

# FashionTex: Controllable Virtual Try-on with Text and Texture

Anran Lin  
anranlin@link.cuhk.edu.cn  
SSE, CUHKSZ  
China

Nanxuan Zhao  
nanxuanzhao@gmail.com  
Adobe Research  
USA

Shuliang Ning  
shuliangning@link.cuhk.edu.cn  
FNii, CUHKSZ  
SSE, CUHKSZ  
China

Yuda Qiu  
yudaqiu@link.cuhk.edu.cn  
FNii, CUHKSZ  
SSE, CUHKSZ  
China

Baoyuan Wang  
zjuwby@gmail.com  
Xiaobing.AI  
China

Xiaoguang Han\*  
hanxiaoguang@cuhk.edu.cn  
SSE, CUHKSZ  
FNii, CUHKSZ  
China



**Figure 1:** FashionTex performs the full-body virtual try-on with multi-modal controls over garment type and texture pattern, allowing anyone to design personalized clothes with simple interactions. We show two design cases here. For each of the cases, the input portrait is presented on the left, with three outputs under different conditions. Each condition contains one text prompt and two texture patches for upper and lower cloth parts.

## ABSTRACT

Virtual try-on attracts increasing research attention as a promising way for enhancing the user experience for online cloth shopping. Though existing methods can generate impressive results, users need to provide a well-designed reference image containing the target fashion clothes that often do not exist. To support user-friendly fashion customization in full-body portraits, we propose a multi-modal interactive setting by combining the advantages of both text and texture for multi-level fashion manipulation. With the carefully designed fashion editing module and loss functions, FashionTex framework can semantically control cloth types and local texture patterns without annotated pairwise training data. We

\*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
SIGGRAPH '23 Conference Proceedings, August 6–10, 2023, Los Angeles, CA, USA  
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0159-7/23/08...\$15.00  
<https://doi.org/10.1145/3588432.3591568>

further introduce an ID recovery module to maintain the identity of input portrait. Extensive experiments have demonstrated the effectiveness of our proposed pipeline. Code for this paper are at <https://github.com/picksh/FashionTex>.

## CCS CONCEPTS

• **Computing methodologies** → **Computer vision; Image manipulation.**

## KEYWORDS

image manipulation, controllable fashion generation, multi-modal learning

## ACM Reference Format:

Anran Lin, Nanxuan Zhao, Shuliang Ning, Yuda Qiu, Baoyuan Wang, and Xiaoguang Han. 2023. FashionTex: Controllable Virtual Try-on with Text and Texture. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings (SIGGRAPH '23 Conference Proceedings)*, August 6–10, 2023, Los Angeles, CA, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3588432.3591568>

## 1 INTRODUCTION

Since the advent of e-commerce, the popularity of online shopping grows sharply. As of 2021, Amazon has over 200 million Prime subscribers [ama 2022]. To enhance user experience for online cloth shopping, virtual try-on has emerged to allow customers to try out the products before buying. It attracts increasing research attention and numerous methods [Lee et al. 2022; Lewis et al. 2021; Neuberger et al. 2020; Raj et al. 2018; Xie et al. 2021a] have been proposed. These works transfer the fashion (*i.e.*, clothes) from a reference image with an existing product to the target one. Though they can generate impressive results, users need to provide reference images containing target fashion clothes that often do not exist.

We thus seek an interactive way of supporting user-friendly fashion customization. Inspired by the recent success of text-based image manipulation [Couairon et al. 2022; Kim et al. 2022; Kwon and Ye 2022; Patashnik et al. 2021; Wei et al. 2022; Xu et al. 2022] powered by visual-textual pretrained models (*e.g.*, CLIP [Radford et al. 2021]), we also aim to conduct text-based virtual try-on, which is natural to everyone. In practical usage, we found that text is efficient for controlling high-level semantic changes (*i.e.*, cloth types), but fails to alter local details (*i.e.*, textures). To solve this problem, we introduce a new interactive setting by combining the advantages of both text and texture for virtual try-on. More specifically, given a full-body portrait, users can edit cloth types (*e.g.*, long sleeves and short pants) through texts and local patterns on clothes through texture patches, as shown in Fig. 1. Besides, we build our model on StyleGAN [Karras et al. 2019], following the recent common practice [Sarkar et al. 2021; Wei et al. 2022].

However, our task has three unique challenges. First, fashions with full-body portraits display diverse poses, cloth types, and appearances. While text takes charge of global structure deformation, texture patches target on changing local appearance. How to learn a model to precisely edit specified regions in different levels without modifying characteristics of the original body (*e.g.*, face, skin, and pose) remains difficult. Second, collecting a large dataset of pairwise data (*i.e.*, original portrait with text instruction, and modified portrait) is impractical. How to enable a model for understanding textual input is unknown. Third, when employing StyleGAN on real images, the reconstruction error often occurs, which is unacceptable for our task.

To this end, we propose a novel pipeline called *FashionTex*, controlling virtual try-on with **Text** and **Texture**. We first explore the latent codes of StyleGAN by grouping them into different levels, disentangling textures from structures for better control. A fashion editing module is then designed to learn two different mappers for textual and textural inputs based on disentangled latent codes respectively. To deform the cloth types based on the input text without paired training data, we utilize the CLIP embedding by proposing a new type loss. Our type loss can accurately modify the cloth regions (*e.g.*, only change the sleeves from long to short) without influencing the surrounding parts. As our model relies on a perfect inversion, the errors that happened during the reconstruction of real images lead to the loss of portrait identity. To deal with this problem, we present an ID recovery module for restoring the input identity to obtain acceptable results.

Through extensive experiments, we have demonstrated the effectiveness of our proposed pipeline. Our main contributions are:

- To the best of our knowledge, we are the first attempt to conduct the full-body virtual try-on with multi-modal controls (*i.e.*, text and texture patches), allowing anyone to design personalized clothes with simple interactions.
- We propose a fashion editing module for better disentangling the editings between textual and textural inputs, and a novel CLIP-based type loss for accurately adjusting the cloth types without paired training data.
- We introduce an ID recovery module to mitigate the reconstruction errors caused by StyleGAN inversion, and obtain satisfied results on real images.

## 2 RELATED WORK

*Portrait Image Generation.* The Generative Adversarial Networks (GAN) [Goodfellow et al. 2020] has been widely leveraged for image generation in recent years [Isola et al. 2017; Karras et al. 2017, 2020; Zhu et al. 2017a]. Its follow-up work StyleGAN [Karras et al. 2019] can further generate high-resolution photo-realistic images while maintaining a disentangled embedding, inspiring recent works on human-centric image synthesis works [Albahar et al. 2021; Frühstück et al. 2022; Fu et al. 2022; Sarkar et al. 2021].

InsetGAN [Frühstück et al. 2022] combines different parts from multiple pretrained GANs to obtain a photo-realistic full-body image. StyleGAN-Human [Fu et al. 2022] offers editing benchmark on clothed full-body images with prior conditions. It adopts facial editing methods [Shen et al. 2020; Shen and Zhou 2021; Wu et al. 2021] that use preset StyleGAN latent space direction to change the appearance. Although these methods [Frühstück et al. 2022; Fu et al. 2022] are well-designed for photo-realistic human synthesis, they can only generate random images and fail to accomplish image synthesis according to specific conditions, which is a crucial requirement for the virtual try-on scenario.

*Image-based Fashion Editing.* Given a portrait image and user instructions, such as a reference fashion image [Han et al. 2018; Raj et al. 2018; Wang et al. 2018; Xie et al. 2021b; Yu et al. 2019] or a texture style [Albahar and Huang 2019; Brown et al. 2022; Issenhuth et al. 2021; Xian et al. 2018], this kind of methods aims to synthesize the target image sharing the same identity as the input portrait while wearing the specific fashion. Most existing works rely on a well-designed fashion product image. Some works [Sarkar et al. 2020; Xie et al. 2021a] are designed to exchange garments between two images with different portraits, without the requirement of a product reference image. Nevertheless, this process still needs a user-satisfied fashion garment, which may be hard to create or provide. Another kind of work [Günel et al. 2018; Zhu et al. 2017b] is proposed to manipulate fashion images based on the text description. More specifically, a few existing works try to conduct fashion editing based on a user-specified attribute [Ak et al. 2019b; Chen et al. 2020; Zhu et al. 2016], such as *sleeve length, color and pattern*. However, only relying on text descriptions may be hard to control local details. Instead, our work proposes to use a multi-modal control by complementing text inputs with texture patches.

*Clip-based Image Editing.* Recently, Contrastive Language-Image Pre-training (CLIP) [Radford et al. 2021] has shown great power in

multimodal learning. Benefit from the vision-language semantic alignment, the combination of the conditional generative model and CLIP brings massive amazing results[Ramesh et al. 2022; Rombach et al. 2022]. DiffusionCLIP[Kim et al. 2022] uses diffusion model[Zhu et al. 2016] with CLIP, achieving high-quality zero-shot image manipulation results. FlexIT[Couairon et al. 2022] embeds the image and text with CLIP encoders to find the target point and edits in the latent space of the VQ-GAN autoencoder [Esser et al. 2021; Yu et al. 2021]. StyleCLIP[Patashnik et al. 2021] combines the powerful image synthesis ability of StyleGAN and the amazing image-text representation ability of CLIP, showing high-quality results of human face editing. HairCLIP[Wei et al. 2022] then introduces the idea of hair editing with a CLIP-based unified architecture for text and image reference. However, the above methods cannot directly apply to full-body fashion images because of lacking control over complex poses, types, and cloth texture. In this work, we propose a new CLIP-based type loss that can better capture the difference between the original image and the target one to perform global type editing and subtle attribute changes.

### 3 METHODOLOGY

Given a full-body portrait  $I_i$ , FashionTex aims to edit the fashion clothes for trying on the original body, by using text prompts  $t$  to indicate changes in cloth types and reference RGB patches,  $P = \{p_{up}, p_{low}\}$ , for the textural pattern of upper and lower clothes. Inspired by previous works[Frühstück et al. 2022; Wei et al. 2022], we take advantage of the generation ability of a pretrained StyleGAN on human bodies[Fu et al. 2022]. As shown in Fig. 2(a), FashionTex first inverts the input portrait image  $I_i$  back into the latent  $W+$  space of StyleGAN using the e4e encoder [Tov et al. 2021]. By manipulating this latent vector  $w$ , we can obtain the edited fashion design  $I_e$  from the pretrained StyleGAN  $G_H$ , with the new latent vector  $w'$  based on the input text  $t$  and texture patches  $P$ . We design a fashion editing module for predicting an offset  $\Delta w$ , and compute the edited latent code as  $w' = w + \Delta w$  (Sec. 3.1). The final try-on image  $I_o$  is derived by fusing the input portrait  $I_i$  with the edited fashion design  $I_e$  (Sec. 3.3). Here we explain our method in more detail.

#### 3.1 Fashion Clothes Manipulation

*Editing with StyleGAN latent code.* Our model takes multi-modal interactions as conditions, controlling different levels of fashion clothes. The text takes charge of high-level semantic structures, such as sleeve length and neckline shape, while texture tends to modify low-level local patterns. To incorporate these differences in a single model, we seek the help of well pre-trained StyleGAN[Fu et al. 2022] given its wide usage in high-quality and realistic synthesis tasks. Different layers in StyleGAN controls different levels of detail in the generated image, disentangling the appearance from structure. Following this idea, we first project the input image  $I_i$  into a StyleGAN latent code  $w$ , and divide the latent code into coarse, medium, and fine groups as  $w = [w_c, w_m, w_f]$ . The objective is to use  $w_m$  control the garment type mainly with shapes, and  $w_f$  to edit the texture and color details, relating more to fine-grain features. Following the common practice[Fu et al. 2022; Wei et al. 2022], we examine the interpretability of the corresponding latent

space by style mixing. Given a source and reference images pair, we copy each layer from the latent code of the reference to those of the source and evaluate the changes in the generated image, comparing with the source portrait. In particular, we group our latent space as 1~4 for  $w_c$ , 5~8 for  $w_m$ , 9~18 for  $w_f$ .

*Fashion editing module.* We then design a fashion editing module for predicting the updated latent code  $w'$  conditioned on text prompt  $t$  and texture patch  $P$ , as shown in Fig. 2(b). Instead of directly predicting the final code  $w'$ , we aim to predict an offset  $\Delta w$  for each condition with more precise control,

$$w' = [w_c, w_m + \Delta w_m, w_f + \Delta w_f]. \quad (1)$$

For input text condition, we utilize recent powerful joint representations of Contrastive Language-Image Pre-training(CLIP)[Radford et al. 2021] to encode  $t$  as a text embedding  $E_t$ . CLIP has successfully been used in many visual-text tasks. For input texture patch, we use a pretrained VGG Network to capture the rich variation in texture patterns, following [Men et al. 2020], and obtain the texture embedding as  $E_p$ . With these two embeddings, we build two different mappers (*i.e.*, type mapper  $M_{tp}$  and texture mapper  $M_{txr}$ ) to learn the manipulation respectively by fusing input latent codes and conditions together as  $\Delta w_m = M_{tp}(w_m, E_t)$  and  $\Delta w_f = M_{txr}(w_f, E_p)$ .

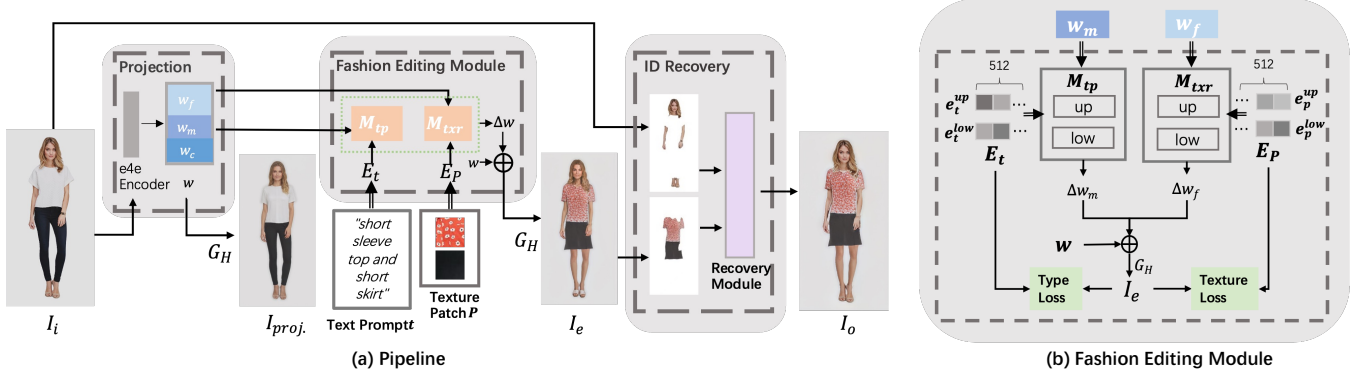
When designing our model, because of the unique feature of fashion, we find that the distributions of upper and lower fashion clothes are quite different. For example, we observe that the upper part of clothes often have more subtle changes in shapes, such as long sleeve versus short sleeve and round neck versus v-neck, while the lower part may have changes in structure, such as changing pants to a skirt. To better decoupling these two parts, we separately modeling them with two part-aware conversion modules in each mapper. Each part conversion module consists of a modulation module to capture the condition-based guidance. The structure of the modulation module is derived from [Huang and Belongie 2017; Park et al. 2019; Tan et al. 2021; Wei et al. 2022]. Taking the type branch (*i.e.* text prompt) as an example, we split the input text before sending into the CLIP for obtaining  $E_t = \{e_t^{up}, e_t^{low}\}$ . After fusing them separately with  $w_m$ , the output feature  $\Delta w_m^{up}$  and  $\Delta w_m^{low}$  would be added together to get the offset latent code  $\Delta w_m$  for the whole-body type adjustment. More specifically, the text embedding  $e_t^{up}$  and  $e_t^{low}$  are first fed into two fully-connected layers to obtain  $\gamma^{up}$ ,  $\beta^{up}$  and  $\gamma^{low}$ ,  $\beta^{low}$ , respectively. The complete process can be formulated as follows:

$$\Delta w_m^i = \beta^i + \gamma^i \frac{w_m^i - \mu_{w_m^i}}{\sigma_{w_m^i}}, i \in [up, low] \quad (2)$$

where  $\mu_{w_m^i}$  is the mean of  $w_m^i$  and  $\sigma_{w_m^i}$  is the variance. Similarly, we follow the same process to obtain the offset  $\Delta w_f$  for the fashion texture.

#### 3.2 Loss Functions

One of the biggest challenges of our task is training without annotated dataset as collecting large-scale pairwise data containing condition inputs and portrait images is impractical. To make the model trainable, we design a novel type loss, allowing the model



**Figure 2: (a) Overview of our pipeline. Our framework contains three modules: latent code projection, fashion editing, and ID recovery modules. In projection, we use e4e encoder to invert input image  $I_i$  to latent code  $w = [w_c, w_m, w_f]$ .  $I_{proj}$  is the reconstruction result from  $w$  using StyleGAN-Human generator  $G_H$ . In fashion editing module, we use two mappers to handle type and texture editing. We feed part-aware editing modules with text embedding  $E_t$  and texture patch embedding  $E_p$  to produce the offsets,  $\Delta w_m$  and  $\Delta w_f$  that lead to attribute changes in StyleGAN-Human latent space. The edited fashion result  $I_e$  is generated from  $(w + \Delta w)$  using  $G_H$  and will further gone through an ID recovery module to obtain the final output image  $I_o$ , maintaining the human characteristics of the input  $I_i$ . (b) The fashion editing module.**

to precisely adjust the cloth types without touching the remaining characteristics of the input portrait, such as face, skin, and pose. To achieve high-quality results, we also introduce other auxiliary losses, which will be introduced next.

**Type Loss.** Given the powerful visual-text representation power, one may think that a straightforward way is to directly calculate the cosine distance between the edited fashion design  $I_e$  and text prompts  $t$ . However, this does not work in our case as directly calculating the loss in a global manner will lead the model to overlook details. Another possible solution is to use a mask to help the model concentrate on the modifiable regions, but this limits the recognition power of the original CLIP embedding. To tackle the above problem, we exploit the linear operations supported by CLIP embeddings [Couairon et al. 2022; Jia et al. 2021]. Based on the cloth tags/types  $t_i$  label in original training data, we can compute a latent code for the unmodifiable region by subtracting the CLIP embedding of  $t_i$  from the CLIP embedding of input portrait image  $I_i$ , as  $E_{I_{un}} = E_{I_i} - E_{t_i}$ . This is benefited from the well aligned embedding space of CLIP for text and image modalities. Then, we can obtain a calibrated ground truth CLIP embedding by adding this unmodifiable embedding on to the CLIP embedding of target text prompt  $E_t$  as  $\tilde{E}_t = E_{I_{un}} + E_t$ . Thus, we can compute the difference between this calibrated ground truth and our obtained results in a more accurate manner with:

$$L_{type} = 1 - \cos(E_{I_e}, \tilde{E}_t). \quad (3)$$

**Texture Loss.** We design a texture loss to transfer the texture of input patch to the edited fashion clothes for the upper and lower parts, respectively. To emphasize the local spatial pattern in the reference texture  $P$ , we compute the feature correlations for RGB texture patches. More specifically, we first obtain the feature maps for an image patch by pretrained VGG-19 [Simonyan and Zisserman 2014], and extract the outputs for the last four layers, which are more relevant to the pixel-level characteristics. For the feature map  $F_i$  of each layer, we calculate the correlations by the Gram

matrix [Portilla and Simoncelli 2000], i.e.  $G_i = F_i F_i^T$ ,

$$L_{txr} = \sum_{i=1}^4 \|G_i(I_e^{crop}), G_i(P)\|_1, \quad (4)$$

where  $I_e^{crop}$  is a random cropped patch from the corresponding semantic region of  $I_e$ . For example, when the textual condition is "pants", the patch  $I_e^{crop}$  is fetched from the lower region of  $I_e$  according to the human parsing results  $\mathcal{P}$  from [Ak et al. 2019a].

**Reconstruction Loss.** To better preserve the unchangeable regions, we further add a set of reconstruction losses for preserving identity  $L_{id}$ , background (i.e., other non-cloth regions)  $L_{bg}$  and skin color  $L_{skin}$ .

**Identity loss.** We rely on the pretrained ArcFace network [Deng et al. 2019]  $Arc(\cdot)$  to keep the face identity by calculating the cosine distance between the original and edited image in ArcFace embedding space:

$$L_{id} = 1 - \cos(Arc(I_e), Arc(I_i)). \quad (5)$$

To obtain the background region, we use the human parsing results  $\mathcal{P}$  in binary format generated from [Rao et al. 2022] by removing the cloth region. We then obtain the background loss to preserve the non-cloth region by calculating the  $L_2$  distance:

$$L_{bg} = \|(I_e * \mathcal{P}_{bg}(I_e) - I_i * \mathcal{P}_{bg}(I_i))\|_2. \quad (6)$$

**Skin color loss.** Though background loss has constrained on the skin also, we find that further add a loss to help better preserve the skin color is necessary. Similar to the background, we obtain the skin parsing binary mask as  $\mathcal{P}_{skin}$ . We then convert the colors within this region into LAB color space which is more aligned with human perception. The skin color loss is to constrain the average color changes in the corresponding skin region with  $L_1$  as:

$$L_{skin} = \|(\text{Avg}(\text{Lab}(I_e) * \mathcal{P}_{skin}(I_e)) - \text{Avg}(I_i * \mathcal{P}_{skin}(I_i)))\|_1 \quad (7)$$

*Regularization Loss.* We also apply an L2 regularization loss on  $\Delta w$  to enable stable training without generating too large offsets as:

$$L_{norm} = \|\Delta w\|_2. \quad (8)$$

In summary, the final loss for training FashionTex to generate edited fashion image is:

$$L = \lambda_{type} L_{type} + \lambda_{txr} L_{txr} + \lambda_{id} L_{id} + \lambda_{skin} L_{skin} + \lambda_{bg} L_{bg} + \lambda_{norm} L_{norm}, \quad (9)$$

where  $\{\lambda_i\}$  are weighted parameters.

### 3.3 Identity (ID) Recovery Module

As mentioned at the start of this section our model relies on the inversion model for converting the input portrait images into the StyleGAN editable latent codes. However, we find that the inversion methods often struggle with the trade-off between the ability of editing and reconstruction. Especially our fashion images share more diverse appearances, causing reconstruction errors noticeable even with the state-of-the-art inversion methods. We follow [Fu et al. 2022; Tzaban et al. 2022] use PTI inversion [Roich et al. 2022] to obtain the latent codes. since PTI inversion needs to finetune the generator for each of the image for obtaining good result, it fails to directly output satisfied results for our task, especially on identities of portraits. A simple answer may be copying the edited fashion cloth region back to the portrait with a guided semantic binary mask. Unfortunately, this does not work as our type changes often adjust the shape of clothes. For example, when adjusting the sleeves to be short, pasting back to the portrait image can generate serious artifacts with some parts of the original sleeves remaining.

To alleviate the identity loss in our final results, while maintaining the well-modified fashion cloths based on input conditions, we use the regularization ability of the StyleGAN space to compensate for the artifacts in part fusion. We design a semantic-aware ID recovery module to obtain satisfied results. In particular, we first fuse the clothes regions in the edited image  $I_e$  using a binary semantic mask [Rao et al. 2022] to gain a guided image  $I'_e = \mathcal{P}_{cloth}(I_e) * I_e + \mathcal{P}_{bg}(I_e) * I_i$ . We then finetune the StyleGAN-Human generator similar to the PTI inversion guiding with LPIPS perceptual [Zhang et al. 2018] and  $L_2$  distances to get the refined output  $I_o$ :

$$L_{ID} = L_{LPIPS}(I'_e, I_o) + \|\mathcal{P}_{bg}(I_o) * (I_i - I_o)\|_2. \quad (10)$$

To obtain  $I_o = G_H(I'_e; \theta^*)$ , we define the optimization as:

$$\theta^* = \arg \max_{\theta^*} L_{ID}, \quad (11)$$

where  $\theta^*$  is the parameters set of the generator  $G_H$ .

## 4 EXPERIMENTS

### 4.1 Implementation Details

We train and evaluate our method on Deepfashion-MultiModal dataset [Jiang et al. 2022; Liu et al. 2016]. It contains 12,701 full-body human images with human parsing labels of 24 classes and descriptions of each image. But in this work, there is no need for segmentation maps or pair-wise captions, we only use the cloth-type labels. We first use the full-body image alignment method with the mean body midpoint mentioned in [Fu et al. 2022] to process

the images and abandon the samples with bad alignment results. For the remaining data, we randomly split 11,265 and 1,136 data for the train and test sets, respectively. For the text prompts, we borrow the common practice in online clothing stores, which clusters the garments in attribute combinations, e.g. "sleeveless top, and short skirt". For the reference texture patch, we crop texture patches from datasets [Jiang et al. 2022; Liu et al. 2016] to get realistic clothing textures.

We utilize pretrained StyleGAN2 model [Fu et al. 2022; Sarkar et al. 2021] as the generator and pretrained e4e encoder [Tov et al. 2021] as the image encoder to invert images into StyleGAN's latent codes. The dimensions of the latent code are  $18 * 512$ . We keep the code from the coarse layer of StyleGAN unchanged and only perform editing on medium and fine layers.

### 4.2 Results of Multi-modal-guided Fashion Editing

We show our results on multi-modal fashion editing in Fig. 3. As we are the first to work on this new task, there are no available methods for direct comparisons. We thus show our complete results here and leave the evaluations on the individual modality in the following subsections. As can be seen from Fig. 3, our FashionTex can deal with various input text prompts and texture patches. The input text prompts can vary from general high-level descriptions "skirt" to fine-grained ones such as "camisole dress" without any pairwise annotated training data. Besides, our model can preserve the input identity and pose well for achieving satisfying try-on results. Even for some less frequently seen design cases (e.g., the 3<sup>rd</sup> output of the 1<sup>st</sup> case in the 2<sup>nd</sup> row.), our model can generate reasonable results. An interesting finding is that FashionTex can automatically find the more suitable cloth even for the same input to meet common sense. For example, as shown in the 2<sup>nd</sup> output of the 2<sup>nd</sup> case each row, it generates long tight joggers for the lady (i.e., upper row), while generating short loose joggers for the man (i.e., lower row). These findings support the effectiveness of our model, and we show more detailed evaluations below.

### 4.3 Comparisons on Fashion Type Editing

*Baseline methods.* To verify the effectiveness of our model on cloth type editing, we compare our method with two state-of-art text-driven image manipulation works. (1) TediGAN [Xia et al. 2021] proposes a visual-linguistic similarity module to project the image and text into a common embedding space. After inverting a fashion portrait into the joint latent space, it can achieve the type editing by changing the text embedding for cloth. We follow the official CLIP-based implementation and the default settings for training. (2) StyleCLIP [Patashnik et al. 2021] proposes three latent mappers to control the different latent group (i.e.  $\{w_c, w_m, w_f\}$ ) in the  $W+$  space of StyleGAN, conditioned on a textual description and a source image. We follow the official instruction to train StyleClip, except that we only apply the mapper for the medium layer  $w_m$  to perform the type editing task. For each text input, StyleClip needs to train corresponding mappers to perform the editing. As for FashionTex, we only use the type mapper  $M_{Tp}$  with  $w_m$  and remove the proposed ID recovery module for fair comparison since these two methods do not have abilities to recover identities.



**Figure 3: Our results on multi-modal editing on fashion clothes try-on. With simple interaction by text prompts and texture patches, our model can generate try-on results meeting the requirement while keeping the input portrait identity.**

*Metrics.* To evaluate the performance of each method, we use two metrics: 1) *Accuracy*. It measures whether the model succeeds in getting the target cloth type. We use a human parsing network[Rao et al. 2022] pretrained on [Liu et al. 2016] to find if the target cloth type is in the manipulated image. Since the parsing results can only reflect the category of clothes (e.g., skirt, pants) and cannot identify more specific attributes (e.g., sleeve length), we choose four common cloth categories for evaluation, i.e., skirt, pants, dress, and rompers. 2) *FID*. It measures the realism of the generated images by computing the Wasserstein-2 distance between distributions of the generated images and the corresponding type of images in our dataset.

*Qualitative Results.* The visual comparisons for type editing are shown in Fig. 4. When the target cloth type is close to the source image, e.g. from "shirt" to "polo shirt", all methods can achieve reasonable results. However, when there are large changes in cloth structure, e.g. from "pants" to "skirt", both TediGAN and StyleCLIP struggle to change the original cloth type, while our FashionTex generates the target fashion style as the textual condition. Since our clip loss can pay more attention to the area that needs to be edited, we achieve good results on both subtle attribute transformations and large type changes. Our results have advantages in the preservation of both color and facial information using our reconstruction and regularization losses.

*Quantitative Results.* We show the quantitative results in Tab. 1, and our model outperforms the previous works for a large margin

**Table 1: Quantitative Comparison for type editing(above) and texture transfer(below).**

	Method	FID ↓	Accuracy ↑
Type	Styleclip[Patashnik et al. 2021]	90.25	22.25%
	Tedigan[Xia et al. 2021]	95.44	15.25%
	Ours	<b>69.22</b>	<b>82.75%</b>
	Method	FID ↓	LPIPS ↓
Texture	TextureGAN[Xian et al. 2018]	225.28	0.4070
	Texture Reformer[Wang et al. 2022]	189.68	0.3687
	DiOr[Cui et al. 2021]	226.18	0.3784
	Ours	<b>184.85</b>	<b>0.3257</b>

**Table 2: Ablation study of our proposed Type Loss and Text Splitting method.**

Method	FID ↓	Accuracy ↑
w/o Type Loss	172.01	58.75%
w/o Text Splitting	97.80	10.75%
Ours	<b>69.22</b>	<b>82.75%</b>

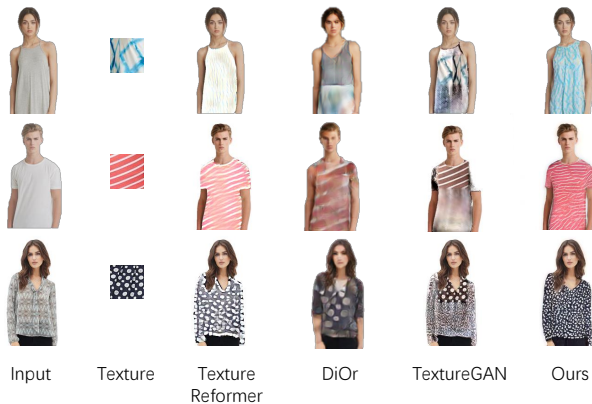
on both metrics, further demonstrating the advantage of our model on fashion type editing.

#### 4.4 Comparisons on Fashion Texture Transfer.

*Baseline methods.* To verify the effectiveness of our FashionTex on the task of texture transfer, three representative texture-guided



**Figure 4: Qualitative comparisons on fashion type editing. We compare our method with two state-of-the-art methods: StyleCLIP[Patashnik et al. 2021] and TediGAN[Xia et al. 2021].**



**Figure 5: Qualitative comparisons on texture transfer task. We compare our method with three state-of-the-art methods: Texture Reformer[Wang et al. 2022], DiOr[Cui et al. 2021], and TextureGAN[Xian et al. 2018].**

generative methods are selected: (1) TextureGAN [Xian et al. 2018]. We follow the default implementation of TextureGAN and replace the input sketch with the ground-truth segmentation map for a fair comparison. (2) Dress in Order (DiOr) [Cui et al. 2021]. We follow the implementation of the Texture Transfer part and only change the texture input for the same comparison. (3) Texture Reformer [Wang et al. 2022]. In this experiment, the texture image is taken as a style image, and the input image is taken as a content image.

**Metrics.** To quantitatively evaluate the quality of the synthesized images, we adopt two widely used evaluation metrics, which are

Fréchet Inception Distance(FID) [Heusel et al. 2017] and Learned Perceptual Image Patch Similarity (LPIPS)[Zhang et al. 2018].

**Qualitative Results.** The visual comparisons of texture transfer are shown in Fig. 5. It is obvious that our approach achieves the best performance within all methods. As shown in the third row of Fig. 5, the quality of the generated image by Texture Reformer is not very bad for the solid-colored texture patch. But for other patch styles, like lines, its performance looks poor. As for DiOr, the visual results look much blurry for all texture cases. The reason is that their pre-train model is overfitted, and we can not get sharp outcomes for all test samples when we change the texture patches by ourselves.

**Quantitative Results.** Table 1 reports the quantitative comparisons between our FashionTex and the baselines. The advantages of our approach are obviously shown in FID and LPIPS scores, which confirms the strengths of our model on texture transfer.

## 4.5 Ablation Study

**The Effect of Multi-modal Interaction.** In our work, we use interactions in different modalities to guide the type and texture editing differently. To demonstrate the effectiveness of this interaction method, we compare it with using only text to guide both type and texture editing. To be more specific, we replace the input texture patch reference with texture descriptions. We use three common kinds of texture, *Plaid*, *Floral*, and *Striped*, with color descriptions, such as *red*, *black*, and *blue*. The results can be seen in Fig. 6. Compared with text, the reference texture patch can bring more diverse and precise texture results. For example, the stripe pattern tends to be the same scale giving a text description for texture generation.



**Figure 6: The results of using text prompts for both type and texture editing. The upper text description represents the target type, and below is the texture description or reference texture patch. We keep the lower cloth’s texture and only show the texture condition for upper cloth.**



**Figure 7: The effect of type Loss and ID recovery module. The input text prompt is *sleeveless dress*.**

*The Effect of Type Loss.* We compare our type loss with the naive CLIP loss (see Eq. 4). The comparison results are shown in Fig. 7. As can be seen, using naive clip loss can achieve the change of sleeve length, but it fails to change the dress. With our type loss, the model pay more attention to the target area, which can make the image match the input condition better. We use the same metrics as type comparison, and the quantitative comparisons are shown in Tab. 2. The first row result is the performance of naive CLIP loss.

*The Effect of ID Recovery Module.* For the ID recovery module, we directly show the results before and after this module (see Fig. 7). It should be noted that the identity has been lost caused by the inversion before adding our ID recovery module, while ours achieve satisfied result.

*The Effect of Text Splitting.* Our fashion editing module separates text prompts into descriptions of upper and lower cloth parts. This explicit separation is designed based on the structure of the human body and leads conditions better focus on the corresponding body part. And we use the same metrics to measure the editing ability. Tab. 2 illustrates the effectiveness of this design.

## 5 CONCLUSION

We introduce a novel and practical task of using multi-modal interactions (*i.e.*, textual description and texture image patch) for virtual

fashion cloth try-on. Based on a StyleGAN structure, we propose a new FashionTex pipeline that can generate high-quality results meeting the input conditions without modifying the identity of the input full-body fashion portrait. The key to our pipeline is a fashion editing module for obtaining the corresponding fashion editing displacement without pairwise training data, and an ID recovery module to preserve personal identity. Experiments have been conducted to demonstrate the effectiveness of our FashionTex. In summary, we believe that our interaction way is a powerful editing tool for virtual try-on, and hope to inspire future works along this research line.

## ACKNOWLEDGMENTS

The work was supported in part by NSFC with Grant No. 62293482, the Basic Research Project No. HZQB-KCZY-2021067 of Hetao Shenzhen-HK S&T Cooperation Zone, the National Key R&D Program of China with grant No. 2018YFB1800800, the Shenzhen Outstanding Talents Training Fund 202002, the Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104, the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), and the Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. ZDSYS201707251409055). It was also supported in part by Outstanding Young Fund of Guangdong Province with No. 2023B1515020055. It was also sponsored by CCF-Tencent Open Research Fund.

## REFERENCES

2022. Amazon Statistics (2022).
- Kenan Emir Ak, Joo Hwee Lim, Jo Yew Tham, and Ashraf Kassim. 2019a. Semantically consistent hierarchical text to fashion image synthesis with an enhanced-attentional generative adversarial network. In *ICCVW*.
- Kenan E Ak, Joo Hwee Lim, Jo Yew Tham, and Ashraf A Kassim. 2019b. Attribute manipulation generative adversarial networks for fashion images. In *CVPR*.
- Badour AlBahar and Jia-Bin Huang. 2019. Guided image-to-image translation with bi-directional feature transformation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9016–9025.
- Badour AlBahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. 2021. Pose with Style: Detail-preserving pose-guided image synthesis with conditional stylegan. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–11.
- Andrew Brown, Cheng-Yang Fu, Omkar Parkhi, Tamara L. Berg, and Andrea Vedaldi. 2022. End-to-End Visual Editing with a Generatively Pre-Trained Artist. In *ECCV*.
- Lele Chen, Justin Tian, Guo Li, Cheng-Haw Wu, Erh-Kan King, Kuan-Ting Chen, Shao-Hang Hsieh, and Chenliang Xu. 2020. Tailorgan: Making user-defined fashion designs. In *WACV*.
- Guillaume Couairon, Asya Grechka, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. 2022. FlexIT: Towards Flexible Semantic Image Translation. In *CVPR*.
- Aiyu Cui, Daniel McKee, and Svetlana Lazebnik. 2021. Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14638–14647.
- Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. 2019. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*.
- Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *CVPR*.
- Anna Frühstück, Krishna Kumar Singh, Eli Shechtman, Niloy J Mitra, Peter Wonka, and Jingwan Lu. 2022. InsetGAN for Full-Body Image Generation. In *CVPR*.
- Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen Change Loy, Wayne Wu, and Ziwei Liu. 2022. StyleGAN-Human: A Data-Centric Odyssey of Human Generation. In *ECCV*.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.
- Mehmet Günel, Erkut Erdem, and Aykut Erdem. 2018. Language guided fashion image manipulation with feature-wise transformations. *arXiv preprint arXiv:1808.04000* (2018).
- Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. 2018. Viton: An image-based virtual try-on network. In *CVPR*.



- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).
- Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*.
- Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR* (2017).
- Thibaut Issenbuth, Ugo Tanielian, Jérémie Mary, and David Picard. 2021. EdiBERT, a generative model for image editing. *arXiv preprint arXiv:2111.15264* (2021).
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*.
- Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Zifei Liu. 2022. Text2human: Text-driven controllable human image generation. *ACM TOG* (2022).
- Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).
- Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *CVPR*.
- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *CVPR*.
- Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. 2022. DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation. In *CVPR*.
- Gihyun Kwon and Jong Chul Ye. 2022. Clipstyler: Image style transfer with a single text condition. In *CVPR*.
- Sangyun Lee, Gyojung Gu, Sunghyun Park, Seunghwan Choi, and Jaegul Choo. 2022. High-Resolution Virtual Try-On with Misalignment and Occlusion-Handled Conditions. In *ECCV*.
- Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. 2021. Tryongan: Body-aware try-on via layered interpolation. *ACM TOG* (2021).
- Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *CVPR*.
- Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. 2020. Controllable Person Image Synthesis with Attribute-Decomposed GAN. In *Computer Vision and Pattern Recognition (CVPR), 2020 IEEE Conference on*.
- Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. 2020. Image based virtual try-on network from unpaired data. In *CVPR*.
- Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*.
- Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *CVPR*.
- Javier Portilla and Eero P Simoncelli. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision* (2000).
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.
- Amit Raj, Patsorn Sangkloy, Huiwen Chang, James Hays, Duygu Ceylan, and Jingwan Lu. 2018. Swapnet: Image based garment transfer. In *ECCV*.
- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
- Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. 2022. Densclip: Language-guided dense prediction with context-aware prompting. In *CVPR*.
- Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2022. Pivotal tuning for latent-based editing of real images. *ACM TOG* (2022).
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*.
- Kripasindhu Sarkar, Vladislav Golyanik, Lingjie Liu, and Christian Theobalt. 2021. Style and pose control for image synthesis of humans from a single monocular view. *arXiv preprint arXiv:2102.11263* (2021).
- Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. 2020. Neural re-rendering of humans from a single image. In *ECCV*.
- Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. 2020. Interfacegan: Interpreting the disentangled face representation learned by gans. *PAMI* (2020).
- Yujun Shen and Bolei Zhou. 2021. Closed-form factorization of latent semantics in gans. In *CVPR*.
- Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
- Zhentao Tan, Dongdong Chen, Qi Chu, Menglei Chai, Jing Liao, Mingming He, Lu Yuan, Gang Hua, and Nenghai Yu. 2021. Efficient semantic image synthesis via class-adaptive normalization. *IEEE TPAMI* (2021).
- Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an encoder for stylegan image manipulation. *ACM TOG* (2021).
- Rotem Tzaban, Ron Mokady, Rinon Gal, Amit Bermano, and Daniel Cohen-Or. 2022. Stitch it in time: Gan-based facial editing of real videos. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.
- Bochao Wang, Huaibin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. 2018. Toward characteristic-preserving image-based virtual try-on network. In *ECCV*.
- Zhizhong Wang, Lei Zhao, Haibo Chen, Ailin Li, Zhiwen Zuo, Wei Xing, and Dongming Lu. 2022. Texture Reformer: Towards Fast and Universal Interactive Texture Transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Zhentao Tan, Lu Yuan, Weiming Zhang, and Nenghai Yu. 2022. Hairclip: Design your hair by text and reference image. In *CVPR*.
- Zongze Wu, Dani Lischinski, and Eli Shechtman. 2021. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*.
- Weihao Xia, Yujun Yang, Jing-Hao Xue, and Baoyuan Wu. 2021. TediGAN: Text-Guided Diverse Face Image Generation and Manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. 2018. Texturegan: Controlling deep image synthesis with texture patches. In *CVPR*.
- Zhenyu Xie, Zaiyu Huang, Fuwei Zhao, Haoye Dong, Michael Kampffmeyer, and Xiaodan Liang. 2021a. Towards Scalable Unpaired Virtual Try-On via Patch-Routed Spatially-Adaptive GAN. *NeurIPS* (2021).
- Zhenyu Xie, Xujie Zhang, Fuwei Zhao, Haoye Dong, Michael C Kampffmeyer, Haonan Yan, and Xiaodan Liang. 2021b. Was-vton: Warping architecture search for virtual try-on network. In *ACM MM*.
- Zipeng Xu, Tianwei Lin, Hao Tang, Fu Li, Dongliang He, Nicu Sebe, Radu Timofte, Luc Van Gool, and Errui Ding. 2022. Predict, Prevent, and Evaluate: Disentangled Text-Driven Image Manipulation Empowered by Pre-Trained Vision-Language Model. In *CVPR*.
- Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 2021. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627* (2021).
- Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. 2019. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In *ICCV*.
- Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.
- Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *ECCV*.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017a. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.
- Shizhan Zhu, Raquel Urtasun, Sanja Fidler, Dahua Lin, and Chen Change Loy. 2017b. Be your own prada: Fashion synthesis with structural coherence. In *ICCV*.