Image Based Virtual Try-on Network from Unpaired Data

Assaf Neuberger Eran Borenstein Bar Hilleli Eduard Oks Sharon Alpert
Amazon Lab126
{neuberg,eran,barh,oksed,alperts}@amazon.com

Abstract

This paper presents a new image-based virtual try-on approach (Outfit-VITON) that helps visualize how a composition of clothing items selected from various reference images form a cohesive outfit *on a person in a query image. Our algorithm has two distinctive properties. First, it is inexpensive, as it simply requires a large set of single (non-corresponding) images (both real and catalog) of people wearing various garments without explicit 3D information. The training phase requires only single images, eliminating the need for manually creating image pairs, where one image shows a person wearing a particular garment and the other shows the same catalog garment alone. Secondly, it can synthesize images of multiple garments* composed *into a single, coherent outfit; and it enables* control *of the type of garments rendered in the final outfit. Once trained, our approach can then synthesize a cohesive outfit from multiple images of clothed human models, while fitting the outfit to the body shape and pose of the query person. An online optimization step takes care of fine details such as intricate textures and logos. Quantitative and qualitative evaluations on an image dataset containing large shape and style variations demonstrate superior accuracy compared to existing state-of-the-art methods, especially when dealing with highly detailed garments.*

Figure 1: Our O-VITON algorithm is built to synthesize images that show how a person in a query image is expected to look with garments selected from multiple reference images. The proposed method generates natural looking boundaries between the garments and is able to fill-in missing garments and body parts.

1. Introduction

In the US, the share of online apparel sales as a proportion of total apparel and accessories sales is increasing at a faster pace than any other e-commerce sector. Online apparel shopping offers the convenience of shopping from the comfort of ones home, a large selection of items to choose from, and access to the very latest products. However, online shopping does not enable physical try-on, thereby limiting customer understanding of how a garment will actually look on them. This critical limitation encouraged the development of virtual fitting rooms, where images of a customer wearing selected garments are generated synthetically to help compare and choose the most desired look.

1.1. 3D methods

Conventional approaches for synthesizing realistic images of people wearing garments rely on detailed 3D models built from either depth cameras [28] or multiple 2D images [3]. 3D models enable *realistic* clothing simulation under geometric and physical constraints, as well as precise *control* of the viewing direction, lighting, pose and texture. However, they incur large costs in terms of data capture, annotation, computation and in some cases the need for specialized devices, such as 3D sensors. These large costs hinder scaling to millions of customers and garments.

1.2. Conditional image generation methods

Recently, more economical solutions suggest formulating the virtual try-on problem as a conditional image generation one. These methods generate realistic looking images of people wearing their selected garments from two input images: one showing the person and one, referred to as the *in-shop garment*, showing the garment alone. These methods can be split into two main categories, depending on the training data they use: (1) *Paired-data, single-garment* approaches that use a training set of image pairs depicting the same garment in multiple images. For example, image pairs with and without a person wearing the garment (e.g. [10, 30]), or pairs of images presenting a specific garment on the same human model in two different poses. (2) *Single-data, multiple-garment* approaches (e.g. [25]) that treat the *entire* outfit (a composition of multiple garments) in the training data as a single entity. Both types of approaches have two main limitations: First, they do not allow customers to select multiple garments (e.g. shirt, skirt, jacket and hat) and then compose them together to fit with the customer's body. Second, they are trained on data that is nearly unfeasible to collect at scale. In the case of paired-data, single-garment images, it is hard to collect several pairs for each possible garment. In the case of single-data, multiple-garment images it is hard to collect enough instances that cover all possible garment combinations.

1.3. Novelty

In this paper, we present a new image-based virtual try-on approach that: 1) Provides an inexpensive data collection and training process that includes using only single 2D training images that are much easier to collect at scale than pairs of training images or 3D data.
2) Provides an advanced virtual try-on experience by synthesizing images of multiple garments *composed* into a single, cohesive outfit (Fig.2) and enables the user to *control* the type of garments rendered in the final outfit.
3) Introduces an online optimization capability for virtual try-on that accurately synthesizes fine garment features like textures, logos and embroidery.
We evaluate the proposed method on a set of images containing large shape and style variations. Both quantitative and qualitative results indicate that our method achieves better results than previous methods.

2. Related Work

2.1. Generative Adversarial Networks

Generative adversarial networks (GANs) [7, 27] are generative models trained to synthesize realistic samples that are indistinguishable from the original training data. GANs have demonstrated promising results in image generation [24, 17] and manipulation [16]. However, the original GAN formulation lacks effective mechanisms to control the output.

Conditional GANs (cGAN) [21] try to address this issue by adding constraints on the generated examples. Constraints utilized in GANs can be in the form of class labels [1], text [36], pose [19] and attributes [29] (e.g. mouth open/closed, beard/no beard, glasses/no glasses, gender). Isola et al. [13] suggested an image-to-image translation network called pix2pix, that maps images from one domain to another (e.g. sketches to photos, segmentation to photos). Such cross-domain relations have demonstrated promising results in image generation. Wang et al.'s pix2pixHD [31] generates multiple high-definition outputs from a single segmentation map. It achieves that by adding an autoencoder that learns feature maps that constrain the GAN and enable a higher level of local control. Recently, [23] suggested using a spatially-adaptive normalization layer that encodes textures at the image-level rather than locally. In addition, composition of images has been demonstrated using GANs [18, 35], where content from a foreground image is transferred to the background image using a geometric transformation that produces an image with natural appearance. Fine-tuning a GAN during test phase has been recently demonstrated [34] for facial reenactment.

2.2. Virtual try-on

The recent advances in deep neural networks have motivated approaches that use only 2D images without any 3D information. For example, the VITON [10] method uses shape context [2] to determine how to warp a garment image to fit the geometry of a query person using a compositional stage followed by geometric warping. CP-VITON [30], uses a convolutional geometric matcher [26] to determine the geometric warping function. An extension of this work is WUTON [14], which uses an adversarial loss for more natural and detailed synthesis without the need for a composition stage. PIVTONS [4] extended [10] for pose-invariant garments and MG-VTON [5] for multi-posed virtual try-on.

All the different variations of original VITON [10] require a training set of paired images, namely each garment is captured both with and without a human model wearing it. This limits the scale at which training data can be collected since obtaining such paired images is highly laborious. Also, during testing only catalog (in-shop) images of the garments can be transferred to the person's query im-

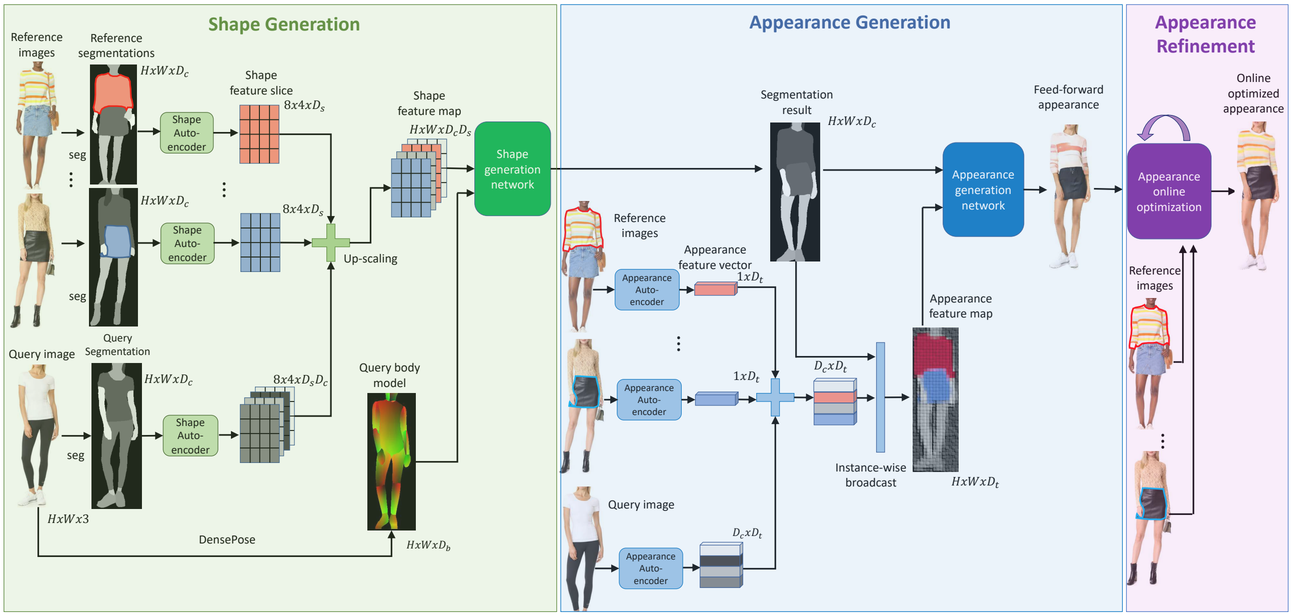

Figure 2: Our O-VITON virtual try-on pipeline combines a query image with garments selected from reference images to generate a cohesive outfit. The pipeline has three main steps. The first *shape generation* step generates a new segmentation map representing the combined shape of the human body in the query image and the shape feature map of the selected garments, using a shape autoencoder. The second *appearance generation* step feed-forwards an appearance feature map together with the segmentation result to generate an a photo-realistic outfit. An online optimization step then refines the appearance of this output to create the final outfit.

age. In [32], a GAN is used to warp the reference garment onto the query person image. No catalog garment images are required, however it still requires corresponding pairs of the same person wearing the same garment in multiple poses. The works mentioned above deal only with the transfer of top-body garments (except for [4], which applies to shoes only). Sangwoo et al [22], apply segmentation masks to allow control over generated shapes such as pants transformed to skirts. However, in this case only the shape of the translated garment is controlled. Furthermore, each shape translation task requires its own dedicated network. The recent work of [33] generates images of people wearing multiple garments. However, the generated human model is only controlled by pose rather than body shape or appearance. Additionally, the algorithm requires a training set of paired images of full outfits, which is especially difficult to obtain at scale. The work of [25] (SwapNet) swaps entire outfits between two query images using GANs. It has two main stages. Initially it generates a warped segmentation of the query person to the reference outfit and then overlays the outfit texture. This method uses self-supervision to learn shape and texture transfer and does not require a paired training set. However, it operates at the outfit-level rather than the garment-level and therefore lacks composability. The recent works of [9, 12] also generate fashion images in a two-stage process of shape and texture generation.

3. Outfit Virtual Try-on (O-VITON)

Our system uses *multiple* reference images of people wearing garments varying in shape and style. A user can select garments within these reference images to receive an algorithm-generated outfit output showing a realistic image of their personal image (query) dressed with these selected garments.

Our approach to this challenging problem is inspired by the success of the pix2pixHD approach [31] to image-to-image translation tasks. Similar to this approach, our generator G is conditioned on a semantic segmentation map and on an appearance map generated by an encoder E. The auto encoder assigns to each semantic region in the segmentation map a low-dimensional feature vector representing the region appearance. These appearance-based features enable control over the appearance of the output image and address the lack of diversity that is frequently seen with conditional GANs that do not use them.

Our virtual try-on synthesis process (Fig.2) consists of three main steps: (1) Generating a segmentation map that consistently combines the silhouettes (shape) of the selected reference garments with the segmentation map of the query image. (2) Generating a photo-realistic image showing the person in the query image dressed with the garments selected from the reference images. (3) Online optimization to refine the appearance of the final output image.

We describe our system in more detail: Sec.3.1 describes the feed-forward synthesis pipeline with its inputs, components and outputs. Sec.3.2 describes the training process of both the shape and appearance networks and Sec.3.3 describes the online optimization used to fine-tune the output image.

3.1. Feed-Forward Generation

3.1.1 System Inputs

The inputs to our system consist of a $H \times W$ RGB query image x^0 having a person wishing to try on various garments. These garments are represented by a set of M additional reference RGB images $(x^1, x^2, \dots x^M)$ containing various garments in the same resolution as the query image x^0. Please note that these images can be either natural images of people wearing different clothing or catalog images showing single clothing items. Additionally, the number of reference garments M can vary. To obtain segmentation maps for fashion images, we trained a PSP [37] semantic segmentation network S which outputs $s^m = S(x^m)$ of size $H \times W \times D_c$ with each pixel in x^m labeled as one of D_c classes using a one-hot encoding. In other words, $s(i, j, c) = 1$ means that pixel (i, j) is labeled as class c. A class can be a body part such as face / right arm or a garment type such as tops, pants, jacket or background. We use our segmentation network S to calculate a segmentation map s^0 of the query image and s^m segmentation maps for the reference images ($1 \leq m \leq M$,). Similarly, a DensePose network [8] which captures the pose and body shape of humans is applied to estimate a body model $b = B(x^0)$ of size $H \times W \times D_b$.

3.1.2 Shape Generation Network Components

The shape-generation network is responsible for the first step described above: It combines the body model b of the person in the query image x^0 with the shapes of the selected garments represented by $\{s^m\}_{m=1}^M$ (Fig. 2 green box). As mentioned in Sec.3.1.1, the semantic segmentation map s^m assigns a one hot vector representation to every pixel in x^m. A $W \times H \times 1$ slice of s^m through the depth dimension $s^m(\cdot, \cdot, c)$ therefore provides a binary mask $M_{m,c}$ representing the region of the pixels that are mapped to class c in image x^m.

A shape autoencoder E_{shape} followed by a local pooling step maps this mask to a shape feature slice $e^s_{m,c} = E_{shape}(M_{m,c})$ of $8 \times 4 \times D_s$ dimensions. Each class c of the D_c possible segmentation classes is represented by $e^s_{m,c}$, even if a garment of type c is not present in image m. Namely, it will input a zero-valued mask $M_{m,c}$ into E_{shape}.

When the user wants to dress a person from the query image with a garment of type c from a reference image m, we just replace $e^s_{0,c}$ with the corresponding shape feature slice of $e^s_{m,c}$, regardless of whether garment c was present in the query image or not. We incorporate the shape feature slices of all the garment types by concatenating them along the depth dimension, which yields a coarse shape feature map $\bar{e}^s$ of $8 \times 4 \times D_s D_c$ dimensions. We denote e^s as the up-scaled version of $\bar{e}^s$ into $H \times W \times D_s D_c$ dimensions. Essentially, combining different garment types for the query image is done just by replacing its corresponding shape features slices with those of the reference images.

The shape feature map e^s and the body model b are fed into the shape generator network G_{shape} to generate a new, transformed segmentation map s^y of the query person wearing the selected reference garments $s^y = G_{shape}(b, e^s)$.

3.1.3 Appearance Generation Network Components

The first module in our appearance generation network (Fig. 2 blue box) is inspired by [31] and takes RGB images and their corresponding segmentation maps (x^m, s^m) and applies an appearance autoencoder $E_{app}(x^m, s^m)$. The output of the appearance autoencoder is denoted as $\bar{e}^t_m$ of $H \times W \times D_t$ dimensions. By region-wise average pooling according to the mask $M_{m,c}$ we form a D_t dimensional vector $e^t_{m,c}$ that describes the appearance of this region. Finally, the appearance feature map e^t_m is obtained by a region-wise broadcast of the appearance feature vectors $e^t_{m,c}$ to their corresponding region marked by the mask $M_{m,c}$. When the user selects a garment of type c from image x_m, it simply requires replacing the appearance vector from the query image $e^t_{0,c}$ with the appearance vector of the garment image $e^t_{m,c}$ before the region-wise broadcasting which produce the appearance feature map e^t.

The appearance generator G_{app} takes the segmentation map s^y generated by the preceding shape generation stage as the input and the appearance feature map e^t as the condition and generates an output y representing the *feed-forward* virtual try-on output $y = G_{app}(s^y, e^t)$.

3.2. Train Phase

The Shape and Appearance Generation Networks are trained independently (Fig.3) using the same training set of single input images with people in various poses and clothing. In each training scheme the generator, discriminator and autoencoder are jointly-trained.

3.2.1 Appearance Train phase

We use a conditional GAN (cGAN) approach that is similar to [31] for image-to-image translation tasks. In cGAN frameworks, the training process aims to optimize a Minimax loss [7] that represents a game where a generator G and a discriminator D are competing. Given a training image x the generator receives a corresponding segmentation map $S(x)$ and an appearance feature map $e^t(x) = E_{app}(x, S(x))$ as a condition. Note that during the train phase both the segmentation and the appearance feature maps are extracted from the same input image x while during test phase the segmentation and appearance feature maps are computed from multiple images. We describe this step in Sec.3.1. The generator aims to synthe-

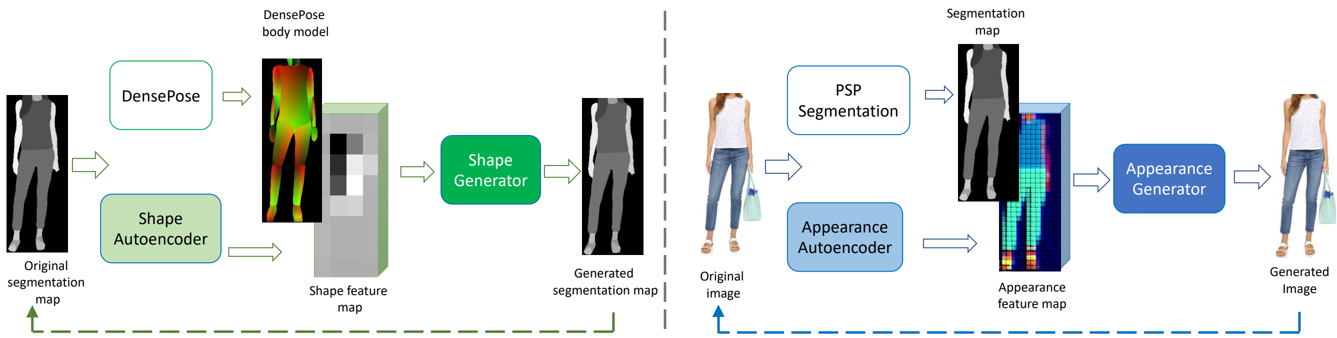

Figure 3: Train phase. (Left) Shape generator translates body model and shape feature map into the original segmentation map. (Right) Appearance generator translates segmentation map and appearance feature map into the original photo. Both training schemes use the same dataset of unpaired fashion images.

size $G_{app}(S(x), e^t(x))$ that will confuse the discriminator when it attempts to separate generated outputs from original inputs such as x. The discriminator is also conditioned by the segmentation map $S(x)$. As in [31], the generator and discriminator aim to minimize the LSGAN loss [20]. For brevity we will omit the *app* subscript from the appearance network components in the following equations.

$$\min_G \mathcal{L}_{GAN}(G) = \mathbb{E}_x[(D(S(x), G(S(x), e^t(x)) - 1)^2]$$
$$\min_D \mathcal{L}_{GAN}(D) = \mathbb{E}_x[(D(S(x), x) - 1)^2] + \mathbb{E}_x[(D(S(x), G(S(x), e^t(x)))^2] \tag{1}$$

The architecture of the generator G_{app} is similar to the one used by [15, 31], which consists of convolution layers, residual blocks and transposed convolution layers for up-sampling. The architecture of the discriminator D_{app} is a PatchGAN [13] network, which is applied to multiple image scales as described in [31]. The multi-level structure of the discriminator enables it to operate both at the coarse scale with a large receptive field for a more global view, and at a fine scale which measures subtle details. The architecture of E is a standard convolutional autoencoder network.

In addition to the adversarial loss, [31] suggested an additional feature matching loss to stabilize the training and force it to follow natural images statistics at multiple scales. In our implementation, we add a feature matching loss, suggested by [15], that directly compares between generated and real images activations, computed using a pre-trained perceptual network (VGG-19). Let ϕ_l be the vector form of the layer activation across channels with dimensions $C_l \times H_l \cdot W_l$. We use a hyper-parameter λ_l to determine the contribution of layer l to the loss. This loss is defined as:

$$\mathcal{L}_{FM}(G) = \mathbb{E}_x \sum_l \lambda_l ||\phi_l(G(S(x), e^t(x))) - \phi_l(x)||_F^2 \tag{2}$$

We combine these losses together to obtain the loss function for the Appearance Generation Network:

$$\mathcal{L}_{train}(G, D) = \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{FM}(G) \tag{3}$$

3.2.2 Shape Train Phase

The data for training the shape generation network is identical to the training data used for the appearance generation network and we use a similar conditional GAN loss for this network as well. Similar to decoupling appearance from shape, described in 3.2.1, we would like to decouple the body shape and pose from the garment's silhouette in order to transfer garments from reference images to the query image at test phase. We encourage this by applying a distinct spatial perturbation for each slice $s(\cdot, \cdot, c)$ of $s = S(x)$ using a random affine transformation. This is inspired by the self-supervision described in SwapNet [25]. In addition, we apply an average-pooling to the output of E_{shape} to map $H \times W \times D_s$ dimensions, to $8 \times 4 \times D_s$ dimensions. This is done for the test phase, which requires a shape encoding that is invariant to both pose and body shape. The loss functions for G_{shape} and discriminator D_{shape} are similar to (3) with the generator conditioned on the shape feature $e^s(x)$ rather than the appearance feature map e^t of the input image. The discriminator aims to separate $s = S(x)$ from $s^y = G_{shape}(S(x), e^s)$. The feature matching loss in (2) is replaced by a cross-entropy loss $\mathcal{L}_{CE}$ component that compares the labels of the semantic segmentation maps.

3.3. Online Optimization

The feed-forward operation of appearance network (autoencoder and generator) has two main limitations. First, less frequent garments with non-repetitive patterns are more challenging due to both their irregular pattern and reduced representation in the training set. Fig. 6 shows the frequency of various textural attributes in our training set. The most frequent pattern is solid (featureless texture). Other frequent textures such as logo, stripes and floral are extremely

Figure 4: Single garment transfer results. (Left) Query image column; Reference garment; CP-VITON [30]; O-VITON (ours) method with feed-forward only; O-VITON (ours) method with online optimization. Note that the feed-forward alone can be satisfactory in some cases, but lacking in others. The online optimization can generate more accurate visual details and better body parts completion. (Right) Generation of garments other than shirts, where CP-VITON is not applicable.

diverse and the attribute distribution has a relatively long tail of other less common non-repetitive patterns. This constitutes a challenging learning task, where the neural network aims to accurately generate patterns that are scarce in the training set. Second, no matter how big the training set is, it will never be sufficiently large to cover all possible garment pattern and shape variations. We therefore propose an online optimization method inspired by style transfer [6]. The optimization fine-tunes the appearance network during the test phase to synthesize a garment from a reference garment to the query image. Initially, we use the parameters of the feed-forward appearance network described in 3.1.3. Then, we fine-tune the generator G_{app} (for brevity denoted as G) to better reconstruct a garment from reference image x^m by minimizing the *reference* loss. Formally, given a reference garment we use its corresponding region binary mask $M_{m,c}$ which is given by s^m in order to localize the reference loss (3):

$$\mathcal{L}_{ref}(G) = \sum_l \lambda_l ||\phi_l^m(G(S(x^m), e_m^t)) - \phi_l^m(x^m)||_F^2 + (D^m(x^m, G(S(x^m), e_m^t))) - 1)^2 \quad (4)$$

Where the superscript m denotes localizing the loss by the spatial mask $M_{m,c}$. To improve generalization for the query image, we compare the newly transformed query segmentation map s^y and its corresponding generated image y using the GAN loss (1), denoted as *query* loss:

$$\mathcal{L}_{qu}(G) = (D^m(s^y, y) - 1)^2 \quad (5)$$

Our online loss therefore combines both the reference garment loss (4) and the query loss (5):

$$\mathcal{L}_{online}(G) = \mathcal{L}_{ref}(G) + \mathcal{L}_{qu}(G) \quad (6)$$

Note that the online optimization stage is applied for each reference garment separately (see also Fig. 5). Since all the regions in the query image are not spatially aligned, we discard the corresponding values of the feature matching loss (2).

4. Experiments

Our experiments are conducted on a dataset of people (both males and females) in various outfits and poses, that we scrapped from the Amazon catalog. The dataset is partitioned into a training set and a test set of 45K and 7K images respectively. All the images were resized to a fixed 512×256 pixels. We conducted experiments for synthesizing single items (tops, pants, skirts, jackets and dresses) and for synthesizing pairs of items together (i.e. top & pants).

4.1. Implementation Details

Settings: The architectures we use for the autoencoders E_{shape}, E_{app}, the generators G_{shape}, G_{app} and discriminators D_{shape}, D_{app} are similar to the corresponding components in [31] with the following differences. First, the autoencoders output have different dimensions. In our case the output dimension is $D_s = 10$ for E_{shape} and $D_t = 30$ for E_{app}. The number of classes in the segmentation map is $D_c = 20$ and $D_b = 27$ dimensions for the body model. Second, we use single level generators G_{shape}, G_{app} instead of the two level generators G_1 and G_2 because we are using a lower 512×256 resolution. We train the shape and appearance networks using ADAM optimizer for 40 and 80 epochs

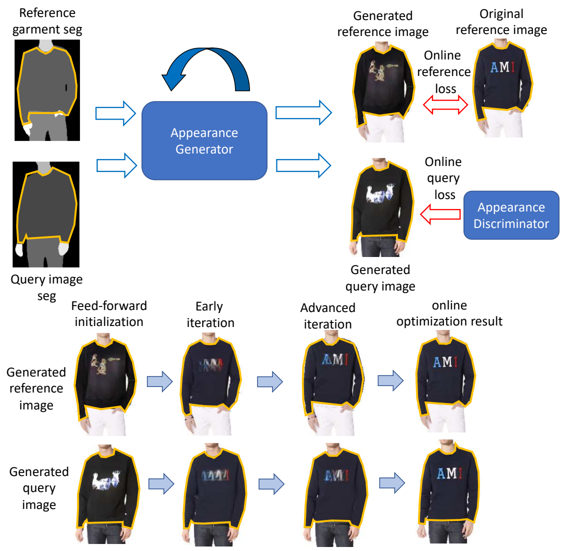

Figure 5: The online loss combines the reference and query losses to improve the visual similarity between the generated output and the selected garment, although both images are not spatially aligned. The reference loss is minimized when the generated reference garment (marked in orange contour) and the original photo of the reference garment are similar. The query loss is minimized when the appearance discriminator ranks the generated query with the reference garment as realistic.

and batch sizes of 21 and 7 respectively. Other training parameters are $lr = 0.0002, \beta_1 = 0.5, \beta_2 = 0.999$. The online loss (Sec.3.3) is also optimized with ADAM using $lr = 0.001, \beta_1 = 0.5, \beta_2 = 0.999$. The optimization is terminated when the online loss difference between two consecutive iterations is smaller than 0.5. In our experiments we found that the process is terminated, on average, after 80 iterations.

Baselines: VITON [10] and CP-VITON [30] are the state-of-the-art image-based virtual try-on methods that have implementation available online. We focus mainly on comparison with CP-VITON since it was shown (in [30]) to outperform the original VITON. Note that in addition to the differences in evaluation reported below, the CP-VITON (and VITON) methods are more limited than our proposed method because they only support generation of tops trained on a *paired* dataset.

Evaluation protocol: We adopt the same evaluation protocol from previous virtual try-on approaches (i.e. [30, 25, 10]) that use both quantitative metrics and human subjective perceptual study. The quantitative metrics include: (1) Fréchet Inception Distance (FID) [11], that measures the distance between the Inception-v3 activation distributions of the generated vs. the real images. (2) Inception score (IS) [27] that measures the output statistics of a pre-trained Inception-v3 Network (ImageNet) applied to generated images.

We also conducted a pairwise A/B test human evaluation study (as in [30]) where 250 pairs of reference and query images with their corresponding virtual try-on results (for both compared methods) were shown to a human subject (worker). Specifically, given a person's image and a target clothing image, the worker is asked to select the image that is more realistic and preserves more details of the target clothes between two virtual try-on results.

The comparison (Table 1) is divided into 3 variants: (1) synthesis of tops (2) synthesis of a single garment (e.g. tops, jackets, pants and dresses) (3) simultaneous synthesis of two garments from two different reference images (e.g. top & pants, top & jacket).

4.2. Qualitative Evaluation

Fig. 4 (left) shows qualitative examples of our O-VITON approach with and without the online optimization step compared with CP-VITON. For fair comparison we only include tops as CP-VITON was only trained to transfer shirts. Note how the online optimization is able to better preserve the fine texture details of prints, logos and other non-repetitive patterns. In addition, the CP-VITON strictly adheres to the silhouette of the original query outfit, whereas our method is less sensitive to the original outfit of the query person, generating a more natural look. Fig. 4 (right) shows synthesis results with/without the online optimization step for jackets, dresses and pants. Both methods use the same shape generation step. We can see that our approach successfully completes occluded regions like limbs or newly exposed skin of the query human model. The online optimization step enables the model to adapt to shape and garment textures that do not appear in the training dataset. Fig. 1 shows that the level of detail synthesis is retained even if the suggested approach synthesized two or three garments simultaneously.

Failure cases Fig.7 shows failure cases of our method caused by infrequent poses, garments with unique silhouettes and garments with complex non-repetitive textures, which prove to be more challenging to the online optimization step. We refer the reader to the supplementary material for more examples of failure cases.

4.3. Quantitative Evaluation

Table 1 presents a comparison of our O-VITON results with that of CP-VITON and a comparison of our results using feed-foward (FF) alone, versus FF + online optimization (online for brevity). Compared to that of CP-VITON, our online optimization FID error is decreased by approximately 17% and the IS score is improved by approximately 15%. (Note however that our FID error using feed-foward alone is higher than that of CP-VITON). The human evaluation study correlates well with both the FID and IS scores, favoring our results over CP-VITON in 65% of the tests.

		Tops	Single garment	Two garments
FID ↓	CP-VITON	20.06	-	-
	O-VITON (FF)	25.68	21.37	29.71
	O-VITON	**16.63**	**20.47**	**28.52**
IS ↑	CP-VITON	2.63±0.04	-	-
	O-VITON (FF)	2.89±0.08	3.33±0.07	3.47±0.11
	O-VITON	**3.02 ± 0.07**	**3.61 ± 0.09**	**3.51 ± 0.08**
Human ↓ O-VITON vs.	CP-VITON	65% ± 3%	-	-
	O-VITON (FF)	94%±2%	78%±3%	76%±3%

Table 1: Two quantitative and one qualitative comparisons: (1) presents the Fréchet Inception Distance (FID) [30] (2) presents the Inception Score (IS) [27] and (3) presents a A/B test human evaluation study of our O-VITON (uses online optimization) results versus the CP-VITON and our feed-forward O-VITON (FF) approach. These metrics are evaluated on three datasets: Tops only garments, single garments and two garments.

Ablation Study of the online optimization To justify the additional computational costs of the online step, we compare our method with (online) and without (FF) the online optimization step (Sec.3.3). Similarly to the comparison with CP-VITON, we use FID and IS scores as well as human evaluation. As shown in Table 1 the online optimization step showed significant improvement in the FID score for tops and comparable results on the one and two garments. We attribute the improvement on tops to the fact that tops usually have more intricate patterns (e.g. texture, logo, embroidery) than pants, jackets and skirts. Please see supplementary materials for more examples. The online optimization step also shows an advantage or comparable results for the IS score for all three scenarios. The human evaluation clearly demonstrates the advantage for the online vs feed-foward alone scheme, with 94% preference on tops, 78% preference on one garment and 76% preference on two garments.

Online loss as a measure for synthesis quality We tested the relation between the quality of the synthesized image and the minimized loss value (Eq. 6) of the online optimization scheme 3.3. We computed FID and IS scores on a subset of highly textured tops and measured a series of loss values as the optimization progresses. Starting from a high loss value of around 6.0 in fixed interval of 1.0 until a loss value of 2.0. Fig. 6 shows the behaviors of the FID error (red) with the IS score (blue). We see a clear decrease in the FID error and an increase in the IS score as the loss value decreases. We argue that the online loss value is highly correlated with the synthesis quality.

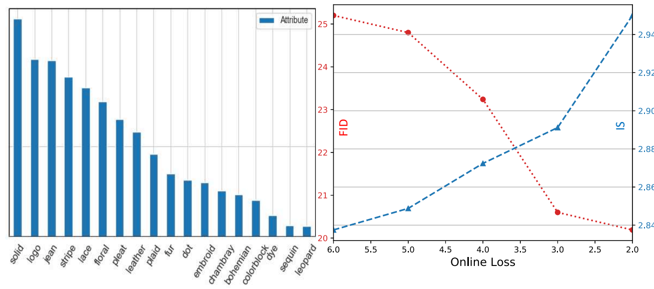

Figure 6: (Left) Textural attributes distribution. Complicated textures are less common and therefore more challenging to generate in a feed-forward operation. (Right) The online loss of the appearance generation serves as a measure for success for both the FID (dotted line) and IS (dashed line).

Figure 7: (Left) Failure cases in generating shapes. (Right) Failure cases in generating appearances.

5. Summary

We presented a novel algorithm (O-VITON) that enables an improved virtual try-on experience where the user can pick multiple garments to be composited together into a realistic-looking outfit. O-VITON works directly with individual 2D training images, which are much easier to collect and scale than pairs of training images. Our approach generates a geometrically-correct segmentation map that alters the shape of the selected reference garments to conform to the target person. The algorithm accurately synthesizes fine garment features such as textures, logos and embroidery using an online optimization scheme that iteratively fine-tunes the synthesized image. Quantitative and qualitative evaluation demonstrate better accuracy and flexibility than existing state-of-the art methods.

References

[1] C. Olah A. Odena and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 2

[2] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):509–522, 2002. 2

[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 *European Conference on Computer Vision*, pages 561–578. Springer, 2016. 1

[4] Chao-Te Chou, Cheng-Han Lee, Kaipeng Zhang, Hu-Cheng Lee, and Winston H Hsu. Pivtons: Pose invariant virtual try-on shoe with conditional image completion. In *Asian Conference on Computer Vision*, pages 654–668. Springer, 2018. 2, 3

[5] Haoye Dong, Xiaodan Liang, Bochao Wang, Hanjiang Lai, Jia Zhu, and Jian Yin. Towards multi-pose guided virtual try-on network. *arXiv preprint arXiv:1902.11026*, 2019. 2

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 6

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2, 4

[8] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 4

[9] Xintong Han, Zuxuan Wu, Weilin Huang, Matthew R Scott, and Larry S Davis. Compatible and diverse fashion image inpainting. *arXiv preprint arXiv:1902.01096*, 2019. 3

[10] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7543–7552, 2018. 2, 7

[11] 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 *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 7

[12] Wei-Lin Hsiao, Isay Katsman, Chao-Yuan Wu, Devi Parikh, and Kristen Grauman. Fashion++: Minimal edits for outfit improvement. *arXiv preprint arXiv:1904.09261*, 2019. 3

[13] 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. 2, 5

[14] Thibaut Issenhuth, Jérémie Mary, and Clément Calauzennes. End-to-end learning of geometric deformations of feature maps for virtual try-on. *arXiv preprint arXiv:1906.01347*, 2019. 2

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[16] E. Shechtman J.Y. Zhu, P. KrPähenbühl and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, 2016. 2

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[18] Chen-Hsuan Lin, Ersin Yumer, Oliver Wang, Eli Shechtman, and Simon Lucey. St-gan: Spatial transformer generative adversarial networks for image compositing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9455–9464, 2018. 2

[19] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017. 2

[20] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2017. 5

[21] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[22] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. *arXiv preprint arXiv:1812.10889*, 2018. 3

[23] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. *arXiv preprint arXiv:1903.07291*, 2019. 2

[24] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[25] Amit Raj, Patsorn Sangkloy, Huiwen Chang, Jingwan Lu, Duygu Ceylan, and James Hays. Swapnet: Garment transfer in single view images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018. 2, 3, 5, 7

[26] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6148–6157, 2017. 2

[27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 2, 7, 8

[28] Masahiro Sekine, Kaoru Sugita, Frank Perbet, Björn Stenger, and Masashi Nishiyama. Virtual fitting by single-shot body shape estimation. In *Int. Conf. on 3D Body Scanning Technologies*, pages 406–413. Citeseer, 2014. 1

[29] Wei Shen and Rujie Liu. Learning residual images for face attribute manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4030–4038, 2017. 2

[30] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 589–604, 2018. 2, 6, 7, 8

[31] 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

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 2, 3, 4, 5, 6

[32] 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

[33] Gökhan Yildirim, Nikolay Jetchev, Roland Vollgraf, and Urs Bergmann. Generating high-resolution fashion model images wearing custom outfits. *arXiv preprint arXiv:1908.08847*, 2019. 3

[34] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233*, 2019. 2

[35] Fangneng Zhan, Hongyuan Zhu, and Shijian Lu. Spatial fusion gan for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3653–3662, 2019. 2

[36] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017. 2

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 4