K80).

3. Outfit Visualization Net

We propose Outfit Visualization Net (OVNet) to generate images of a model (person) wearing multiple garments (outfit), faithfully capturing the garments details and the interactions between them. OVNet consists of two components trained separately: a Semantic Layout Generator G_{layout} and a Multi-Warps Garment Generator $G_{garment}$.

Semantic Layout Generator G_{layout} predicts semantic layout m' (in the form of segmentation map) conditioned on a garment image x, a pose map p of the model and an incomplete layout m_i (more details in appendix). This in-

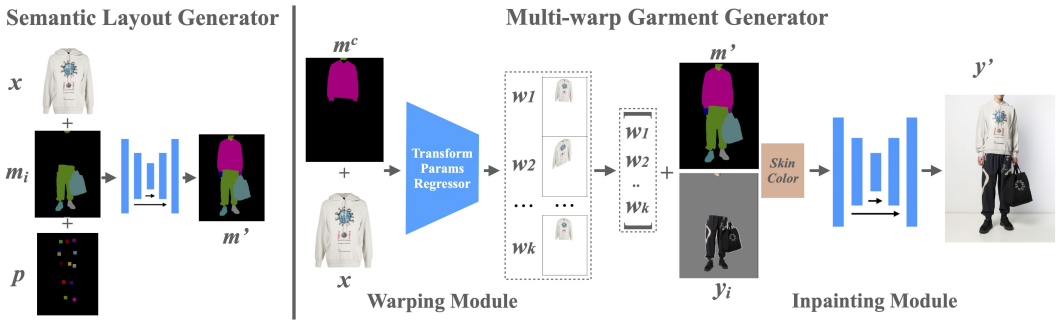

Figure 4. **Outfit Visualization Net**, which synthesizes an image of a model wearing multiple garments, consists of two components. The Semantic Layout Generator G_{layout} (left) takes in the garment image x, the pose representation p and an incomplete semantic layout m_i, and learns to reconstruct the ground truth layout m. The multi-warp garment generator $G_{garment}$ (right) has two modules. The **warping module** is a spatial transformer that takes in the garment image x and its semantic layout m^c and regress k sets of transformation parameters $\theta_1..\theta_k$. It then samples k warps $w_1..w_k$ where $w_1 = \mathcal{W}(x, \theta_1)$, using the predicted transformations. The **inpainting module** takes in the predicted warps $w_1..w_k$, the full semantic layout m', the skin color channel s (median color of the face) and the incomplete model image y_i and generates the final image y' of the model wearing garment x. Two modules are trained jointly.

complete layout m_i hides relevant semantic layout classes. For example, when generating the incomplete layout m_i for a top, we take the ground truth layout m and set the top, neckline, and arm classes to the background class. The generated layout m' is then used to guide the image generation.

Multi-Warps Garment Generator $G_{garment}$ takes in the garment image x, the predicted full layout m' and the model image y, and produces an image y' with model y wearing garment x. $G_{garment}$ only modifies one garment on the model at a time. Thus, garments of other categories remain unchanged from y to y'.

Using our formulation, synthesizing an outfit requires multiple sequential operations, with each operation swapping a single garment. Compared to Neuberger *et al.*'s [41] formulation, which is forced to generate a complete layout per inference, our formulation enables users to modify a single garment at a time, leaving the rest of the image untouched (Figure 3). Having this property benefits the user experience, as most people modify an outfit one piece at a time. The proposed method can be adopted to all application scenarios **(a), (b), and (c)** (from the Intro 1).

3.1. The Semantic Layout Generator

When synthesizing a person image, it is common practice to produce a semantic layout as structural constraints can guide the image generation [11, 33, 21, 66, 16, 58] and we follow a similar procedure. To train the layout generator, we obtain pairs of garment images x and model images y wearing x. From y, we obtain the semantic layout m using off-the-shelf human parsing models [34] and the pose map p using OpenPose [55, 5, 50, 6] (Figure 4 top left). Based on the garment category of x, we produce an incomplete layout m_i by setting the garment prediction classes as the background class. A full list of semantic categories and the detailed procedure for producing the incomplete layout m_i for different categories of garments are in Appendix.

The layout generator takes in the incomplete layout concatenated with the pose and the garment as input, and learns to predict the original layout $m' = G_{layout}([x, m_i, p])$. Because skip connections propagate information from the input to the output, we use a U-Net architecture to retain information from m_i in the output m'. The network is trained using a pixel-wise cross-entropy loss and a LSGAN [37] loss to encourage the generated semantic layouts to resemble real semantic layouts. The total training loss for G_{layout} can be written as

$$\mathcal{L}_{layout} = \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{GAN} \quad (1)$$

where λ_1 and λ_2 are the weights for each loss. Because the argmax function is non-differentiable, we adopt the Gumbel softmax trick *et al.* [26] to discretize the layout generator's output such that the gradient generated by the discriminator can flow back to the generator.

During experiments, we observed that the type of garment a model is wearing greatly influences pose prediction results, as in Figure 5. For example, between models with highly similar poses, one wearing a jacket and another one wearing a t-shirt, we observe vastly different pose predictions. Also, because we train the network to reconstruct the ground truth semantic layout conditioned on garment and pose, the pose representation may impose a prior on the type of garment to expect. This sometimes leads to errors during inference. As in Figure 6, when there is a mismatch between the provided garment (a tank) and what the pose representation implies (a jacket), the layout generator may output a layout that doesn't respect the garment shape. In Section 4, we propose a garment-pose matching procedure to alleviate this issue.

3.2. Multi-Warps Garment Generator

Our garment generation network $G_{garment}$ (Figure 4 right) takes in a garment image x^c of class c (write as x for simplicity), a model image y and a segmentation mask

Figure 5. We notice that the human pose annotation from OpenPose embeds information differently depending on the type of garment. For example, the pose predictor consistently predicts wider distance between shoulder and elbow anchor for models wearing coats (3, 4) than models wearing shirts (1, 2), despite both models having similar posture and body shape. This implies that the pairing between pose and garments can influence the predicted layout.

m^c covering the target garment's class c on the model image y, and generates a realistic synthesized image of the model wearing the provided garment. $G_{garment}$ consists of two modules: (a) a warper to create k **specialized warps**, by aligning the garment image x with the semantic layout m^c of the garment class; (b) an inpainting module to generate the final image leveraging the warps, the semantic layout m, the skin color of the model s (median color of the face), and the incomplete model image y_i where the target garment, skin, and background are masked out. Unlike prior works [18, 53, 16, 58] that learn a single warp with high degrees of freedom to align garments, our method learns a family of warps, each specializing on certain features. The inpainting network is fed all the warps and learns to combine them by choosing features to look for from each warp, as it is trained jointly with the warper.

The **Warping Module** resembles a spatial transformer network [24]. First, a regressor takes in the garment image x^c and the mask m^c as input, and regress k sets of spatial transformation parameters $\theta_1 ... \theta_k$. Then, it generates a grid for each of the transformation parameters, and samples grids from the garment image x to obtain k warps $w_1..w_k$ where $w_1 = \mathcal{W}(x, \theta_1)$. The warps are optimized to match the garment worn by the target model $m^c \otimes y$ using per pixel $\mathcal{L}_1$ loss. Inspired by [16], we impose a structure loss to encourage the garment region z (a binary mask separating garment and background as in Figure 7) of x to overlap with the garment layout of the garment mask m^c on the model after warping. The warping loss can be written as:

$$\mathcal{L}_{warp}(k) = |\mathcal{W}(x, \theta) - (m^c \otimes y)| + \beta |\mathcal{W}(z, \theta_k) - m^c| \quad (2)$$

where β controls the strength of the structure loss. This loss is sufficient to train a single warp baseline method. The choice of warper here is unimportant, and in our implementation, we use affine transformation with 6 parameters.

Cascade Loss: With multiple warps, each warp w_j is trained to address the mistakes made by previous warps w_i where $i < j$. For the k th warp, we compute the minimum loss among all the previous warps at every pixel location, written as

$$\mathcal{L}_{warp}(k) = \frac{\sum_{u=1,v=1}^{W,H} min(\mathcal{L}_{warp}(1)_{(u,v)} .. \mathcal{L}_{warp}(k)_{(u,v)})}{wh} \quad (3)$$

where u, v are pixel locations; W, H are the image width and height; and $\mathcal{L}_{warp}(k)_{(u,v)}$ is the loss of the kth warp at pixel location u, v. The cascade loss computes the average loss across all warps. An additional regularization term is added to encourage the transformation parameters of all later warps to stay close to the first warp.

$$\mathcal{L}_{casc}(k) = \frac{\sum_{i=1}^{k} \mathcal{L}_{warp}(i)}{k} + \alpha \frac{\sum_{i=2}^{k} \|\theta_k - \theta_1\|^2}{k-1} \quad (4)$$

The cascade loss enforce a hierarchy among all warps, making it more costly for an earlier warp to make a mistake than for a later warp. This prevents oscillation during the training (multiple warps competing for the same objective).

The idea is comparable with boosting – using multiple simple warpers (weak learners), each with a small degree of freedom but can handle complex geometric shape when combined. Warpers interact with each others differently compared to classifiers. Concatenating multiple warps channel-wise allows a generator to reason about the geometrics while also leveraging the parallelism of the computation (less latency). Training end-to-end allows all warps to share gradients, making it possible for warps to adjust according to each other and the image generator to guide the warpers.

The **Inpainting Module** concatenates all the warps $w_1..w_k$, the semantic layout m (or m' during inference), and the incomplete image y_i as input, and outputs the final image y' of model y wearing garment x. This is different from a standard inpainting task because the exact content to inpaint is provided through the input channels. We use a U-Net architecture to encourage copying information from the input. The network is trained to reconstruct the ground truth image using a per pixel $\mathcal{L}_1$ loss, a perceptual loss [29], and a Spectral Norm GAN with hinge loss [39]. The total loss for training $G_{garment}$ with k warps is written as

$$\mathcal{L}_{garm}(k) = \gamma_1 \mathcal{L}_{casc}(k) + \gamma_2 \mathcal{L}_1 + \gamma_3 \mathcal{L}_{perc} + \gamma_4 \mathcal{L}_{GAN} \quad (5)$$

where γ_1, γ_2, γ_3 and γ_4 are the weights for each loss.

4. Garment-Pose Matching

While our Outfit Visualization Network and other prior works [16, 58] support shape changes (e.g., skirt to pants, long sleeve to short sleeve), we notice that semantic layout generators strongly favor certain garment-model(person) pair over others. The root cause is because the pose detection results are heavily biased by garments (Figure 5). For example, the pose representation extracted from a person wearing a long dress has attributes (e.g., odd position

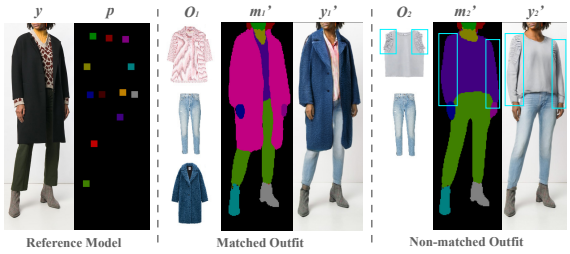

Figure 6. This figure shows an example result from a matched garment-pose pair versus a non-matched pair. A model y with extracted pose p is fed two different outfits O_1 and O_2. The garments in O_1 match with the shape of garments worn by the original model y, thus results in an accurate layout prediction m_1' and output y_1'. In contrast, the sleeveless tank in O_2 does not match with pose p, thus was wrongly generated with sleeves in y_2'.

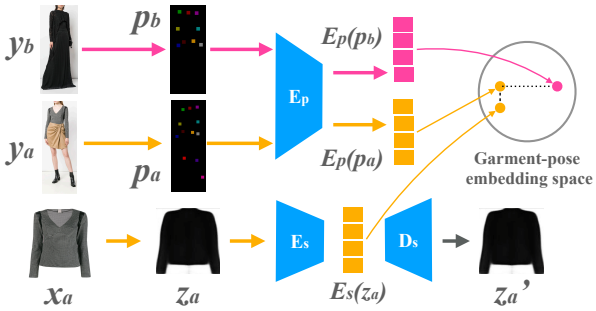

Figure 7. This figure shows the training procedure for the garment-pose matching embedding. We obtained a foreground mask z_a from garment image x_a, and learn a shape Auto-encoder $\{E_s, D_s\}$ to produce a shape embedding. The pose p_a from the corresponding model y_a is embedded closer to z_a than a random pose p_b. This only works for scenarios **(a) and (b)** (from Intro 1)

of the feet, wide legs, etc.) that hint to the generator to expect a long dress, Figure 5. Thus, during inference, putting a different garment (e.g. trousers) on this model will cause problems (Figure 6), because the garment and pose are always extracted from the same person during training. Fully addressing this problem may require improving pose representations and is left as a future direction.

To overcome such deficiency, we propose that choosing a suitable model for a given set of garments will result in better quality generation compared to using a random model. The strategy can be adopted in application scenarios **(a) and (b)** (from the Intro 1) where we are not forced to operate on a fixed model image. The general relationship between pose and garment is hard to capture, but we expect a garment x_a to work well with its paired model y_a. Also, because shape is the only relevant attribute to the semantic layout, we expect a garment x_b with similar shape as x_a to work better with y_a than a garment x_c with a different shape. We want an embedding to capture such property.

To train the garment-pose embedding, we first learn a Garment Shape Auto-encoder $\{E_s, D_s\}$ to obtain a condensed garment shape representation (Figure 7). We use the garment's foreground mask z (a binary mask of 1's for foreground and 0's for background) as input, and train the Auto-encoder to reconstruct the garment mask $z' = D_s(E_s(z))$ using mean squared error as the reconstruction loss. Additionally, we apply $\mathcal{L}_2$ normalization on the Auto-encoder's embedding space and we regard the data encoding $E_s(z)$ as an embedding for garment shape. Subsequently, we learn a pose encoder E_p to project Openpose map p into the shape embedding space. E_p is trained using a Triplet loss $\mathcal{L}_{triplet}$ [49] to encourage p_a and z_a with an identical garment a to have a closer embedding to each other compared to a randomly sampled pose p_b by a margin of α. The full training loss is written as

$$\mathcal{L}_{match} = \|D_s(E_s(z_a)) - z_a\|^2 + \mathcal{L}_{triplet}(E_s(z_a), E_p(p_a), E_p(p_b)) \quad (6)$$

Because the same pose may correspond to garments of multiple categories, we train a set of specific pose encoders $\{E_p^{c_1}..E_p^{c_n}\}$ for each garment category $c \in C$.

At inference time, we search for a set of suitable poses given a query outfit $O = \{z^{c_1}, ..., z^{c_m}\}$ (a set of garments of different categories). We compute the distance between the outfit O and a pose p as the maximum distance between the shape embedding of any garment in the outfit and the pose embedding: $d(O, p) = max(\{\|E_s(z^{c_i}) - E_p^{c_i}(p)\|^2, z^{c_i} \in O\})$. The images whose pose have the shortest distances to the query outfit are preferably chosen.

5. Experiments

5.1. Datasets & Experiment Setup

Because publicly available try-on datasets do not include rich garment categories, we experiment on a new dataset of 321k fashion products scraped from e-commerce websites, containing all the available garment categories. Each product includes a neutral garment image (front-view, laying flat, plain background), and a model image (single person, front-view). Garments are grouped into four types (top, bottoms, outerwear, or full-body). We randomly split the data into 80% for training, 5% for validation and 15% for testing. Because the model images do not come with body parsing annotation, we use off-the-shelf human parsing models [34] to generate semantic layouts as training labels.

We also compare with prior work on the established VITON dataset [18]. Note we do not compare with single-garment try-on methods on the new multi-category dataset because single-garment try-on methods do not work reasonably on our dataset, we expand on this in our supplementary. Because the original VITON test set consists of only 2,032 garment-model pairs (insignificant for computing FID), we resample a larger test set of 50k mismatched garment-model pairs, following the procedure of the original work [18]. To quantify the effect of garment-poses on generation quality,

Figure 8. The figures shows qualitative comparison between using multiple (2) warpers and a single warper. Note for single warp: the buttons are in the wrong place in A and D; problems with sleeve boundaries in E; a severe misalignment in C; a misplaced tag in B. All problems are fixed in multi-warp results.

Methods	SSIM	IS
VITON [18]	.783	2.650
CP-VTON [53]	.745	2.757
GarmentGAN [44]	-	2.774
VTNFP [63]	.803	2.784
SieveNet [25]	.766	2.820
ClothFlow [16]	.841	-
ACGPN [58]	.845	2.829
Ours (4 warps)	**.852**	**2.846**

Table 1. This table compares SSIM [65] and IS [48] (larger is better) reported on the original VITON test set. Results show that our garment generation pipeline outperforms prior works.

Methods	Random Pairs	Matched Pairs
CP-VTON [53]	15.11	13.42
ACGPN [58]	11.13	9.03
Ours (4 warps)	9.81	**7.02**

Table 2. This table compares the FID_∞ [9] score on two resampled test sets (see Sec. 5.1), one randomly sampled and the other using our pose-garment matching. Results show that choosing compatible pairs yield significantly improves to all try-on methods.

we create another resampled test set where garment-model pairs are selected using our Garment-Pose matching procedure: every garment in the original test set is paired with its 25 nearest neighbor models in the pose embedding space.

Other details about network architectures, training procedures and hyper parameters are provided in the Appendix.

5.2. Quantitative Results

Following prior works, we report SSIM [65] and IS [48] scores on the original VITON test set [18]. As shown in Table 1, our multi-warp generation pipeline outperforms prior works in both metrics. Additionally, while Frechet Inception Distance (FID) [19] is commonly used to evaluate generated image quality [4, 64, 32], Chong *et al.* [9] recently showed that FID is biased and proposed an extrapolation to an unbiased score (FID_∞). We adopt FID_∞ in our work over FID. Results from WUTON [23] are excluded because their experiments were conducted on a different dataset.

Neuberger *et al.*'s [41] is the only known prior work that

Figure 9. We compare visual results between O-VITON [41] and ours. The top rows show the garments in the oufit and the bottom row shows the generated try-on results. For fair comparison, we found garment images that most closely resemble the garments chosen in [41] in terms of style, color, and texture. Image results for O-VITON are directly taken from their paper. There are substantial difference in quality between results. The unnaturally flat torso and uneven shoulders of A-1 are not present in B-1. In A-2, the buttons on the jacket are distorted/missing, whereas B-2 represents them accurately. In A-3, the jacket and top lack realism due to missing creases, folds, and bumps compared to B-3. Properties of the arms are also kept intact in B-3. (See Appendix for more)

warp	bottoms	full-body	tops	outerwear	overall
1	1.930	4.461	2.489	2.233	1.577
2	1.472	2.207	1.215	1.349	.927
4	1.461	2.069	**1.163**	1.328	.874
8	**1.458**	**2.057**	1.165	**1.323**	**.872**

Table 3. This table reports the FID_∞ [9] score (smaller is better) of our method on the new multi-category dataset. We compare the performance between using different numbers of warps. Results shows that using more warps significantly increase performance.

supports multi-garment try-on. However, quantitative comparison is impossible as (1) their code and dataset are not released and (2) their formulation uses images of people wearing garment rather than neutral garment images. Instead, we compare with them qualitatively (Figure 9).

To evaluate our garment-pose matching procedure, we run OVNet and prior methods with released implementations [53, 58] on two resampled test sets of 50k pairs, one sampled using the garment-pose matching procedure and the other sampled randomly. We report results in Table 2. Using garment-pose matching significantly improves results for all methods, even those that are designed to accept arbitrary garment-model pairs (ours and ACGPN [58]). Additionally, our garment generation pipeline shows consistently better FID_∞ scores compared to other methods.

Table 3 reports the FID_∞ for our method on the multi-

Figure 10. Our method has forgiving failure modes. When it fails, it still outputs an image of the person wearing realistic garments, but with misrepresented attributes. In A, it turns spaghetti straps into thick straps, and has difficulty with laces; in B, the coat is generated as open-back; the asymmetrical neckline in C is turned into panels; and transparency is not captured in D.

category test dataset using different number of warps. Using more warps substantially improves the performance on all garment categories, with diminishing returns as it increases. We set the number of warps to 4.

5.3. Qualitative Comparison

We show comprehensive qualitative examples of our method. In Figure 8, we show how multiple warpers can significantly improve and correct the details. In Figure 1, we show examples of how garment details are realistically captured: patterns (A-1), shadows (B-1, C-2, D-2), hemlines (C-2), buttons (B-1) and numerous other features are all accurately represented (refer to figure for more details).

In Figure 2, we show that our method can generate the same outfit selection on a diverse set of models and poses (e.g. different stances, skin colors, and hand/arm positions). The garments' properties are consistent across all models, suggesting that the network has learned a robust garment representation. Pay attention to Pose 2 & 5 when the hands are in the pockets; the jacket/jean pocket plumps up and the sleeve interacts seamlessly with the pocket. These realistic details are likely results of using a GAN loss.

Finally in Figure 9, we compare our results to O-VITON [41], the state-of-the-art in multi-garment try-on. Compared to O-VITON, our method applies clothes more naturally onto models (A-1 vs B-1), localizes buttons more accurately (A-2 vs B-2), and generates more realistic textures and more convincing fabric properties (A-3 vs B-3).

We also show common mistakes made by our method in Figure 10. Our mistakes tend to be quite forgiving, resulting in inaccurate but realistic interpretations of the outfits. These failures are caused by inaccurate layout predictions.

To further substantiate the quality of our image generation from a provided layout, we perform a user study to verify how often users can distinguish synthesized images from real images. A user is presented with an image of the product and an image of a model wearing the product. The user is then asked if the image of the model wearing the product is real or synthesized.

Figure 11. Two synthesized images that 70% of the participants in the user study thought were real. Note, e.g., the shading, the wrinkles, even the zip and the collar.

	Participants	Acc	FP	FN
Crowds	31	0.573	**0.516**	0.284
Researchers	19	0.655	**0.615**	0.175

Table 4. The user study results show that participants have difficulty distinguishing between real and synthesized images. 51.6% and 61.5% of fake images were thought to be real by crowds and researchers, respectively. Some of the real images were marked as fake, suggesting participants were actively trying to spot flaws.

The results of our case study show that users are mostly fooled by our images; there is a very high false-positive rate (i.e. synthesized image is marked real by a user; Table 4). Figure 11 shows two examples of synthesized images that 70% of participants reported as real. These are hard outerwear examples with multiple segmented regions and complicated shading. Nevertheless, our method manages to generate high quality synthesized images that consistently fool users. See supplementary material for the complete settings and results of the user study.

6. Conclusion & Discussions

In this work, we propose a systematic method to enable outfit-level generation with realistic garment details. Several design choices are crucial. (1) We operate on neutral garment images rather than images of garments worn by models. We believe using neutral product images is more accessible for consumers and readily provided by clothing brands, making our solution easily adoptable. (2) Using warping is important toward accurately preserving geometric textures. Warping multiple garments with complicated shapes is extremely challenging, and we are the first to demonstrate success in generation of all garment categories through warping. (3) Even though, our try-on generation pipeline (as well as others) support arbitrary pairs of garment and model images, we demonstrate that it is highly advantageous to carefully choose the pair when possible.

Despite the success, our method can be improved in many aspects. Our method can handle variations in body pose and skin tone, but not body shape. Enabling body shape variations would get us one step closer to achieving the difficult goal of dressing garments directly on consumers' photos. For such a task, the main challenge lies in handling out of distribution user-uploaded photos. Additionally, enabling try-on for shoes, bags, and other accessories would make the outfit generation complete.

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 Recognition (CVPR)*, June 2018. 3

[2] Kumar Ayush, Surgan Jandial, Ayush Chopra, and Balaji Krishnamurthy. Powering virtual try-on via auxiliary human segmentation learning. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 3

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016. 3

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 7

[5] 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. 4

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 4

[7] Ming Chen, Yingjie Qin, Lizhe Qi, and Yunquan Sun. Improving fashion landmark detection by dual attention feature enhancement. In *ICCV Workshops*, 2019. 3

[8] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Synthesizing training images for boosting human 3d pose estimation. 2015. 2

[9] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. *arXiv preprint arXiv:1911.07023*, 2019. 7

[10] R. Danerek, Endri Dibra, A. Cengiz Oztireli, Remo Ziegler, and Markus H. Gross. Deepgarment : 3d garment shape estimation from a single image. *Comput. Graph. Forum*, 2017. 3

[11] 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, 3, 4

[12] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bowen Wu, Bing-Cheng Chen, and Jian Yin. Fw-gan: Flow-navigated warping gan for video virtual try-on. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 3

[13] Haoye Dong, Xiaodan Liang, Bochao Wang, Hanjiang Lai, Jia Zhu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *ICCV*, 2019. 3

[14] A. K. Grigor'ev, Artem Sevastopolsky, Alexander Vakhitov, and Victor S. Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. *CVPR*, 2019. 3

[15] Peng Guan, Loretta Reiss, David Hirshberg, Alexander Weiss, and Michael Black. Drape: Dressing any person. *ACM Transactions on Graphics - TOG*, 2012. 2

[16] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R. Scott. Clothflow: A flow-based model for clothed person generation. In *ICCV*, 2019. 1, 2, 3, 4, 5, 7

[17] Xintong Han, Zuxuan Wu, Weilin Huang, Matthew R. Scott, and Larry S. Davis. Compatible and diverse fashion image inpainting. 2019. 3

[18] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *CVPR*, 2018. 1, 2, 3, 5, 6, 7

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. 7

[20] Wei-Lin Hsiao and Kristen Grauman. Dressing for diverse body shapes. *ArXiv*, 2019. 3

[21] Wei-Lin Hsiao, Isay Katsman, Chao-Yuan Wu, Devi Parikh, and Kristen Grauman. Fashion++: Minimal edits for outfit improvement. In *ICCV*, 2019. 3, 4

[22] HyugJae, Rokkyu Lee, Minseok Kang, Myounghoon Cho, and Gunhan Park. La-viton: A network for looking-attractive virtual try-on. In *ICCV Workshops*, 2019. 3

[23] Thibaut Issenhuth, J. Mary, and Clément Calauzènes. Do not mask what you do not need to mask: a parser-free virtual try-on. *ECCV*, 2020. 1, 2, 3, 7

[24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015. 3, 5

[25] Surgan Jandial, Ayush Chopra, Kumar Ayush, Mayur Hemani, Abhijeet Kumar, and Balaji Krishnamurthy. Sievenet: A unified framework for robust image-based virtual try-on. In *WACV*, 2020. 1, 2, 3, 7

[26] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 4

[27] Moon-Hwan Jeong, Dong-Hoon Han, and Hyeong-Seok Ko. Garment capture from a photograph. *Journal of Visualization and Computer Animation*, 2015. 3

[28] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. Deep view morphing. In *CVPR*, 2017. 3

[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 5

[30] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. *CVPR*, 2018. 3

[31] Angjoo Kanazawa, David Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016. 3

[32] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 7

[33] Christoph Lassner, Gerard Pons-Moll, and Peter V. Gehler. A generative model for people in clothing. In *ICCV*, 2017. 4

[34] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *arXiv preprint arXiv:1910.09777*, 2019. 4, 6

[35] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *CVPR*, 2018. 3

[36] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018. 3

[37] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *arXiv preprint arXiv:1611.04076*, 2016. 4

[38] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In *Computer Vision and Pattern Recognition (CVPR), 2020 IEEE Conference on*, 2020. 3

[39] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 5

[40] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope : Silhouette-based clothed people – supplementary materials. In *CVPR*, 2019. 3

[41] Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. Image based virtual try-on network from unpaired data. In *CVPR*, 2020. 1, 2, 3, 4, 7, 8

[42] Natalia Neverova, Riza Alp Güler, and Iasonas Kokkinos. Dense pose transfer. In *ECCV*, 2018. 3

[43] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[44] Amir Hossein Raffiee and Michael Sollami. Garmentgan: Photo-realistic adversarial fashion transfer. *ArXiv*, 2020. 3, 7

[45] Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, and Jingwan Lu. Swapnet: Image based garment transfer. In *ECCV*, 2018. 3

[46] I. Rocco, R. Arandjelović, and J. Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017. 1, 2, 3

[47] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *ICCV*, 2019. 3

[48] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 7

[49] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 6

[50] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 4

[51] Dan Song, Tianbao Li, Zhendong Mao, and Anan Liu. Sp-viton: shape-preserving image-based virtual try-on network. *Multimedia Tools and Applications*, 2019. 3

[52] Kristen Vaccaro, Tanvi Agarwalla, Sunaya Shivakumar, and Ranjitha Kumar. Designing the future of personal fashion. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018. 1

[53] 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. 1, 2, 3, 5, 7

[54] Jiahang Wang, Wei Zhang, Wen-Hao Liu, and Tao Mei. Down to the last detail: Virtual try-on with detail carving. *ArXiv*, 2019. 3

[55] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 4

[56] Zhonghua Wu, Guosheng Lin, Qingyi Tao, and Jianfei Cai. M2e-try on net: Fashion from model to everyone. In *MM '19*, 2018. 3

[57] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017. 3

[58] 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. 1, 2, 3, 4, 5, 7

[59] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019. 3

[60] Jiahui Yu, Zhe L. Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 3

[61] Li Yu, Yueqi Zhong, and Xin Wang. Inpainting-based virtual try-on network for selective garment transfer. *IEEE Access*, 2019. 3

[62] L. Yu, Y. Zhong, and X. Wang. Inpainting-based virtual try-on network for selective garment transfer. *IEEE Access*, 2019. 3

[63] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. 3, 7

[64] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 7

[65] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 7

[66] Shizhan Zhu, Sanja Fidler, Raquel Urtasun, Dahua Lin, and Change Loy Chen. Be your own prada: Fashion synthesis with structural coherence. In *CVPR*, 2017. 3, 4