Tailor Me: An Editing Network for Fashion Attribute Shape Manipulation

Youngjoong Kwon[1] Stefano Petrangeli[2] Dahun Kim[3] Haoliang Wang[2]
Viswanathan Swaminathan[2] Henry Fuchs[1]

[1]University of North Carolina at Chapel Hill [2]Adobe Research [3]KAIST

Abstract

Fashion attribute editing aims to manipulate fashion images based on a user-specified attribute, while preserving the details of the original image as intact as possible. Recent works in this domain have mainly focused on direct manipulation of the raw RGB pixels, which only allows to perform edits involving relatively small shape changes (e.g., sleeves). The goal of our Virtual Personal Tailoring Network (VPTNet) is to extend the editing capabilities to much larger shape changes of fashion items, such as cloth length. To achieve this goal, we decouple the fashion attribute editing task into two conditional stages: shape-then-appearance editing. To this aim, we propose a shape editing network that employs a semantic parsing of the fashion image as an interface for manipulation. Compared to operating on the raw RGB image, our parsing map editing enables performing more complex shape editing operations. Second, we introduce an appearance completion network that takes the previous stage results and completes the shape difference regions to produce the final RGB image. Qualitative and quantitative experiments on the DeepFashion-Synthesis dataset confirm that VPTNet outperforms state-of-the-art methods for both small and large shape attribute editing.

1. Introduction

Fashion attribute editing aims to manipulate the appearance of a fashion image based on a user-specified attribute (e.g, sleeve, cloth length or width) and corresponding attribute value (e.g, long, short, narrow, or wide). It has a wide range of applications in fashion industries, online shopping, personalized marketing, advertising and entertainment.

Since pixel-level groundtruth for target image is *not* available, fashion attribute editing is learned in an unsupervised manner. The lack of strong supervision leads to the main challenges of this task: 1) the desired target attribute often requires editing a large area with complex shape operations and 2) the source image details and identity should be retained in the attribute-irrelevant regions. Existing works [4, 10, 17, 14, 2] perform the unsupervised editing directly on the input image, trying to manipulate both shape and appearance at the same time. Although these direct methods have already achieved high-quality results on appearance editing, their performance on the **shape** counterparts is still quite limited. For example, most prior works only deal with minimal shape changes on sleeves or collars which only take up very small portion of the entire image. Therefore, we focus on more flexible shape editing without affecting the wearer's identity. In addition, we demonstrate that the prior works are *sub-optimal* when it comes to large-shape manipulations, *e.g.*, cloth length and width (see Figure 4-(c)). Finally, since existing methods generate whole image pixels from scratch, they involve unwanted changes in attribute-irrelevant regions which can hurt preservation of the wearer's identity (see Figure 4 - 6th column). We note that although this paper's focus is on more flexible change manipulation, it can be easily extended for appearance editing by adding an existing appearance editor.

Figure 1: **Fashion Attribute Editing.** Our Virtual Personal Tailoring Network (VPTNet) is able to perform complex fashion shape attribute edits, even for challenging poses, while preserving the details and identity of the original image.

To address these challenges, we present our Virtual Personal Tailoring Network (VPTNet), where we decouple the fashion attribute editing into two stages: shape editing (parsing map attribute editing) and appearance completion (parsing-guided fashion image inpainting). Figure 1 shows the effectiveness of our two-stage - *shape-then-appearance* - decoupling strategy. VPTNet performs the desired at-

tribute edit on the attribute-relevant regions, while maintaining identity and fine-grained details of the source image (*e.g.*, face and hair), even for large shape changes (*e.g.*, cloth length change in 1st and 2nd row) and with challenging asymmetric poses (2nd row).

Given a target attribute, the first stage of VPTNet consists of a shape editing network which leverages an external parsing map estimator. High-quality parsing maps can be obtained at ***near-free cost*** by using an off-the-shelf module [24]. Here, we learn to edit shapes on the parsing map instead of the raw input image. Such explicit shape manipulation enables learning more complex and large-shape editing as it the parsing map contains less details than the original input image. However, a naive unsupervised training often leads to undesirable shape changes and distortions near edges due to lack of pixel-level supervision (see Figure 3 - (b, c)). To tackle this issue, we employ edge map information obtained from the parsing map, and propose ***edge-preserving constraints*** during training that can provide an effective guidance. The network learns to jointly modify the shape of the parsing map and the edge map, while fusing both complementary information for more accurate shape manipulation. We also introduce Target Region Localization (TRL) module to accurately localize which semantic components (*e.g.*, uppercloth and arms) and spatial parts (*e.g.*, around the upper arm region) should be edited. This leads to effective manipulation of the attribute-relevant region, even when the editing requires larger shape changes and the human subject presents challenging poses.

In the second stage, our appearance completion network directly samples the RGB pixels from the source image into the intersection region between the source and the synthesized parsing maps. The final result is obtained by only inpainting the cloth shape difference regions, guided by the synthesized parsing map. This approach minimizes the generation of raw RGB pixels and maximizes the usage of the source pixels from attribute-irrelevant regions, which results in high-quality results where the fine-grained details of the source image are well-preserved (Figure 1).

In summary, our contributions are as follows:

- We propose VPTNet, a two-stage *shape-then-appearance* framework for fashion attribute editing. It enables performing more flexible shape manipulation and, in turn, more accurate attribute editing;
- The proposed shape editing network and edge-preserving constraints exploit the complementariness of the edge and parsing maps. Also, the proposed TRL attention module accurately localizes the attribute-relevant regions.
- An appearance completion network to inpaint the attribute-relevant regions only, which allows to better retain the fine details and identity of the source image.
- To evaluate our method, we have extended the DeepFashion-Synthesis dataset [27] by adding cloth length and width attribute annotations. Extensive quantitative and qualitative results, including a user study, confirm the benefits of VPTNet, when compared to several state-of-the-art methods [10, 17, 4, 14].

2. Related work

In this section, we review prior works in the area of fashion image editing that are conditioned on two types of user inputs: attribute vector and user sketches.

Attribute vector conditioned editing. Isola *et al*. [11] present pix2pix and Zhu *et al*. [26] propose CycleGAN, which perform image-to-image translation in a supervised and unsupervised setting, respectively. However, arbitrary attribute editing is a multi-domain image-to-image problem, which cannot be fully solved by pix2pix or CycleGAN, which only support translation between two domains, *i.e.*, a new generator should be trained for every attribute value pair. StarGAN [4] addresses this issue by adopting a single generator that learns to perform multi-domain translations through a classification loss. Nonetheless, StarGAN is still limited when applied to the fashion domain, where many fine-grained details should be accurately manipulated. This happens because the downsampling of the StarGAN generator diminishes spatial resolution and fine details of the feature map.AttGAN tackles this problem by adopting skip connections in the generator, which however limits its ability to perform attribute editing for better image quality. STGAN [14] alleviates this problem by adopting the Selective Transfer Unit instead of plain skip connections. It is challenging to apply these image attribute editing works to the fashion image editing task, as fashion editing often requires more global shape changes (*e.g.*, changing cloth length). Fashion-AttGAN [17] extends AttGAN to the fashion domain, while improving the attribute manipulation ability of the generator by backpropagating the classification loss only to the decoder. AMGAN [2] leverages an attention mechanism to perform manipulations on attribute-relevant regions. All the aforementioned methods directly operate on the RGB pixels, which require the simultaneous manipulation of both shape and appearance. Our VPTNet employs a two-stage shape-then-appearance editing strategy. This allows VPTNet to effectively perform shape attribute editing while at the same time retaining the source image identity and fine-grained details.

User sketch conditioned editing. Yu *et al*. [21] propose a general image inpainting framework that inpaints incomplete images guided by user-provided sketchs. Portenier *et al*. [18] and Jo *et al*. [12] present a face editing system that takes sketch and color as input. Directly applying general inpainting or face inpainting approaches to the fash-

ion image inpainting is challenging because fashion images present many fine-grained details. Therefore, Han *et al*. [8] and Dong *et al*. [6] propose fashion image-specific inpainting frameworks.

3. Virtual Personal Tailoring Network

The overall architecture of the Virtual Personal Tailoring Network (VPTNet) is illustrated in Figure 2. Instead of manipulating the raw RGB images directly, VPTNet performs *shape-then-appearance* editing in a two-stage fashion: a shape editing network followed by an appearance completion network. The shape editing network manipulates the parsing map of the fashion image based on the target attribute. This is a crucial stage to synthesize a new parsing map, which is then used to guide the appearance completion network, whose goal is to fill in pixel-level textures/content to generate the final edited image. The two networks are trained separately and used together at inference time. To better introduce notations and the two stages operations performed our framework, we first detail the inference operations of VPTNet (Section 3.1), followed by training (Sections 3.2 and 3.3).

3.1. Inference

The inference pipeline of our VPTNet is structured as follows (Figure 2-third row). x^a, e^a and b are the inputs of the parsing network. $x^a \in^{K\times H\times W}$ is the source parsing map with n binary attributes $a = [a_1, \cdots, a_n]$. It consists of K binary masks, each corresponding to the semantic parsing of a clothed human, *i.e*., hair, face, ..., feet. $e^a \in^{H\times W}$ is the edge map calculated from the parsing map x^a, while b is the target attribute vector. G^P, the generator of the shape editing network, synthesizes the parsing map $\hat{x}^b$ edited according to the target attribute vector b, denoted as $G^P(x^a; e^a, b) = \hat{x}^b$. G^I, the generator of the appearance completion network, takes three inputs to generate the final inpainted image $\hat{I}^b$. First, the synthesized target parsing map $\hat{x}^b$. Second, the cloth shape difference mask $M_{diff} = [x^a - (x^a \odot \hat{x}^b)] + [\hat{x}^b - (x^a \odot \hat{x}^b)]$ caused by the attribute editing operation. Third, the source RGB image I^a multiplied by the inversion of the cloth shape difference mask M_{diff}. G^I inpaints the cloth shape difference regions to output the final RGB image, denoted as $G^I(\hat{x}^b, M_{diff}, I^a \odot (1 - M_{diff})) = \hat{I}^b$.

3.2. Shape Editing Network

The goal of this stage is to manipulate the source parsing map based on the given target attribute vector b (see Figure 2-first row). As no pixel-wise guidance is available, we propose to leverage edge information of the input shape to better preserve the delineation of the editing results. Our shape editing network incorporates the source parsing map x_a and its edge map e_a as inputs, and learns to manipulate both x_a and e_a into the target shape by $(x_b, e_b) = G^P((x_a; e_a), b)$.

Specifically, our shape editor G_P is an encoder-decoder network. The encoder G^P_{enc} takes as inputs the concatenation of the source parsing map and edge map and transforms them into the latent representation z. It is then concatenated with the target attribute vector b, and it is fed into the the two branches for map prediction $\hat{x}^b$ and edge map prediction $\hat{e}^b$, respectively. For training, we use the discriminator D^P, which is composed of two branches D^P_{adv} and D^P_{att}. D^P_{adv} is used to determine whether an image is fake or real; D^P_{att} predicts an attribute vector.

Edge map pose cue. The manipulation of the source parsing map should conform with the underlying human pose. While the input parsing map itself contains the pose information implicitly, we empirically found that adding the edge information to the shape editing network can have a stronger pose cue and achieve more pose-faithful synthesis results. The edge map can be easily computed by Laplacian operator on the input parsing map.

Edge-preserving parsing map editing. When manipulating the parsing map, results are often coarse and unstable due to the lack of pixel-level ground truth. Moreover, such self-supervised approach mainly relies on classification to drive learning, which ignores the shape information of the source clothed humans. This often generates unreasonable shapes and poor edge results. (see Figure 3 - $(b.c)$) To address this issue, we propose an edge-preserving constraints where the edge map is used to provide more explicit shape information, and to better guide the parsing map synthesis.

Our shape editing network improves the parsing map synthesis by exploiting edge features and edge prediction, as illustrated in Figure 2-first row. The parsing map predictor and edge map predictor of the decoder jointly learn the parsing and edge map in an end-to-end manner. This approach allows to exploit the close relationship between the two maps. Indeed, features from the parsing map predictor can provide high-level semantic information for learning the edge map. On the other hand, after obtaining the edge map, the implicit shape information in the edge map features can guide more precise parsing map synthesis results. Parsing map features contain rich high-level information, *i.e*., the pixel-wise semantic parsing information, which is beneficial to predict the edge map. To exploit this mutual information, our VPTNet employs a fusion block that integrates parsing map features and edge map features for edge map prediction. The parsing map feature is first applied a 1×1 convolution followed with a ReLU activation. Next, the output is summed with the edge map features to become *fused* edge map features. We also fuse the final edge map features with the parsing map features so that the edge map features can guide a more precise parsing map synthesis. A similar fusion block is employed to enrich the edge map

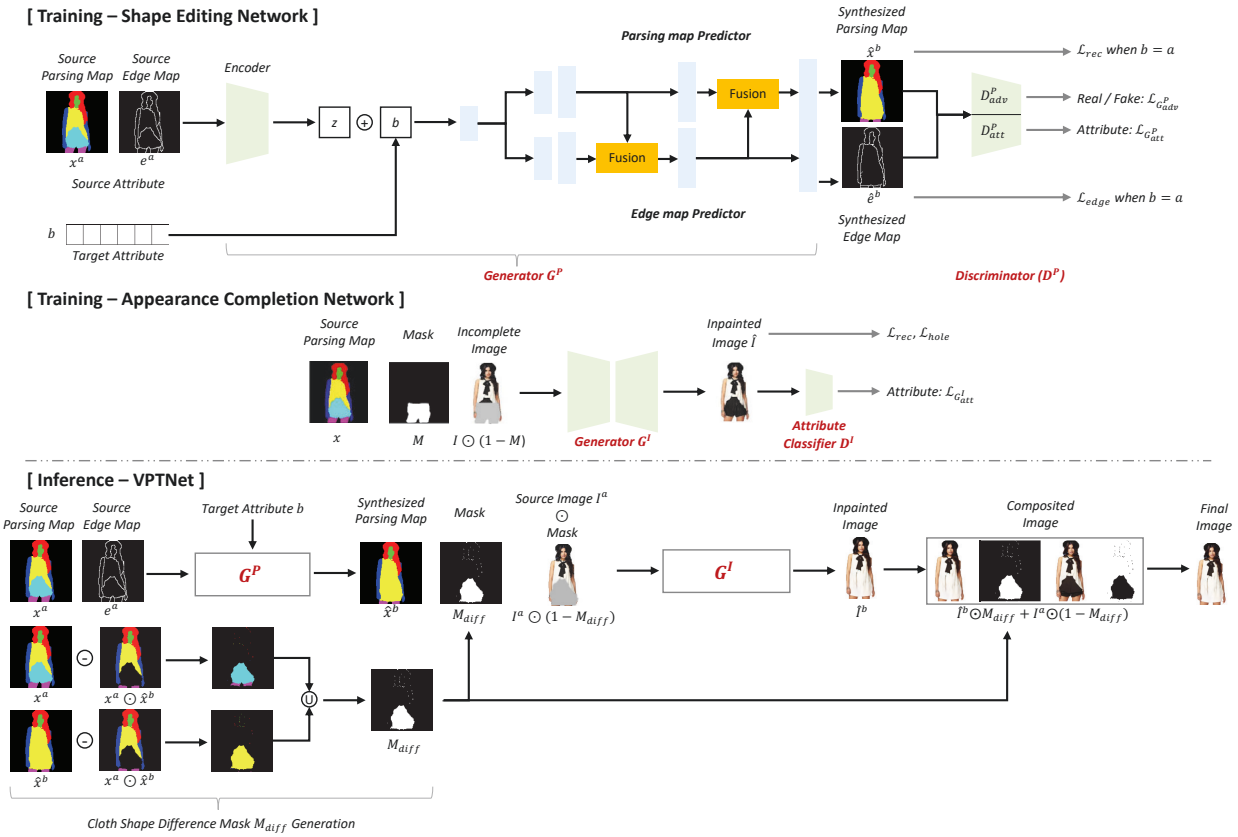

Figure 2: **Architecture**. VPTNet consists of a shape editing network and an appearance completion network, which are trained separately and used together at inference time.

with parsing map features. Each fused feature goes through the rest of its branch to become parsing map and edge map respectively. Synthesized parsing and edge maps are supervised as described in the following section.

Target Region Localization module. In order to correctly localize the parsing map channels relevant to the target manipulation as well as the spatial locations within the selected channel where the editing should be performed, we employ the Target Region Localization (TRL) module that consists of channel-wise and spatial-wise attention modules [19].

Zhu et al [2] also makes use of a localization module, where they use a pretrained attribute classifier to obtain the class activation map (CAM) [25]. This is often not optimal for shape editing task because CAM tends to fire on the dominant region in an image regardless of the target attribute, and is low-resolution and coarse. Our TRL attention is trained end-to-end with the shape editing objective, and thus it focuses on the shapes as well as better delineates the attribute-relevant regions, which will be shown later in the experiments (see Figure 3 - (a, c)).

The channel attention module focuses on localizing *which* parsing semantics (i.e., channels of the parsing map) are relevant for the target attribute change. Given the intermediate feature F, the channel attention is computed as $M_c(F) = \sigma(w(F^c_{avg})) + (w(F^c_{max}))$. F^c_{avg} and F^c_{max} denote spatial feature maps aggregated by average-pooling and max-pooling operations, respectively. w denotes a multi-layer perceptron with one hidden layer followed by ReLU activation, and σ denotes the sigmoid function. The spatial attention module focuses on *where* the attribute-relevant modifications should be performed. The spatial attention is computed as $M_s(F) = \sigma(g([F^s_{avg}; F^s_{max}]))$. F^s_{avg} and F^s_{max} denote the average-pooled features and max-pooled features across the channel axis, respectively, which are concatenated before being fed to a convolutional layer g. The final refined output is obtained by sequentially applying the channel attention and the spatial attention.

Our proposed TRL module is applied to the first layer of the encoder, as well as the first and second layers of the decoder, and effectively improves the parsing map attribute manipulation quality.

Training. Our shape editing network is trained by a reconstruction, adversarial, and attribute manipulation loss.

First, the shape editing network should be able to cor-

rectly reproduce the source map when the target attribute is the same as the source attribute. We therefore define the reconstruction loss $\mathcal{L}_{rec}$ as the L_1 pixel regression loss between the source parsing map and the synthesized parsing map when the target attribute is the same as the source attribute. We consider the edge map reconstruction as a pixel-level classification problem, following common practice in the edge detection domain [20, 23]. Most edge detection works [20, 23, 1] take advantage of the weighted cross-entropy to alleviate the class-imbalance problem in edge prediction. However, weighted binary cross-entropy leads to thick and coarse boundaries [5]. Following Deng *et al*. [5], we use the dice loss [16] and binary cross-entropy to optimize the edge map learning. The dice loss measures the overlap between predictions and ground truths, and is insensitive to the number of foreground/background pixels, thus alleviating the class-imbalance problem. Our edge reconstruction loss $\mathcal{L}_{edge}$ is formulated as $\mathcal{L}_{edge} = \mathcal{L}_{Dice}(\hat{e}^a, e^a) + \mathcal{L}_{BCE}(\hat{e}^a, e^a)$.

where $\hat{e}^a \in^{H \times W}$ denotes the predicted edge and $e^a \in^{H \times W}$ denotes the edge ground truth. where i denotes the i-th pixel and ϵ is a smooth term to avoid zero division (set to $\epsilon = 1$ in this paper).

Second, when the target attributes are different from the source ones, we do not posses the ground truth for the editing result anymore. Therefore, we employ an adversarial loss to help the network generating realistic parsing and edge maps results. Specifically, our adversarial loss $\mathcal{L}_{D^P_{adv}}$ and $\mathcal{L}_{G^P_{adv}}$ to train D^P_{adv} and G^P, respectively, are implemented following the Wasserstein GAN (WGAN) [3] and WGAN-GP [7] works. The adversarial loss is applied to the concatenation of the synthesized parsing and edge maps.

Third, we actively leverage the attribute manipulation loss to enforce that the synthesized parsing map correctly possess the desired target attribute, despite the lack of ground truth. For this reason, we introduce the attribute classifier D^P_{att}. D^P_{att} and G^P are jointly trained together through the attribute manipulation loss $\mathcal{L}_{D^P_{att}}$ and $\mathcal{L}_{G^P_{att}}$. The attribute manipulation loss is applied on the concatenation of the synthesized parsing and edge maps.

In summary, the objective to train the discriminator D^P can be formulated as $\min_{D^P} \mathcal{L}_{D^P} = -\mathcal{L}_{D^P_{adv}} + \lambda_1 \mathcal{L}_{D^P_{att}}$, and that for the generator G^P as:

$$\min_{G^P} \mathcal{L}_{G^P} = -\mathcal{L}_{G^P_{adv}} + \lambda_2 \mathcal{L}_{G^P_{att}} + \lambda_3 \mathcal{L}_{rec} + \lambda_4 \mathcal{L}_{edge}, \quad (1)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are set to 1, 10, 100, 100 respectively.

3.3. Appearance Completion Network

After the shape editing operation, the difference between the source and the shape-edited parsing map specifies the attribute-relevant regions of the source image. The goal of our appearance completion network is to inpaint the pixels in these specific regions. The architecture of the proposed appearance completion network is illustrated in Figure 2-second row. It is composed of two main components: a generator G^I and an attribute classifier D^I. For the generator, we use a simple encoder-decoder network. We replace all vanilla convolutions with gated convolutions, which have been proven effective on image inpainting tasks [21, 22]. The attribute classifier consists of five convolution layers and two fully-connected layers; given an image, its role is to predict the associated attribute vector.

The input to G^I is the concatenation of the target parsing map x, the inpainting mask M, and the incomplete RGB image $I' = I \odot (1 - M)$, where I denotes the ground truth image. G^I performs inpainting under the guidance of the input parsing map, as $G^I(x, M, I') = \hat{I}$.

Training. The goal of our appearance completion network is to inpaint the cloth shape difference regions in the final image caused by the attribute edit task, so that the inpainted regions are semantically aligned with the synthesized parsing map generated at inference time. Typical training approaches of classical inpainting works [21, 22, 13] are not directly applicable in this context for two reasons. First, they train with random masks, *e.g*., free-form, rectangle, scribbles, which are very different from the artifacts introduced by fashion attribute editing (see M_{diff} in Figure 2). Second, inpainting operations are learned to fill the mask regions with anything plausible, while our goal is to teach the network to inpaint while respecting the input semantic parsing map.

We therefore automatically generate masks on-the-fly to resembles the cloth shape difference generated at inference time, and use them during training. In addition, to produce better inpainting results, we introduce an attribute classifier D^I. D^I and G^I are jointly trained through the attribute classification loss $\mathcal{L}_{D^I_{att}}$ and $\mathcal{L}_{G^I_{att}}$. The objective to train the discriminator D^I is formulated as $\min_{D^I} \mathcal{L}_{D^I} = \mathcal{L}_{D^I_{att}}$, and that for the generator G^I is:

$$\min_{G^I} \mathcal{L}_{G^I} = \gamma_1 \mathcal{L}_{recon} + \gamma_2 \mathcal{L}_{hole} + \gamma_3 \mathcal{L}_{G^I_{att}}, \quad (2)$$

where γ_1, γ_2, and γ_3 are set to 1, 5, and 1 respectively. $\mathcal{L}_{recon}$ is calculated as the L_1 and $SSIM$ losses between the synthesized image $\hat{I}$ and the ground truth image I. $\mathcal{L}_{hole}$ is the masked loss between $M \odot \hat{I}$ and $M \odot I$.

4. Experiments

In this section, we evaluate our VPTNet both quantitatively and qualitatively. For comparison, we select AM-GAN and Fashion-AttGAN, two state-of-the-art methods in the fashion attribute editing task [2, 17], and STGAN and AttGAN, two state-of-the art methods in the face attribute editing task [14, 10]. Following Ak *et al*. [2], we evaluate our VPTNet for fashion attribute editing on the DeepFashion-Synthesis dataset [27], a refined version of

	AttGAN	Fashion-AttGAN	STGAN	AMGAN	VPTNet	VPTNet w/o A	VPTNet w/o E	VPTNet w/o F	VPTNet w/o C
Sleeve↑	76.63	79.54	78.06	81.66	**85.71**	83.35	82.29	83.76	82.83
Length↑	75.05	76.74	82.41	82.05	**85.90**	82.57	82.42	83.65	82.23
Avg↑	75.84	78.14	80.24	81.86	**85.81**	82.96	82.36	83.71	82.53

Table 1: **Evaluation of attribute editing accuracy**. We train a classifier to predict the attribute of a fashion image. Higher values indicate that the attribute editing task has been successful. Our VPTNet approach consistently outperforms the other methods.

	AttGAN	Fashion-AttGAN	STGAN	AMGAN	VPTNet	VPTNet w/o A	VPTNet w/o E	VPTNet w/o F	VPTNet w/o C
$L_1 \downarrow$	0.0477	0.0433	0.0228	0.0222	**0.0039**	0.0047	0.0101	0.00529	0.0045
PSNR↑	23.5324	24.0409	29.9748	30.5343	**32.2967**	31.4136	29.6751	30.7484	31.8406
SSIM↑	0.8591	0.8695	0.9424	0.9410	**0.9862**	0.9835	0.9624	0.9816	0.9760

Table 2: **Evaluation of image reconstruction quality.** We keep the same target attribute as the source one to evaluate the reconstruction capabilities of our method. VPTNet is able to retain the highest level of fidelity.

the DeepFashion dataset [15], consisting of 78,979 images. We perform editing on two fashion attributes: sleeve length (long, short, sleeveless) and cloth length. To enable the latter, we automatically create additional pseudo-labels in the DeepFashion-Synthesis dataset for the length of the upper-cloth, *i.e*., tops and dresses, by calculating the ratio between the uppercloth channel of the parsing map and the shorts plus legs channels (labels will be made public upon publication). The cloth length attribute consists of five values ranging from short to long. All images are resized to 128x128; we use the original train and test sets of the DeepFashion-Synthesis dataset (70,000 and 8,979 images, respectively).

4.1. Quantitative Experiments

We evaluate the performance of our attribute editing approach regarding two aspects, *i.e*., attribute editing accuracy and final image overall quality.

Attribute editing accuracy. To measure the attribute editing accuracy, we use the classification accuracy score of an attribute classifier, which allows us to evaluate if the attribute manipulation is successfully applied to the original image. Following Ak *et al*. [2], we train a ResNet-50 architecture [9] with cross-entropy loss as attribute classifier. We report the classification accuracy results in Table 1, where higher values indicate that the attribute has been successfully modified in the final image. Our VPTNet achieves the best performance against the other methods for both sleeve and cloth length attribute manipulation. We also investigate the impact of the different components of VPTNet by removing each one at the time: TRL module, edge branch, parsing-edge fusion of the shape editing network, and classification loss of the appearance completion network. Removing the TRL module from the shape editing network (VPTNet w/o A in Table 1) has a strong impact on the performance of the cloth length editing task, which indicates that the proposed TRL module can help manipulating larger shape attributes. Removing the edge branch (VPTNet w/o E) leads to the highest performance drop in the sleeve manipulation task. This shows the importance of the information provided by the edge map to perform high-quality attribute editing. Lastly, removing either the parsing-edge fusion (VPTNet w/o F) or the classification loss from the appearance completion network (VPTNet w/o C) produces slight performance decreases in both editing tasks, albeit more limited than in the previous cases. Overall, when all the improvements are enabled, our VPTNet is able to increase the classification accuracy by almost 4% when compared to the second-best method (AMGAN in Table 1).

	Attribute generation↑			Image quality↑		
	Sleeve	Length	Avg	Sleeve	Length	Avg
VPTNet (ours)	**67.9**	**71.7**	**69.8**	**58.5**	**78.1**	**68.3**
AMGAN	9.2	12.4	10.8	20.5	5.2	12.9
STGAN	3.9	5.8	4.9	14.8	5.8	10.3
F-AttGAN	14.9	7.0	11.0	4.3	7.0	5.7
AttGAN	4.1	3.1	3.6	1.9	3.9	2.9

Table 3: **User study results (63 participants)**. VPTNet outperforms all other methods in the human perspective evaluation.

Image quality. To evaluate the final image quality produced by VPTNet, we keep the same target attribute vector as the source image, and compute the L_1, PSNR, and SSIM reconstruction results (Table 2). Our VPTNet achieves the best reconstruction performance on all metrics. Particularly, VPTNet outperforms the other state-of-the-art methods by a large margin in terms of L_1 loss, since VPTNet directly re-uses the source pixels in the regions of the original image that are irrelevant for the target attribute manipulation. In terms of ablative effects, we observe that removing the edge branch leads to the most performance degradation in all metrics, which is consistent with the results in Table 1. Similar trends can be found for the other components, which confirms that each of the proposed improvements has a positive impact on the final image quality as well as on the attribute editing accuracy.

User Study. In order to confirm the objective benefits of our VPTNet, we perform a user study to evaluate the attribute editing accuracy and image quality from a human perspective, for both the sleeve and length attribute manipulation tasks. 63 people were involved in the study; each partici-

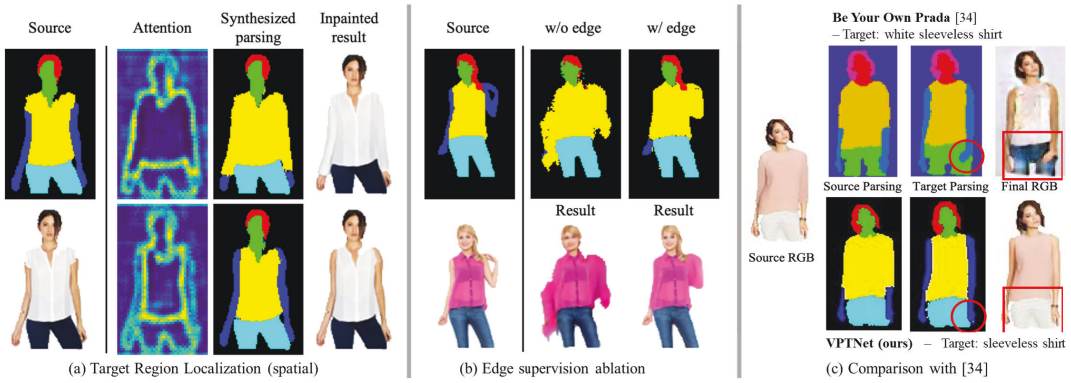

Figure 3: Visualization of the effect of the TRL attention module and edge branch.

Figure 4: Comparison results on attribute manipulation on asymmetric poses (b, c) and large shape operations (c, d).

pant was asked to answer 21 questions, each composed of 2 sub-questions (total 21 $\times$ 2 = 42). We randomly sample 21 corresponding source images that are manipulated by a target attribute of sleeve (10 images) and upperclothlength (11 images). We shuffle the results of compared methods [2, 14, 10, 17] and ours.

In each question, participants were given a pair of source images and edited results from the test set, obtained from all compared methods. First, participants were asked to identify the image presenting the *highest visual quality* and *preserving the identity and fine details of the source image*, regardless of how successful the target attribute manipulation was. Second, participants had to evaluate the image with the *most successful attribute edit manipulation*. The responses on the faithfulness and visual quality are separately summarized in Table 3. Each element in 'Sleeve' (or 'Length') column is calculated by averaging the scores over 10 (or 11) image tuples and over 63 participants. The 'Avg' column is the average of the previous two columns. Our VPTNet again achieves the best performance, both from an image quality and attribute manipulation perspective, for both sleeve and cloth length editing tasks. Particularly, these results confirm that VPTNet is superior in altering the target attribute editing without altering the source image identity and details. Moreover, our VPTNet greatly outperforms competing methods in the cloth length editing task, which confirms that our approach can produce convincing results even when the manipulation requires larger shape changes, as opposed to the other methods that often fail in this case.

4.2. Qualitative Experiments

Impact of Edge-preserving constraints. in Figure 3, We first present the effectiveness of two main components of VPTNet: the TRL module and the edge branch. As presented in Section 3, we propose the TRL module to better localize the semantic and spatial regions where to perform the attribute manipulation. Figure 3-(a, c) clearly shows that the attention mechanism leads to improved parsing map synthesis quality.

Impact of Target Region Localization (TRL). The second column in the figure shows a visualization of the TRL spatial attention when the target manipulation is to change the short sleeve into long sleeve and sleeveless, respectively. We can observe that our TRL module accurately *attend to* the target region around the arms. In Figure 3-(b, c), we present the editing results without and with the edge branch. Adding the edge branch allows VPTNet to synthesize more precise parsing maps, especially around the boundary regions.

Comparison with other methods. We also present a comparison in Figure 4 on four representative attribute ma-

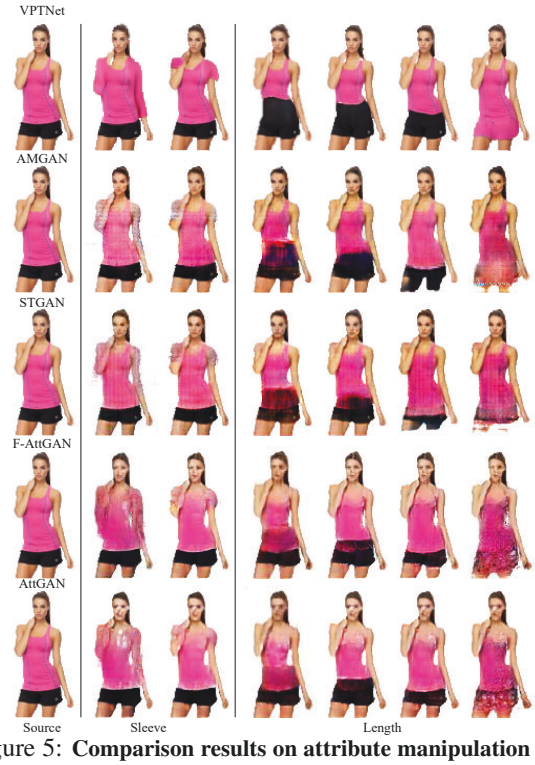

Figure 5: **Comparison results on attribute manipulation with one source.** VPTNet outperforms all the other benchmarking methods, resulting in high-quality realistic images.

nipulation tasks: long-to-short sleeve (and vice-versa) and long-to-short cloth length (and vice-versa) changes. VPTNet achieves precise attribute manipulation and shows the best results overall. We notice that even when the required shape change is small (e.g., sleeve changes as in Figure 4-(a, b)), AMGAN and STGAN generate sleeves with unclear boundaries (hands region) and inconsistencies (shoulders region). The VPTNet is able to generate realistic-looking sleeves. When the human subject presents a highly asymmetric pose (Figure 4-(b)), the other benchmarking methods fail to accurately synthesize a realistic image. Moreover, we can notice how Fashion-AttGAN and AttGAN fail retaining several fine-grained details of the source image (arms, skirt color, face details etc.). For the cloth length editing task (Figure 4-(c, d)), all the other methods fail to localize the regions to be edited and show severe artifacts. On the other hand, VPTNet is able to successfully modify the cloth length producing high-quality, realistic results. This confirms that our VPTNet can provide superior results in the shape attribute editing, even for asymmetric poses ((b) and (c)) and challenging tasks that require multiple regions to be edited, as in the cloth length manipulation ((c) and (d)).

We also evaluate on all attributes manipulation with a challenging asymmetric pose (Figure 5). In the sleeve editing task, AMGAN and STGAN only generate the silhouette or an incomplete sleeve. Also, while editing the cloth

Figure 6: **Editing of more shape attributes, and multiple attributes.** Our VPTNet can manipulate more (width of hemline and waist) and multiple attributes (sleeve and cloth length) at the same time.

length, Fashion-AttGAN and AttGAN alter the neckline region, which should remain unchanged. Moreover, all the benchmarking methods leave visible artifacts of the original cloth. This behavior is due to the difficult nature of the cloth length editing, which involves editing multiple parts at the same time: uppercloth, bottomcloth and legs. Even though AMGAN employs a similar attention mechanism as VPTNet to localize the target region to manipulate, it directly operates on the RGB pixels, which can cause suboptimal results when dealing with large edits and/or challenging poses, as in Figure 5. On the other hand, our VPTNet is able to successfully perform the target attribute manipulation in all cases, while maintaining the source image details that should remain unchanged.

Multiple attribute editing. The VPTNet can work on several other shape attributes: width of hemline and waist. We show the results in Figure 6 - (a). This confirms the applicability of VPTNet on a wide range of shape attributes. Also, the VPTNet can also successfully perform multiple attributes editing operations (sleeve and cloth length) at the same time, as shown in Figure 6 - (b).

5. Conclusion

We presented VPTNet, a two-stage framework for high-quality fashion attribute editing. First, a shape editing network modifies the source parsing map with respect to the queried attribute. Second, an appearance completion network completes the pixels on the modified regions. The VPTNet enables complex editing operations with large shape changes while retaining the identity the original wearer. Extensive quantitative and qualitative experiments confirm that our VPTNet is able to provide higher quality attribute editing results compared to several state-of-the-art methods [10, 17, 14, 2].

References

[1] David Acuna, Amlan Kar, and Sanja Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11075–11083, 2019.

[2] Kenan E Ak, Joo Hwee Lim, Jo Yew Tham, and Ashraf A Kassim. Attribute manipulation generative adversarial networks for fashion images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10541–10550, 2019.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[5] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 562–578, 2018.

[6] Haoye Dong, Xiaodan Liang, Yixuan Zhang, Xujie Zhang, Xiaohui Shen, Zhenyu Xie, Bowen Wu, and Jian Yin. Fashion editing with adversarial parsing learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8120–8128, 2020.

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[8] 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.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019.

[11] 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.

[12] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1745–1753, 2019.

[13] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019.

[14] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3673–3682, 2019.

[15] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[17] Qing Ping, Bing Wu, Wanying Ding, and Jiangbo Yuan. Fashion-attgan: Attribute-aware fashion editing with multi-objective gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[18] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *arXiv preprint arXiv:1804.08972*, 2018.

[19] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

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

[21] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[22] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[23] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5964–5973, 2017.

[24] Ziwei Zhang, Chi Su, Liang Zheng, and Xiaodong Xie. Correlating edge, pose with parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8900–8909, 2020.

[25] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[27] Shizhan Zhu, Raquel Urtasun, Sanja Fidler, Dahua Lin, and Chen Change Loy. Be your own prada: Fashion synthesis with structural coherence. In *Proceedings of the IEEE inter-*

national conference on computer vision, pages 1680–1688, 2017.