TailorGAN: Making User-Defined Fashion Designs

Lele Chen† Justin Tian† Guo Li
University of Rochester
lchen63@ur.rochester.edu, jtian@pas.rochester.edu, gli27@ur.rochester.edu

Cheng-Haw Wu Erh-Kan King Kuan-Ting Chen Shao-Hang Hsieh
Viscovery
{jason.wu,jeff.king,dannie.chen,shao}@viscovery.com

Chenliang Xu
University of Rochester
chenliang.xu@rochester.edu

Abstract

*Attribute editing has become an important and emerging topic of computer vision. In this paper, we consider a task: given a reference garment image A and another image B with target attribute (collar/sleeve), generate a photo-realistic image which combines the texture from reference A and the new attribute from reference B. The highly convoluted attributes and the lack of paired data are the main challenges to the task. To overcome those limitations, we propose a novel self-supervised model to synthesize garment images with disentangled attributes (*e.g.*, collar and sleeves) without paired data. Our method consists of reconstruction learning step and adversarial learning step. The model learns texture and location information through reconstruction learning. And the model capability is generalized to achieve single-attribute manipulation by adversarial learning. Meanwhile, we compose a new dataset, named GarmentSet, with annotation of landmarks of collar and sleeves on clean garment images. Thoughtful experiments on this dataset and real-world samples demonstrate that our method can synthesize significantly better results than the state-of-the-art methods in both quantitative and qualitative comparisons. The code is available at:https://github.com/gli-27/TailorGAN.*

1. Introduction

Deep generative techniques [8, 14] have led to highly successful image/video generation, some focusing on style transfer [36], and others on the synthesis of desired conditions [15, 24]. We propose a novel schema to disentangle attributes and synthesize high-quality images with the desired attribute while keeping other attributes unchanged. In this paper, we focus on the problem of fashion image attribute manipulation to demonstrate the capability of our method. By our method, users can switch a specific part of garments to the wanted designs (*e.g.*, round collar to V-collar). The objective is to synthesize a photo-realistic new fashion image by combining different parts seamlessly together. Potential applications of such systems range from item retrieval to professional fashion design assistant. In Fig. 1, we make a graphical demonstration of our task and result.

Figure 1. A graphical demonstration of our task and result. Given a reference fashion garment (on the left) and the desired design attribute (in the middle), we aim to generate a new fashion garment that seamlessly integrates the desired design attribute to the reference image. The first row shows collar-editing, and the second row shows sleeve-editing. The results generated by our system are shown on the right.

Recently, Generative Adversarial Networks (cGAN) [8, 15, 24], image-to-image translation [9], and CycleGAN [36] have been proved effective in generating photo-realistic images, image syntheses using such models usually involve highly entangled attributes and may fail in editing target attribute/object separately [18, 29, 36]. Furthermore, the large variance in the garment textures causes additional

problems. It is impossible to build a dataset that is large enough to approximate the distribution of all garment texture and design combinations, which serve as paired learning examples to train models such as [24, 9]. Novel learning paradigm is expected to overcome this difficulty.

To solve these challenges, we propose a novel self-supervised image generative model, TailorNet, that can make disentangled attribute manipulations. TailorNet exploits the latent space expression of input images without paired training images. Image edge maps are used to isolate the structures from textures. By using edge maps instead of RGB color images, we achieve good results with only a small amount of data. Our model is data-parsimonious and achieves fashion attribute-editing that is robust to various geometric transformations between the reference and design attribute images. The model capacity is further generalized in a GAN framework to achieve single-attribute manipulation using random fashion inputs. Besides the reconstruction learning and the adversarial learning, an attribute-aware discriminator is weaved into the model to guide the image editing. This attribute-aware discriminator helps in making high-quality single attribute editing and guides a better self-supervised learning process.

Currently available fashion image datasets (*e.g*., DeepFashion [13]) mostly consist of street photos with complex backgrounds or with user's body parts presented. That extra visual information may hinder the performance of an image synthesis model. Thus, to simplify the training data and screen out noisy backgrounds, we introduce a new dataset, GarmentSet. The new dataset contains fashion images with no human user presented, and most images have single-color backgrounds.

Contributions made in this paper can be summarized as:

- We propose a new task in deep learning fashion studies. Instead of generating images with text guidance, virtual try-on, or texture transferring, our task is to make new fashion designs with disentangled user-appointed attributes.
- We exploit a novel training schema consists of reconstructive learning and adversarial learning. The self-supervised reconstructive learning guides the network to learn shape, location information from the edge map, and texture information from the RGB image. The unpaired adversarial learning gives the network generalizability to synthesize images with new disentangled attributes. Besides the reconstructive learning and adversarial learning, we propose a novel attribute-aware discriminator, which helps high-quality attribute editing by isolating the design structures from the garment textures.
- A new dataset, GarmentSet, is introduced to serve our attribute editing task. Unlike existing fashion datasets in which most images illustrate the user's face or body parts with complicated backgrounds, GarmentSet filters out the most redundant information and directly serves the fashion design purpose.

The rest of this paper is organized as follows: A brief review of related work is presented in Sec. 2. The details of the proposed method are described in Sec. 3. We introduce our new dataset in Sec. 4. Experimental details and results are presented in Sec. 5. In Sec. 5.6, we further conduct ablation studies to investigate the performances of our model explicitly. And finally, Sec. 6 concludes the paper with discussions of limitations.

2. Related Work

Generative Adversarial Network (GAN) [8] is one of the most popular deep generative models and has shown impressive results in image synthesis studies, like image editing [22, 28] and fine-grained objects generating [24, 12]. Researchers utilize different conditions to generate images with desired properties. Existing works have explored various conditions, from category labels [21], audio [6, 5], text [24], skeleton[19, 10, 35, 23] to attributes [26]. There are a few studies that investigate the task of image translations using cGAN [15, 24, 4]. In the context of fashion-related applications, researchers apply cGAN in automated garment textures filling [31], texture transferring [11], and virtual try-on [37, 30] by replacing dress on a person with a new one. More related work is sequential attention GAN proposed by Cheng *et al*. [7]. Their model uses text as the guidance and continuously changes the fashion designs based on user's requests, but the attribute changes are highly entangled. In contrast to this work, we propose a new training algorithm that combining self-supervised reconstruction learning with adversarial learning to make disentangled attribute manipulations with user-appointed images.

Self-supervised generation is recently introduced as a novel and effective way to train generative models without paired training data. Unpaired image-to-image translation framework such as CycleGAN [36] removes pixel-level supervision. In CycleGAN, the unpaired image to image translation is achieved by enforcing a bi-directional translation between two domains with an adversarial penalty on the translated image in the target domain. The CycleGAN variants [34, 16] are moving towards the direction of unsupervised learning approaches. However, CycleGAN-family models also create unexpected or even unwanted results, which are shown in the experiment section. One reason for such a phenomenon is due to the lack of straightforward knowledge of the target translation domain in the circularity training process. Inherent attributes of the source samples may be changed in a translation process. To avoid such unwanted changes, we keep an image reconstruction penalty

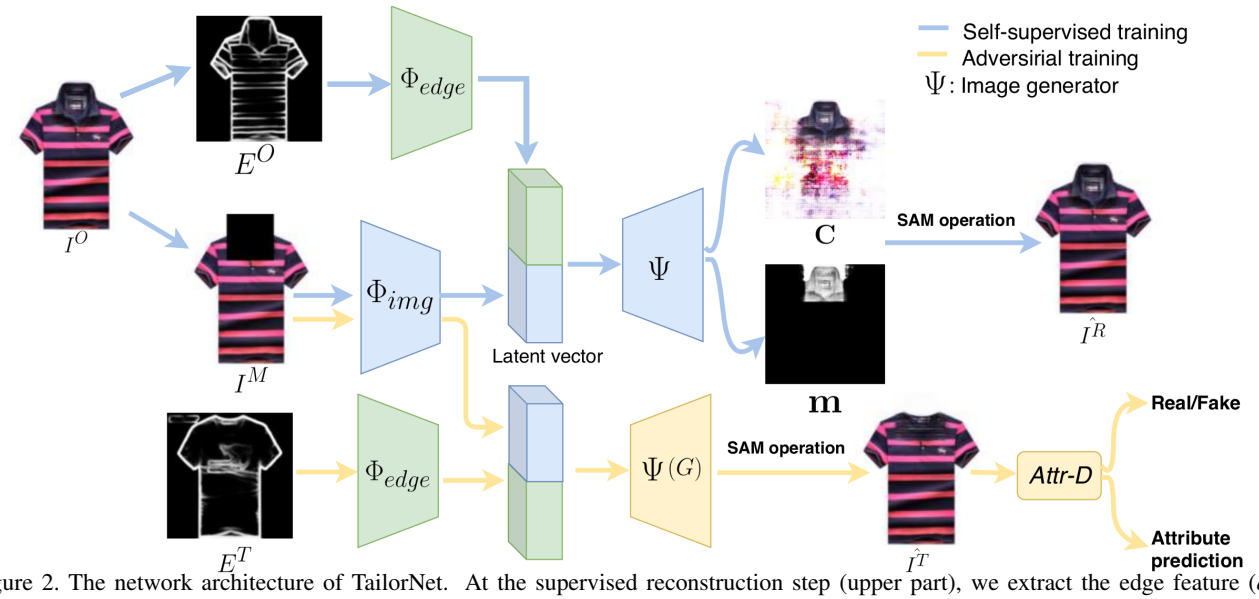

Figure 2. The network architecture of TailorNet. At the supervised reconstruction step (upper part), we extract the edge feature (*e.g.* location and shape feature) by Φ_{edge} and extract image feature (texture feature) by Φ_{img}. Then we merge the two vector in latent space and pass through image generator Ψ to output mask and attention. At the adversarial training step (the lower part), we extract the shape feature of target edge map E^T usingΦ_{edge} and extract texture feature using Φ_{img}. The attribute discriminator (Attr-D) outputs real/fake score and the attribute class of the fake image $\hat{I^T}$ yield by Ψ.

in our image editing task.

Attracted by the huge profit potentials in the fashion industries, deep learning methods have been conducted on fashion analysis and fashion image synthesis. Most existing researches focus on fashion trend prediction [2], clothing recognition with landmarks [13], clothing matching with fashion items in street photos [17] and fashion recommendation system [3, 32]. Different from those research lines, we focus on the fashion image synthesis task with single attribute manipulations.

3. Methodology

This section presents the implementation details of our method. There are two crucial steps in the model training: (1) self-supervised reconstruction learning and (2) generalized attribute manipulations using adversarial learning. The model learns how to fill the correct texture and to locate the fashion pattern in the correct position. The second step helps the model generating high-quality images with desired attributes. Although we use collar translating example in Fig. 2, we emphasize here that our model can be applied to attributes other than collar parts. We show the sleeve editing results in the later section.

3.1. TailorNet: Learning to Manipulate Designs

Self-Supervised Learning Step. The motivation of formulating a self-supervised model is the fact that it is almost impossible to collect paired training images for a fully supervised model. Using the collar editing task as an example, for each image in a fully supervised training process, one needs to collect paired images for each collar type. In these paired images, only the collar parts are different while the other attributes, like body decorations, clothing textures, etc., must stay unchanged and match with other paired images. Such data is usually unavailable. Also, the dataset size increases exponentially when multiple attribute annotations are needed for each image.

Based on the motivations discussed above, we employ an encoder-decoder structure for the self-supervised reconstruction training step. Given a masked garment image I^M (mask out the collar/sleeve part from the original garment image I^O) and edge map E^O, our reconstruction step reconstructs the original garment image (collar region). From daily experiences, individual fashion designs may highly entangle with textures or colors. For instance, light pink color is rarely used on men's garments, and leather is usually used to make jackets, etc. In our task, we want our model focusing only on the design structure editing rather than the clothing texture translating, which is inherited from the reference garment image. Specifically, the self-supervised learning step is defined as:

$$\hat{I^R} = SAM(\Psi(\Phi_{img}(I^M) \oplus \Phi_{edge}(E^O)), I^M) \ , \quad (1)$$

where Φ_{img} and Φ_{edge} are image encoder and edge encoder, respectively. Φ_{img} and Φ_{edge} consist of several 2D convolution layers and residual blocks. Ψ is the image decoder, which consists of several 2D transpose-convolution layers. $\oplus$ is channel-wise concatenation. After encoding, we feed the concatenated latent vector to Ψ to output attention mask $\mathbf{m}$ and new pixel $\mathbf{C}$. The SAM block (see Eq.3) outputs

the reconstructed image $\hat{I^R}$ based on $\mathbf{m}$, $\mathbf{C}$ and I^M. This learning step learns how to reconstruct I^O according to the texture feature from I^M and shape feature from E^O.

Different from other methods [6, 5], we only use perceptual loss, which is computed by taking the L1 distance of the VGG features (first 16 layers) for this step, which yields better results (see Fig. 10). Specifically, the loss function for this training step is defined as:

$$\begin{aligned}\mathcal{L}_R =& \mathcal{L}_{VGG}(I^O, \hat{I^R}) \\ =& \mathop{\mathbb{E}}_{\hat{I_R}}[||\phi(I^O) - \phi(\hat{I^R})||_1^1] \ ,\end{aligned} \tag{2}$$

where ϕ is a feature extractor pretrained from image classification [27]. From the reconstruction training, the generator learns to fill the garment texture and allocates the desired part at the correct position to output the original unmasked image I^O. In our empirical study, we observe that this learning step is critical to the full model since it learns how to synthesize the collar part by reconstruction. In this step, the model learns how to do texture matching and do pattern locating. During the training process, we apply random rotations, translations, and scale shifts to the input edge maps. Thus, the model can is trained to handle potential geometric transformations and can allocate the desired fashion pattern in the correct position.

Self-Attention Mask Operation (SAM). During the reconstruction step, the model should learn to change only the target part and keep the rest parts of an image untouched. Thus, we introduce a self-attention mask to the generator. The self-attention mechanism can guide the model to focus on the target region. This subsection reveals how the self-attention mask helps in making high-quality results.

After the edge map and the masked image are encoded, the latent space vectors are concatenated and then are decoded by the decoder network. The decoder produces two outputs: single-channel self-attention mask $\mathbf{m}$ and new pixel $\mathbf{C}$. The final output of the generator combines the masked color image with the input cropped image I^M. The combination step follows the equation:

$$\hat{I^R}_{i,j} = \mathbf{m}_{i,j} \times \mathbf{C}_{i,j} + (1 - \mathbf{m}_{i,j}) \times I^M_{i,j} \ , \tag{3}$$

where $\mathbf{m}_{i,j}$, $\mathbf{C}_{i,j}$ and $\hat{I^R}_{i,j}$ are the pixel at i^{th} row and j^{th} column in the self-attention mask, the new pixel and the final output image. The self-attention mask layer and the color layer share the bottom transpose convolutional blocks in the decoder. The output of the last transpose convolutional layer is fed into two activation layers: a Sigmoid layer with a single-channel output (self-attention mask) and a hyperbolic tangent layer with three-channels output (the new pixel). The self-attention mask guides the network focusing on the attribute related region while training the network in a fully self-supervised manner.

3.2. Generalized Single Attribute Manipulations

We introduced the self-supervised reconstruction learning step in Sec. 3.1, which can reconstruct the original image. However, our task is to synthesize new images by manipulating the attributes. The model trained with the reconstruction step can not yield good results since it is not generalizable to synthesize a new image with other new attribute types (*e.g.*, new collar type, and new sleeve type). Meanwhile, the synthesized image with the reconstruction step is blurry, which makes the results unrealistic. To tackle those problems, we have another adversarial learning step, which consists of the encoder-decoder network(see Sec. 3.1) and a novel attribute-aware discriminator.

To enforce the model to output image with correct attributes, we propose a discriminator with two different regression scores: a binary (real/fake) label and an attribute prediction vector. The attribute prediction vector is further optimized by cross-entropy loss, which is defined as:

$$\begin{aligned}\ell_{attr}(I), \vec{V}^{T/O}) =& - \vec{V}^{T/O} * log(f(I)) \\ & - (1 - \vec{V}^{T/O}) * log(1 - f(I)) \ ,\end{aligned} \tag{4}$$

where the vector $\vec{V}^{T/O}$ is the class vector of target attribute and f is an attribute classifier that outputs the class label vector of image I. During adversarial training, when we forward I^O and $\vec{V}^O$ to discriminator, the parameters of discriminator is updated to learn how to classify the attribute; when we forward $\hat{I^T}$ and $\vec{V^T}$ to discriminator, we do not update the parameters in discriminator, but update the parameters in generator according to the output attribute label. Thus, the full GAN loss can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{GAN}} = & \mathop{\mathbb{E}}_{\{E^T, I^M\}}[(1 - \text{D}(\text{G}(E^T, I^M))^2] + \\ & \frac{1}{2} \mathop{\mathbb{E}}_{\{E^T, I^M\}}[\text{D}(\text{G}(E^T, I^M))^2] + \\ & \frac{1}{2} \mathop{\mathbb{E}}_{\{I^O\}}[(\text{D}(I^O) - 1)^2] + \\ & \lambda_1 * \mathop{\mathbb{E}}_{\{E^T, I^M\}}[\ell_{attr}(\text{G}(E^T, I^M), \vec{V^T})] + \\ & \lambda_1 * \mathop{\mathbb{E}}_{\{I^O\}}[\ell_{attr}(I^O), \vec{V^O})] \ ,\end{aligned} \tag{5}$$

where the λ_1 is a hyper-parameter to balance the loss terms. We set $\lambda_1 = 0.1$ in all experiments. Besides the GAN loss, we also apply perceptual loss in this training step. Thus, the loss function for this adversarial learning step is defined as:

$$\mathcal{L}_{adv} = \mathcal{L}_{GAN} + \lambda_2 * \mathcal{L}_{VGG} \ , \tag{6}$$

. We set the hyper-parameter $\lambda_2 = 0.5$ to balance the loss terms. In our empirical study, we observe that the model is sensitive to λ_2 and we need to choose different λ_2 if we use different VGG layer feature to compute perceptual loss.

Figure 3. Samples of each collar type and sleeve type from GarmentSet dataset. The pie charts demonstrate the collar type and sleeve type distribution. There are in total 12 collar types and 2 sleeve types. That is the data distribution used in this paper. We will release the dataset once the paper is accepted.

Algorithm 1 Training steps

Require: α is the learning rate. B is the batch size.
Require: θ_G generator parameter. θ_D is the discriminator parameter.
for number of iterations **do**
Sample $\{I_i^O, E_i^O, I_i^M\}_{i=1}^B$
Updating the generator G in reconstruction step:
$\{\hat{I_i^R}\}_{i=1}^B \leftarrow G_\theta(\{E_i^O\}_{i=1}^B, \{I_i^M\}_{i=1}^B)$
$\theta_G \leftarrow Adam\{\nabla_{\theta_G}(\frac{1}{B}\sum_{i=1}^B(\mathcal{L}_R(I_i^O, \hat{I_i^R}), \alpha))\}$
for number of iterations **do**
Sample $\{I_i^O, E_i^T, I_i^M, \vec{V}_i^T\}_{i=1}^B$
Updating the discriminator D in adversarial step:
$\hat{I_i^T} \leftarrow G_\theta(E_i^T, I_i^M)$
$\theta_D \leftarrow Adam\{\nabla_{\theta_D}(\frac{1}{B}\sum_{i=1}^B(\mathcal{L}_{GAN}(I_i^O, \hat{I_i^T}, \vec{V}_i^T), \alpha))\}$
Updating the generator G in adversarial step:
$\hat{I_i^T} \leftarrow G_\theta(E_i^T, I_i^M)$
$\theta_G \leftarrow Adam\{\nabla_{\theta_G}(\frac{1}{B}\sum_{i=1}^B(\mathcal{L}_{adv}(I_i^O, \hat{I_i^T}, \vec{V}_i^T), \alpha))\}$

3.3. Training Algorithm

Combining the self-supervised reconstruction learning step (Sec. 3.1) with the adversarial learning step (Sec. 3.2), our full model is trained in two separately training steps. Specifically, we formulate the training algorithm in Algorithm 1. When training the generator G in reconstruction step without discriminator D, the generator G receives masked image I^M and an original attribute type E_o as input, and it outputs the reconstructed image $\hat{I^R}$. We try to minimize the reconstruction loss $\mathcal{L}^R$ to enforce the network to learn to generate correct texture and put it to geometry location. When training the discriminator D in adversarial step, the generator G receives masked image I^M and a new attribute type E^T as input, and it outputs the edited image $\hat{I^T}$. We try to minimize the GAN loss ($\mathcal{L}_{GAN}$) since there is no paired ground truth in this step. This step enforces the network to learn to synthesize images by manipulating the target attribute type E^T. Then we update parameters in generator G in the adversarial step by minimizing the adversarial loss ($\mathcal{L}_{adv}$). By optimizing the loss in reconstruction and adversarial steps, the TailorNet can learn realistic texture and geometry information to yield high-quality images with the new attribute type.

4. GarmentSet dataset

This part serves as a brief introduction to GarmentSet dataset. Currently available datasets like DeepFashion [13] and FashionGen [25] mostly consist of images including the user's face or body parts and street photos with noisy backgrounds. The redundant information raises unwanted hardness in the training process. To filter out such redundancy information in the images, we build a new dataset: GarmentSet. In this dataset, we have 9636 images with collar part annotations and 8616 images with shoulder and sleeve annotations. Both classification types and landmarks are annotated. Although in our studies, the landmark locations are only used in the image pre-processing steps, they still can be useful in future researches like fashion item retrieve, clothing recognition, etc.

In Fig. 3, we present sample pictures for each collar type, each sleeve type, and the overall data distributions in the dataset. Round collar, V-collar, and lapel images together contribute over eighty percent of the total collar-annotation images. The sleeve-dataset only contains two types: short and long sleeves. Although not used in training, the dataset also contains attribute landmark locations, including collars, shoulders, and sleeve ends. We keep collecting more data images and adding new attribute annotations. This distribution may change in the published version.

Type	Type 1 ⇒ Type 2			Type 2 ⇒ Type 1			Type 1 ⇒ Type 6			Type 6 ⇒ Type 1			Type 2 ⇒ Type 6			Type 6 ⇒ Type 2		
	C.E.	SSIM	PSNR	C.E.	SSIM	PSNR	C.E.	SSIM	PSNR	C.E.	SSIM	PSNR	C.E.	SSIM	PSNR	C.E.	SSIM	PSNR
CycleGAN	12.48	0.77	18.72	**1.74**	0.64	13.97	**5.21**	0.74	23.40	**2.97**	0.78	18.77	6.02	0.89	18.89	10.02	0.77	17.63
Pix2pix	20.01	0.87	22.75	12.73	0.89	21.42	21.62	0.88	17.63	15.79	0.89	21.88	15.12	0.77	**23.15**	19.67	0.88	**23.04**
Ours	**9.17**	**0.89**	**23.53**	3.70	**0.89**	**23.75**	6.29	**0.90**	**23.92**	3.25	**0.90**	**22.47**	**5.62**	**0.89**	23.08	**8.13**	**0.90**	22.24

Table 1. Measurements for all three models based on target translating types on testing set. In the table C. E. column is the average classification error scores for each paired type translation. The bold numbers in each column are the best scores.

Figure 4. Testing results of collar editing. The images are clustered based on the target collar types. The generated results of simple collar patterns (round and V-collar) are, in general, better than complicated ones (lapel). We train CycleGAN, pix2pix models separately for each specific transformation.

5. Experiments

5.1. Data Pre-processing

In this paper, we randomly sample 80% data for training and 20% for testing. Meanwhile, in Sec. 5.3, we keep one collar type out in the training set to demonstrate the robustness of our model. In the data pre-processing step, we generate mask-out images I^M and edge maps of data images. We use the left/right collar landmark locations to make a bounding box around the collar region. A pretrained holistically edge detection (HED) model [33] takes charge of making edge maps. The HED model is trained on the BSDS dataset [20].

We notice that the image quality of the edited results depends on the input edge map resolution. Since the HED model used is pretrained on a general image dataset, it struggles in catching structural details and may also generate unwanted noise maps. The HED model produces low-resolution edge maps for a target image with complicated, detailed structures.

5.2. Comparative Studies

For the comparative studies, we compare our model with CycleGAN [36] and pix2pix [9]. In Table 2, we compare our model with pix2pix using random collar types. Since the CycleGAN model can not handle two specific collar type translations, we drop the cycleGAN model in this random translating comparison test. In the testing results, our models out-performances pix2pix model in all collar-type translating tasks. To compare with [36], we have trained three different CycleGAN models: collar type 1 (round collar) ⇔ type 2 (V-collar), collar type 1 (round collar) ⇔ type 6 (lapel) and collar type 2 ⇔ type 6. But our model is trained with all different types together.

	C.E.	SSIM	PSNR
Our model	**5.78**	**0.8921**	**23.36**
pix2pix	16.12	0.8783	22.83

Table 2. Testing results of TailorGAN and pix2pix trained on all collar type inputs.

Qualitative results. In Fig. 4, we present testing results of three models. As one can see in the sample images, CycleGAN does not preserve garment textures. The trained pix2pix model performs poorly in all examples. At the collar part, the pix2pix outputs only show color bulks with no structural patterns.

Quantitative results. For the quantitative comparisons, in measuring model performances, we use three metrics: classification cross-entropy errors (C.E.), structure similarity index (SSIM), and peak signal to noise ratio (PSNR). The classification error is measured with a classifier pretrained on GarmentSet dataset. We use a classification error since there are no paired testing images for the edited results. The classification error can measure the distance from the target collar designs. The SSIM and the PSNR scores are derived from the differences between the original image and the edited image. From the numerical results presented in Table.1, our model outperforms both CycleGAN and pix2pix in making high-quality images with higher classification accuracy. We attribute this to the poor texture preserving the ability of the CycleGAN/pix2pix model.

5.3. Synthesizing Unseen Collar Type

To test our model's capability of processing collar types that are missing in the dataset, we take one collar type out in the training stage and test the model's performance on this unseen collar type. In this test, we take collar type 1, 2, and 6 out. We also test the taken-one-out model with a fully trained model that meets all collar types in its' training process. The classification errors for each pair of models are calculated for each taken out collar type on the testing set. The qualitative result is shown in Fig.5

Figure 5. The testing results of the unseen target collar type. The left side indicates which type we are generating (remove this type in training set). The $1^{th}, 4^{th}$ columns are the original reference images. The $2^{th}, 5^{th}$ columns are images with target collar types. The $3^{th}, 6^{th}$ columns are generated images with target collar types with texture/style of reference image.

C. E.	type 1 out	type 2 out	type 6 out
full model	**3.48**	**7.41**	**5.27**
one-out model	4.74	8.59	5.34

Table 3. The one-out model V.S. the fully trained model.

Based on the qualitative analysis and quantitative comparisons (see Tab.3), our model shows a strong generalizing ability in synthesizing unseen collar types.

5.4. Sleeve Generation

In the previous discussions, we applied our model in collar part editing. GarmentSet dataset also contains sleeve landmarks and types information. Thus, we test our model's capability in editing sleeves and present testing results in Fig. 6. We used the same training scheme. Instead of collars, the sleeve parts are masked out. Due to simpler edge structures and better resolutions in the edge maps, the edited sleeve images have better image qualities and are close to the real images. We discuss more details of sleeve editing results in the user evaluation section.

5.5. User Evaluations and Item Retrieves

To evaluate the performance in a human perceptive level, we conduct thoughtful user studies in this section. Human subjects evaluation (see Fig.7) is conducted to investigate the image quality and the attribute (collar) similarity of our generated results compared with [36, 9]. Here, we present the average scores for each model based on twenty users' evaluations. The maximum score is ten. As shown in Fig. 7,

Figure 6. Testing results of editing sleeves using TailorGAN. The $1^{th}, 4^{th}$ columns are the original reference images. The $2^{th}, 5^{th}$ columns are images with target sleeve types. The $3^{th}, 6^{th}$ columns are generated images with target sleeve types with texture/style of reference image.

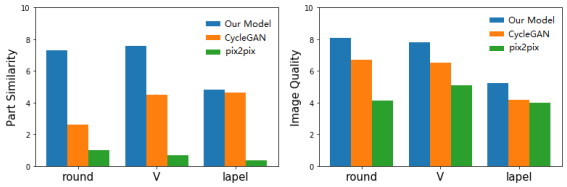

Figure 7. Average user evaluation scores based on image quality and target part similarity. The part similarity is based on users' scores on the edited part structure similarity between the edited image and the target image.

our model receives the best scores in both image quality and similarity evaluations.

We also collected users' feedback to the sleeve changing results, and the feedback shows that users can not distinguish real/fake between our generated images and real images. Sleeve-tests only evaluate image quality. In each test case, there are two pictures: (1) the original picture and (2) the edited picture. Users decide scores ranging from zero to ten to both pictures based on the image quality. Translating from short to long sleeves is, in general, a harder task due to auto-texture filling and background changing. Users may find that it is harder to distinguish the real images from the edited ones for short sleeve garments. Those observations are reflected in the evaluation scores.

To prove that TailorGAN can be useful in image item retrieves, we upload the edited images to a searching-based website[1] and show sample search results in Fig.9.

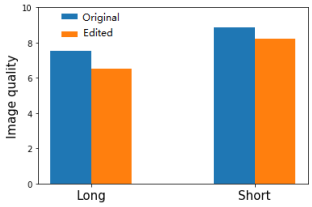

Figure 8. User feedback to sleeve editing results.

Figure 9. Top 5 matching items from [1]. The first column (green box) are generated images and the rest of the column (yellow) are retrieved images based on the generated images from [1].

Figure 10. Results of using L1 loss (w/o $\mathcal{L}_{VGG}$) and using RGB pictures (w/o E^T) as inputs instead of edge maps. We also present results of Pix2Pix in the last column.

	C.E.	SSIM	PSNR
Full model	**5.78**	**0.8921**	**23.36**
w/o reconstruction step	6.34	0.7706	19.46
w/o edge input	7.21	0.8782	21.58

Table 4. Measurements of the final model versus two variants on testing set. w/o reconstruction step represents we use only adversarial training step rather than two separately steps. w/o edge input indicates that we use RGB image as input rather than grey-scale edge map.

5.6. Ablation Studies

In this section, we want to clarify our choices of two separately training steps are crucial to generate photo-realistic results. As we argued, to disentangle the texture and the design structure meanwhile keep the rest similar, the model needs to have the capacity to reconstruct. Also, using two separately training steps makes sharper results. Tab. 4 shows that without using the reconstruction training step produces blurry images. Similarly, feeding RGB reference images as input instead of gray edge images, the model produces blurry results since the lack of clear geometric edge information. For qualitative analysis, We plot testing results of RGB inputs and results of only use adversarial training step and compare those changes with our full model.

Fig.10 shows edited image results of our current model versus L1 loss and RGB input results. L1 loss model doesn't prioritize high-frequency details and tends to average the pixel values in the editing region. On the other hand, VGG layers capture various features from the edge, color features in the starting layers to the texture, and common image structures in the high-level layers. On the other hand, using RGB images as inputs, the results show unexpected effects. The RGB-input includes extra structures that are not related to the target images or original images. We hypothesize that those unexpected structures are from the texture-design entanglement. We measure the classification error, SSIM, and PSNR for three variants. Since the major part of the image is left untouched in the result, the leading score may not be impressive in numbers. But, through the qualitative analysis, we can confirm that our current model can generate high-quality images with mode details preserved.

6. Conclusion

In this paper, we introduce a novel task to the deep learning fashion field. We investigate the problem of doing image editing to a fashion item with user defined attribute. Such methods can be useful in real world applications. We propose a novel training schema that can manipulate a single attribute to an arbitrary fashion image. To serve a better model training, we collect our own dataset. Our model outperformances the baseline models and successfully generates photo realistic images with desired attribute.

Acknowledgement. This work was partly supported by a research gift from Viscovery.

References

[1] Taobao. www.TaoBao.com. Accessed: 2019-07-30.

[2] Z. Al-Halah, R. Stiefelhagen, and K. Grauman. Fashion forward: Forecasting visual style in fashion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 388–397, 2017.

[3] H. Chen, A. C. Gallagher, and B. Girod. Describing clothing by semantic attributes. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III*, pages 609–623, 2012.

[4] L. Chen, Z. Li, R. K. Maddox, Z. Duan, and C. Xu. Lip movements generation at a glance. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, pages 538–553, 2018.

[5] L. Chen, R. K. Maddox, Z. Duan, and C. Xu. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7832–7841, 2019.

[6] L. Chen, S. Srivastava, Z. Duan, and C. Xu. Deep cross-modal audio-visual generation. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017, Mountain View, CA, USA, October 23 - 27, 2017*, pages 349–357, 2017.

[7] Y. Cheng, Z. Gan, Y. Li, J. Liu, and J. Gao. Sequential attention gan for interactive image editing via dialogue. *arXiv preprint arXiv:1812.08352*, 2018.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. 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.

[10] N. Jetchev and U. Bergmann. The conditional analogy gan: Swapping fashion articles on people images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2287–2292, 2017.

[11] S. Jiang and Y. fu. Fashion style generator. *IJCAI*, 2017.

[12] N. Kato, H. Osone, K. Oomori, C. W. Ooi, and Y. Ochiai. Gans-based clothes design: Pattern maker is all you need to design clothing. In *Proceedings of the 10th Augmented Human International Conference 2019*, page 21. ACM, 2019.

[13] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3343–3351, 2015.

[14] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] C. Lassner, G. Pons-Moll, and P. V. Gehler. A generative model of people in clothing. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 853–862, 2017.

[16] M. Li, H. Huang, L. Ma, W. Liu, T. Zhang, and Y. Jiang. Unsupervised image-to-image translation with stacked cycle-consistent adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–199, 2018.

[17] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3330–3337, 2012.

[18] Y. Lu, Y. Tai, and C. Tang. Attribute-guided face generation using conditional cyclegan. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, pages 293–308, 2018.

[19] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[21] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[22] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[23] A. Raj, P. Sangkloy, H. Chang, J. Lu, D. Ceylan, and J. Hays. Swapnet: Garment transfer in single view images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[24] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1060–1069, 2016.

[25] N. Rostamzadeh, S. Hosseini, T. Boquet, W. Stokowiec, Y. Zhang, C. Jauvin, and C. Pal. Fashion-gen: The generative fashion dataset and challenge. *arXiv preprint arXiv:1806.08317*, 2018.

[26] W. Shen and R. Liu. Learning residual images for face attribute manipulation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1225–1233, 2017.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[28] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016.

[29] Z. Wang, X. Tang, W. Luo, and S. Gao. Face aging with identity-preserved conditional generative adversarial networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 7939–7947, 2018.

[30] R. Y. X. Han, Z. Wu and L. S. Davis. Viton: an image based virtual try-on network. *CVPR*, 2018.

[31] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays. Texturegan: Controlling deep image synthesis with texture patches. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8456–8465, 2018.

[32] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2691–2699, 2015.

[33] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[34] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2018.

[35] B. Zhao, X. Wu, Z.-Q. Cheng, H. Liu, Z. Jie, and J. Feng. Multi-view image generation from a single-view. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 383–391. ACM, 2018.

[36] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2242–2251, 2017.

[37] S. Zhu, S. Fidler, R. Urtasun, D. Lin, and C. C. Loy. Be your own prada: Fashion synthesis with structural coherence. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1689–1697, 2017.