Text-Guided Human Image Manipulation via Image-Text Shared Space

Xiaogang Xu, Ying-Cong Chen, Xin Tao, and Jiaya Jia, *Fellow, IEEE*

Abstract—Text is a new way to guide human image manipulation. Albeit natural and flexible, text usually suffers from inaccuracy in spatial description, ambiguity in the description of appearance, and incompleteness. We in this paper address these issues. To overcome inaccuracy, we use structured information (e.g., poses) to help identify correct location to manipulate, by disentangling the control of appearance and spatial structure. Moreover, we learn the image-text shared space with derived disentanglement to improve accuracy and quality of manipulation, by separating relevant and irrelevant editing directions for the textual instructions in this space. Our model generates a series of manipulation results by moving source images in this space with different degrees of editing strength. Thus, to reduce the ambiguity in text, our model generates sequential output for manual selection. In addition, we propose an efficient pseudo-label loss to enhance editing performance when the text is incomplete. We evaluate our method on various datasets and show its precision and interactiveness to manipulate human images.

Index Terms—Human Image Manipulation, Adversarial Generative Networks, Image and Text.

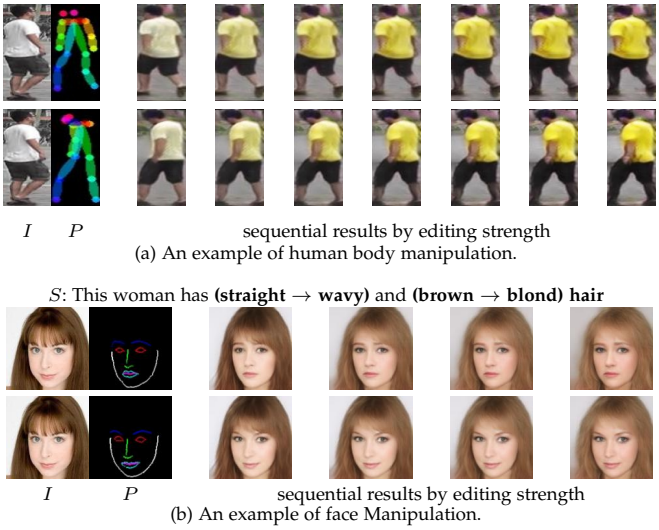
1 Introduction

Popularity of social networks stimulates online photo sharing, which requires development of important photo editing tools. In this paper, we aim to enable users to customize their photos in a natural and comfortable way by seeking an intelligent model that can communicate with users by natural language. This line of development is termed as "text-guided image manipulation".

Existing text-guided image manipulation methods [1], [2], [3], [4], [5], [6], [7] formulate the task as a conditional image generation process. A text encoder is employed to extract semantic features of input text, and an image encoder obtains the deep representation of images. Then a generative adversarial network [8] is applied to synthesize images based on the text and image features. Ideally, generated images reflect visual change according to the given textual descriptions.

There are still a few critical problems that need to be addressed. They include improvement accuracy of manipulation, reducing ambiguity of natural language, and using possibly incomplete description provided by users. To elaborate, first, it is difficult to locate correct regions to manipulate through text, since text lacks precise spatial or structural information in general. We provide an example in Fig. 1. When the "yellow shirt" is mentioned in the sentence, it means only shirt color is to change while other places should keep untouched. The system needs to identify diverse parts correctly.

Second, natural language is ambiguous. It is hard to specify an exact degree for manipulating appearance with only text. For example, in Fig. 1(a), "yellow" can be dark or light – it is unknown which is preferred by users based solely on language. Thus, instead of outputting a single edited image, a more desired way is to generate a series of images for users to choose. Third, users may face difficulties in giving full description. The task is therefore to complete it based on common sense.



(a) An example of human body manipulation.

(b) An example of face Manipulation.

Fig. 1. Our framework allows users to manipulate appearance of image I with textual input S, where strength is controllable. Moreover, spatial information is editable by adjusting pose input P.

In this paper, we address these major issues by proposing a flexible framework. To accurately locate regions to edit, our approach extracts structural information (e.g., face landmark and poses for pedestrians) from original images and uses it to disentangle appearance and spatial. It has been proved that embedding multimodal inputs into a

- *Xiaogang Xu and Jiaya Jia are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong (CUHK). E-mail: xgxu@cse.cuhk.edu.hk, leojia@cse.cuhk.edu.hk*
- *Ying-Cong Chen is with the Computer Science and Artifical Intelligence Lab, Massachusetts Institute of Technology (MIT). E-mail: yingcong.ian.chen@gmail.com*
- *Xin Tao is with Kuaishou Technology. E-mail: jiangsutx@gmail.com*
- *Corresponding Author: Ying-Cong Chen.*

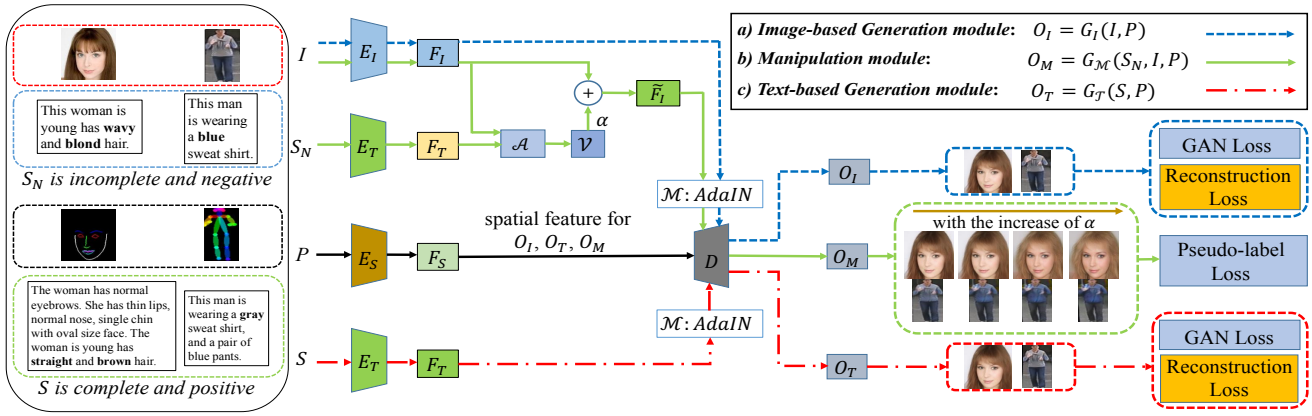


Fig. 2. Our framework is composed of manipulation module $G_{\mathcal{M}}$, image-based generation module G_I and text-based generation module $G_{\mathcal{T}}$. a) Image-based generation module $G_I(I, P)$ adopts the pose encoder E_S to obtain spatial feature F_S from P and image feature F_I with the image encoder E_I. It uses AdaIN-based decoder D to complete generation. b) Manipulation module $G_{\mathcal{M}}(S_N, I, P)$ is built upon $G_I(I, P)$. Text input S_N is embedded into the same feature space with image feature through text encoder E_T. The attribute vector $\mathcal{V}$ is computed from the comparison between text and image feature in shared space through $\mathcal{A}$. $\mathcal{V}$ updates F_I to $\tilde{F}_I$, which generates output through D with F_S. The manipulation strength is controllable by adjusting weight α of $\mathcal{V}$. c) Text-based generation module $G_{\mathcal{T}}(S, P)$ gives output based on S and P to help the training of $\mathcal{A}$, where S describes I. O_I, O_M and O_T are output from these three modules respectively.

shared feature space can benefit several computer vision tasks [9], [10], and we in this paper propose to learn image-text shared space, which allows manipulating images with "attribute vectors" defined by the text descriptions. Similar motivation is the semantic latent space embedding [7], [11], [12], [13], [14], which allows manipulating image features with arithmetic operations.

Besides, we adjust manipulation strength by setting varying moving distances in latent space. Our framework allows for outputting a series of generation results for each source image. This gives users high freedom to choose satisfying results and reduce ambiguity of textual instruction. As illustrated in Fig. 1, when taking the human body (pedestrian) and face synthesis as examples, our method generates a series of results. In addition, our pseudo-label loss improves the manipulation performance when the text input is incomplete.

Extensive experiments are conducted on CUHK-PEDES [15] and CelebA [16] to demonstrate effectiveness and generality of our framework. Compared with state-of-the-art approaches, our framework achieves text-guided human image manipulation with high accuracy and quality. A video is provided to demonstrate that our approach generates results in a user-friendly way.

2 Related Work

There have been a few methods using text to guide image manipulation, which can be divided into two main categories according to whether they adopt deep generative models or not. The methods not using deep generative models normally adopt rule-based strategies instead [17], [18]. These rule-based methods would pre-define the language templates to parse the editing request, and then call the off-the-shelf operations. However, such strategies are limited by capacities of the corresponding language templates.

Recent work designs frameworks with diverse editing targets, by adopting deep generative models [1], [2], [3], [4], [6], [19], [20], [21], [22], [23], [24] – especially Adversarial Generative Network (GAN) [8]. Chen et al. [6] colorized source images conditioned on the description of the input language, and Cheng et al. [21] completed the same task allowing for sequence generation. Wang et al. [19] modified the brightness of photos with textual instruction. Zhu et al. [4] and Gunel et al. [20] let the system learn to manipulate fashion images via textual descriptions. Nam et al. [1] and Dong et al. [2] both illustrated the effectiveness of their frameworks by manipulating the appearance of birds and flowers. Li et al. [3] proposed a framework with hierarchical structure and attention mechanism for general purposes.

All these methods only allow users to edit attributes of appearance instead of spatial information. Unlike all previous work, our framework edits images not only with appearance change but also by adjusting object layout (e.g., poses in pedestrian).

Albeit various frameworks, existing approaches implement conditional text modeling by simply concatenating text features with image ones [1], [2], [3], [4], [19], [20], which do not learn matching between textual descriptions and visual components. Our method is advantageous by learning image-text space, where any image editing operations can be realized by shifting image features towards the direction of attributes changes defined by textual instruction.

Text-to-image generation [25], [26], [27], [28], [29], [30] is another task to synthesize images with input language as a condition. Such approaches aim to generate an image whose content is consistent with the description of given natural language. Existing approaches usually achieve consistency by adversarial training with text as condition [31], [32], or adopting reconstruction loss [29].

3 Our Framework

Our framework achieves text-guided human image manipulation in a general way, which involves both *edit for attribute of appearance* (e.g., color) and *spatial editing* (e.g., pose and expression modification). The former is achieved

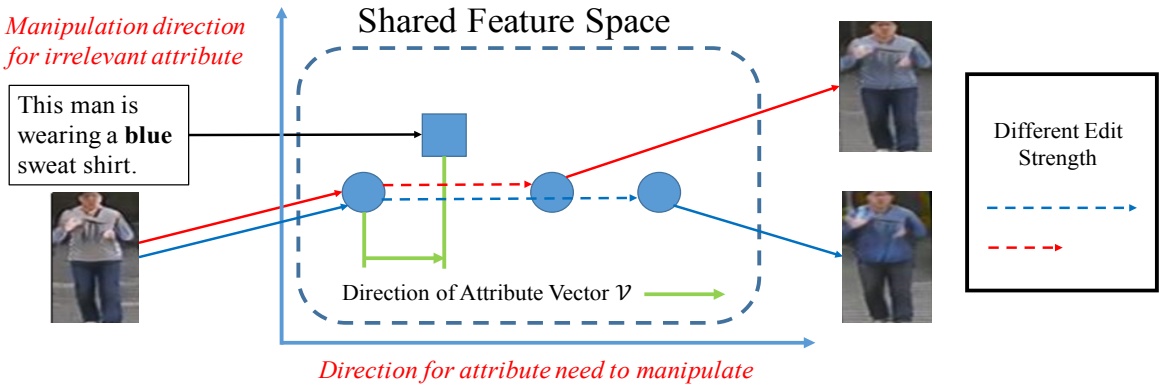


Fig. 3. Illustration of manipulation in image-text latent space. Both text and image are mapped to the same space, while they are not aligned in "direction for irrelevant attributes" since text may only cover part of concerned attributes. Our attribute vector is constructed by projecting the difference onto the direction computed by $\mathcal{A}$. It thus modifies attributes in the image without involving irrelevant attributes. Editing strength can be controlled for a sequence of output.

by textual guidance, and the latter uses structural input, e.g., landmarks that determines the structure of image content.

Here, structural input helps in three aspects. First, structural input helps locate correct parts to manipulate, since textual instruction normally corresponds to specific parts of structure. For example, "shirt" normally locates in the upper part of pose for pedestrian. Such guidance is achieved by disentangling the control of appearance and spatial structure with our designed AdaIN-based structure [33], [34], [35], [36], [37].

Besides, we obtain image-text shared space with this disentanglement. It yields superior effect for editing appearance, and solves ambiguity of text. Finally, structural input is easy to estimate, and allows precise control of spatial structure. Compared with manipulating spatial structure through post-processing [38], [39], [40], [41], ours controls appearance and spatial structure simultaneously with real-time speed.

Our framework is composed of two parts as shown in Fig. 2, i.e., two generation modules (image-based generation module G_I and text-based generation module $G_\mathcal{T}$) and a manipulation module ($G_\mathcal{M}$). The generation part synthesizes target images based on source image input, textual description, and structure information. Especially, the shared space for image and text is learned by these two generation modules.

The manipulation module guides generation with structural information and outputs continuous sequences of synthesized results by varying editing strength. We take skeletons and landmarks as examples of structure information for pedestrian and face synthesis in this paper. Our method, in fact, can generalize to more types of structural input.

3.1 Manipulation Module

Since the pose input P provides accurate structural information [38], [42], [43], it remedies the spatial inaccuracy of text for manipulation. Information P is initially obtained by applying the pose inference network to either image or text [5], [44], [45], and is then adjusted by users. In our work, we extract poses of pedestrians using OpenPose [44], and obtain the face orientation using Dlib [45].

Understanding textual instruction and manipulating the image accordingly is not trivial. As shown in Fig. 3, our main idea is to learn image-text shared space, where any manipulation operations for appearance can be done by moving images in the latent space along specific *attribute vectors* extracted from the textual instruction, like translational embedding [7], [11]. Thus, we modify the image along the instructional direction. This operation is advantageous since the manipulation strength is controllable, which allows to generate a sequence of output from the same input. Users pick the one they like to refine details. This solution resolves the ambiguity for textual instruction.

3.1.1 Manipulation Procedure $O_M = G_\mathcal{M}(S, I, P)$

To manipulate image I with the text S and the structure information P, we obtain image feature F_I from I by the image encoder E_I and the spatial feature F_S from P by the spatial encoder E_S. To acquire text feature F_T from S, we set a text encoder E_T with LSTM structure [46]. We average the hidden feature of each token in S as the output feature of LSTM, denoted as F_T. Note that F_I and F_T are both vectors with the same shape, which are utilized to compute identical category of AdaIN parameters. They locate in the same feature space.

The manipulation is achieved with the attribute vector $\mathcal{V}$ generated from the attribute-vector generator $\mathcal{A}$ as

$$\begin{aligned} &\mathcal{V}_A = \mathcal{A}(F_I \oplus F_T), \tilde{F}_I = F_I + \alpha \times \mathcal{V}_A, \\ &F_S = E_S(P), O_M = D(F_S, \mathcal{M}(\tilde{F}_I)), \end{aligned} \tag{1}$$

where $\oplus$ denotes channel concatenation, D is the AdaIN-based decoder, $\mathcal{M}$ is used to compute the AdaIN parameter with MLP (multilayer perceptron) structure, and α is the weight for attribute vector. We denote this manipulation procedure as $O_M = G_\mathcal{M}(S, I, P)$.

The attribute vector $\mathcal{V}$ is derived from projection of the difference between image and text features in the shared space, on "direction of attributes to manipulate" obtained by $\mathcal{A}$. It enables us to edit image feature without changing irrelevant attributes that are not described in the text. Further, adjusting α (i.e., manipulation strength) helps synthesize sequential output for selection, to remedy imprecise textual description.

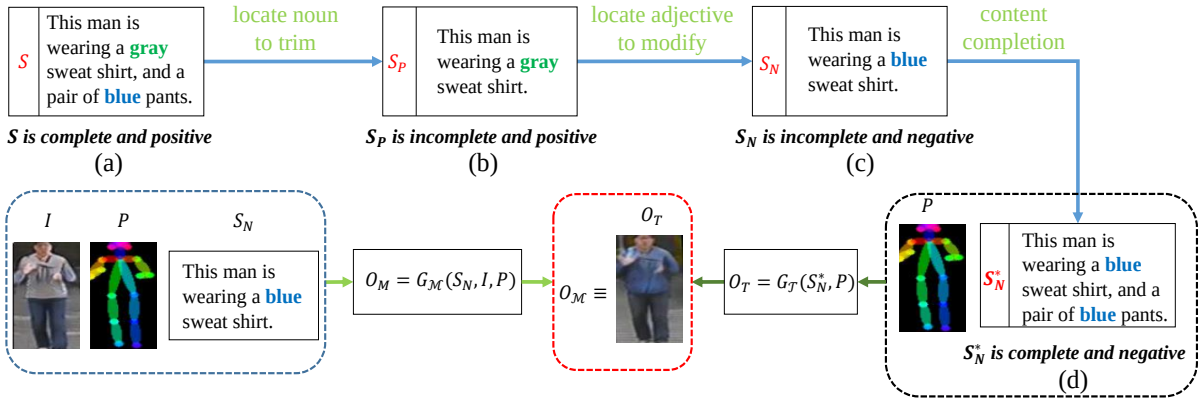


Fig. 4. The pseudo-label loss is designed to train manipulation module $G_{\mathcal{M}}(S, I, P)$, and improve performance when text input is incomplete and negative. For the text whose description is not consistent with content of I (like S_N), we use our trained text-to-image generation module $G_{\mathcal{T}}(S, P)$ to form pseudo labels to guide training.

3.1.2 Training the Manipulation Module

During training, for each image in the dataset, we have a textual description of S that fully describes its content and all concerned attributes. However, during testing, the textual description provided by users may only cover part of attributes. To address this problem, we propose a text augmentation approach that generates *incomplete* and *negative* textual description, aiming to cover more possibilities during practical use. Here *incomplete* text stands for the textual description that does not adequately describe all concerned attributes of the corresponding image (like S_P and S_N in Fig. 4), and *negative* text refers to the textual description that does not match the corresponding image (like S_N and S_N^* in Fig. 4).

Textual description generally describes the properties of specific attributes. For example, S in Fig. 4(a) provides descriptions about the color of shirt and pants. In these descriptions, the attribute name is usually a noun, and its corresponding properties are usually adjective.

Motivated by this observation, we generate incomplete text by trimming a passage of description on the location of a noun and generate negative text by randomly changing the adjective. We illustrate the procedure in Fig. 4(b) and (c). Specifically, we trim the full sentence S after the noun "shirt", and obtain an incomplete sentence S_P. Then the adjective "gray" is changed to "blue" to produce a negative sentence S_N. By randomly trimming the full descriptions and modifying adjective words, we obtain numerous incomplete and negative descriptions, covering most situations in practical use.

The loss of the manipulation module is composed of two parts, guiding this module to handle positive/negative textual descriptions respectively.

When inputted text is positive (that is, its content is consistent with the image), $G_{\mathcal{M}}$ should generate an image that is identical with the input. This constraint can be modeled with a reconstruction loss. In our implementation, we use both pixel- and feature-level [12], [47], [48], [49] reconstruction loss, which is defined as

$$\mathcal{L}_{\mathcal{M}_{pos}} = \sum_{i=0}^{5} \mathbb{E}(||\Phi_i(G_{\mathcal{M}}(S_{pos}, I, P)) - \Phi_i(I)||), \quad (2)$$

where $\mathbb{E}$ is the operation to compute mean value, S_{pos} denotes positive textual descriptions, either complete or not, and $\Phi_0(\cdot)$ is the raw pixel space. $\Phi_1(\cdot)$ to $\Phi_5(\cdot)$ are feature spaces defined by the ReLU1_2, ReLU2_2, ReLU3_3, ReLU4_3, ReLU5_3 layers of an ImageNet-pretrained VGG-16 network [50].

When the input text is negative (that is, the content of text is different from the input image), the manipulation module is expected to modify the image accordingly. In this case, there is no ground truth to train the model. Fortunately, the generation module, which will be described in the next section, can be used for constructing a pseudo ground truth for training.

Specifically, the generation module $G_{\mathcal{T}}(S, P)$ takes textual description and a pose feature as input and generates an image whose content is consistent with the textual description S. The pose input guarantees that pseudo labels have correct spatial structures. So for negative text, it is expected that the output from the manipulation module is consistent with that of the generation module once the text input is rational. This leads to *pseudo-label loss*, which can be formulated as

$$\mathcal{L}_{\mathcal{M}_{neg}} = \sum_{i=0}^{5} \mathbb{E}(||\Phi_i(G_{\mathcal{M}}(S_N, I, P)) - \Phi_i(G_{\mathcal{T}}(S_N^*, P))||), \quad (3)$$

where S_N denotes the incomplete and negative text, and S_N^* is the complete counterpart, as illustrated in Fig. 4(d). Note that we cannot use the incomplete text S_N in the generation module $G_{\mathcal{T}}$, as it does not describe all concerned attributes and would instead generate image that is vastly different from I. Moreover, the appearance of $G_T(S_N^*, P)$ does not need to be totally matched with the appearance of $G_{\mathcal{M}}(S_N, I, P)$, since we can further adjust details of manipulation results by controlling α.

3.2 Generation Module

There are two generation modules (image-based generation module $G_I(I, P)$ and text-based generation module $G_{\mathcal{T}}(S, P)$), which learn image-text shared space to help the training of manipulation module. Note $G_I(I, P)$ takes

image I and pose P as input, and $G_T(S, P)$ takes text S and pose P as input.

3.2.1 Image-based Generation Module $G_I(I, P)$

This module aims to generate image O_I with feature of image I and pose input P. O_I has structure guided by pose P and appearance as I . As shown in Fig. 2, to output O_I, we obtain the image feature F_I and the spatial feature F_S from I and P respectively. Then we use F_I to compute the AdaIN parameter $A_I = \mathcal{M}(F_I)$ to generate $O_I = D(F_S, A_I)$, where D is the AdaIN decoder in Eq. (1).

The generator in this module consists of the image encoder E_I, the spatial encoder E_S, the network to compute AdaIN paramter $\mathcal{M}$ and the decoder D.

We use training pair as (I, P) where P is the pose of I, and set reconstruction loss as the distance between O_I and I in both pixe- and feature-level as

$$\mathcal{L}_{\mathcal{I}_{rec}} = \sum_{i=0}^{5} \mathbb{E}(||\Phi_i(O_I) - \Phi_i(I)||). \tag{4}$$

Meanwhile, we use GAN loss [8] to enhance the quality of generation, by setting a discriminator $\mathcal{D}$ whose input is the concatenation of real/fake images and pose input. The loss for the generator and the discriminator is designed as LSGAN [51] as

$$\begin{aligned} \mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{R}}} =& \mathbb{E}_{I\in p_\mathcal{R}}((\mathcal{D}(I \oplus P) - 1)^2), \\ \mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{I}}} =& \mathbb{E}_{I\in p_\mathcal{R}}((\mathcal{D}(G_I(I, P) \oplus P) - 0)^2), \\ \mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_\mathcal{I}}} =& \mathbb{E}_{I\in p_\mathcal{R}}((\mathcal{D}(G_I(I, P) \oplus P) - 1)^2), \end{aligned} \tag{5}$$

where $I \in p_R$ is the distribution of real images, P is the pose of image I. Loss term $\mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_*}}$ is for the discriminator while $\mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_*}}$ is for the generator. We adopt the feature match loss [52], [53], [54], [55] as an auxiliary part of GAN loss. Specifically, we obtain features from $\mathcal{D}$ for fake and real images, and compute their distance as

$$\mathcal{L}_{F_\mathcal{I}} = \mathbb{E}(\|\mathcal{F}(O_I) - \mathcal{F}(I)\|), \tag{6}$$

where $\mathcal{F}(X)$ is the feature obtained from the layer before final output in the discriminator for real/fake image X.

3.2.2 Text-based Generation Module $G_\mathcal{T}(S, P)$

This module generates image O_T based on the sentence input S and the pose input P. The output of this module is utilized to provide the pseudo ground truth for the training of the manipulation module. Therefore, we ensure that O_T has the required pose as P by adopting the AdaIN structure for this module.

We first obtain text feature F_T from text encoder, and then compute AdaIN parameters as $A_T = \mathcal{M}(F_T)$. Combined with spatial feature F_S, O_T is obtained as $O_T = D(F_S, A_T)$. It is noted that the AdaIN decoder here is the same as that in the module $G_I(I, P)$. Parameters are shared so that the feature derived from E_I and E_T share the same space for final synthesis.

Therefore, the generator in this module consists of E_T, E_S, $\mathcal{M}$ and D.

Besides, the same loss from Eqs. (4)-(6) is adopted in this module and the reconstruction loss is written as

$$\mathcal{L}_{\mathcal{T}_{rec}} = \sum_{i=0}^{5} \mathbb{E}(||\Phi_i(O_T) - \Phi_i(I)||). \tag{7}$$

Meanwhile, similar to Eqs. (5) and (6), we use the GAN loss of

$$\begin{aligned} \mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{T}}} =& \mathbb{E}_{S\in p_T}((\mathcal{D}(G_\mathcal{T}(S, P) \oplus P) - 0)^2), \\ \mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_\mathcal{T}}} =& \mathbb{E}_{S\in p_T}((\mathcal{D}(G_\mathcal{T}(S, P) \oplus P) - 1)^2), \end{aligned} \tag{8}$$

$$\mathcal{L}_{F_\mathcal{T}} = \mathbb{E}(\|\mathcal{F}(O_T) - \mathcal{F}(I)\|), \tag{9}$$

where S is the consistent text input for (I, P) pair in Eq. (5), which contains sufficient description for the whole content of I. $S \in p_T$ is its distribution. Other notations are the same as Eqs. (5) and (6).

Besides, to let the text and image feature, which contain the same information, approach each other in the shared feature space, we set text-and-image similarity loss proposed in [31]. This loss is computed from word- and sentence-level, which are denoted as $\mathcal{L}_{W_{sim}}$ and $\mathcal{L}_{S_{sim}}$ respectively.

3.3 Overall Loss Term

In summary of our method, the overall loss terms for the image encoder E_I, text encoder E_T, spatial encoder E_S, network to compute AdaIN paramter $\mathcal{M}$, decoder D, attribute-vector generator $\mathcal{A}$, and discriminator $\mathcal{D}$ are defined as

$$\begin{aligned} \mathcal{L}_{E_I,E_T} &= \mathcal{L}_{Rec} + \mathcal{L}_{Aug} + \mathcal{L}_{Adv} + \mathcal{L}_{Mat} + \mathcal{L}_{Sim}, \\ \mathcal{L}_{E_S,\mathcal{M},D} &= \mathcal{L}_{Rec} + \mathcal{L}_{Aug} + \mathcal{L}_{Adv} + \mathcal{L}_{Mat}, \\ \mathcal{L}_\mathcal{D} &= \mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{R}}} + \mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{T}}} + \mathcal{L}_{\mathcal{GAN}_{\mathcal{D}_\mathcal{I}}}, \\ \mathcal{L}_\mathcal{A} &= \mathcal{L}_{Aug}, \end{aligned} \tag{10}$$

where

$$\begin{aligned} \mathcal{L}_{Aug} =& \lambda_1(\mathcal{L}_{\mathcal{M}_{pos}} + \mathcal{L}_{\mathcal{M}_{neg}}), \\ \mathcal{L}_{Rec} =& \lambda_2\mathcal{L}_{\mathcal{I}_{rec}} + \lambda_3\mathcal{L}_{\mathcal{T}_{rec}}, \\ \mathcal{L}_{Adv} =& \lambda_4(\mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_\mathcal{I}}} + \mathcal{L}_{\mathcal{GAN}_{\mathcal{G}_\mathcal{T}}}), \\ \mathcal{L}_{Mat} =& \lambda_5(\mathcal{L}_{F_\mathcal{I}} + \mathcal{L}_{F_\mathcal{T}}), \\ \mathcal{L}_{Sim} =& \lambda_6(\mathcal{L}_{W_{sim}} + \mathcal{L}_{S_{sim}}), \end{aligned} \tag{11}$$

and λ_1-λ_6 refer to loss weights. All loss terms, except for $\mathcal{L}_{\mathcal{M}_{pos}}$ and $\mathcal{L}_{\mathcal{M}_{neg}}$, are computed on training tuple (S, I, P) where the content of text S is complete and positive with regard to the image I, and P is the pose of I.

3.4 Network Details

In experiments, the network configuration for each component in our framework is summarized as the following. 1) The image encoder E_I consists of several convolution layers and one global average pooling layer, as shown in Fig. 5(a). Here, "$Conv, 7 \times 7, 3 \times 64, 1, ReLU$" means that this convolution layer adopts kernel size of 7×7 with stride size of 1, and has 3 input feature channels and 64 output feature channels. An activation function ReLU is applied to the output of this convolution layer. Meaning of other convolution layers can be interpreted in the same way.

2) Spatial encoder E_S consists of four convolution layers, and two convolutional residual blocks [56] with instance normalization (IN), as shown in Fig. 5(b). 3) The AdaIN-based decoder D consists of two convolutional residual blocks with AdaIN as the normalization operation, and several convolution layers with AdaIN, as shown in Fig. 5(c). 4) The network to compute AdaIN parameters $\mathcal{M}$, and the attribute vector generator $\mathcal{A}$ are both implemented as MLP, as shown in Fig. 5(d)&(e) respectively, where "$FC, \mathcal{X} \times \mathcal{Y}$" means the fully connected layer with $\mathcal{X}$ input feature channels and $\mathcal{Y}$ output channels.

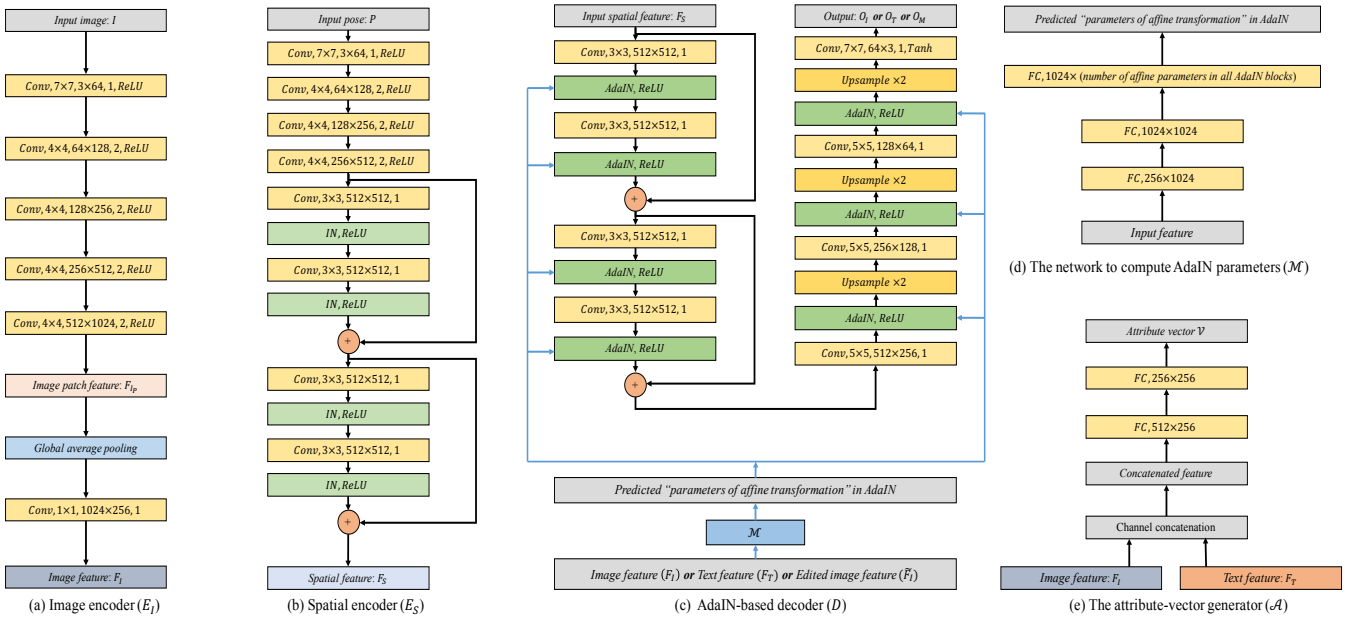


Fig. 5. Detailed structure of the image encoder E_I, spatial encoder E_P, AdaIN-based decoder D, network to compute AdaIN parameters $\mathcal{M}$, and attribute vector generator $\mathcal{A}$.

The text encoder E_T adopts structure as bidirectional LSTM [46], and it has two types of output. Given text input S, we first obtain the feature of each word in S, which is the recurrent output of LSTM. Such feature is denoted as F_{T_W} with shape $C \times L$, where L is the number of words in S and C is the number of feature channel. F_{T_W} can also be denoted as $\{f_{w1}, ..., f_{wL}\}$, where f_{wi} is the feature of i-th word with shape as $C \times 1$. Suppose the final hidden state of LSTM after process each word is h with shape $C \times 1$ (which is a global and coarse representation of S). We compute the cosine similarity between h and f_{w_i} as $R_i = (f_{wi}^T h)/(\|f_{wi}\|\|h\|)$. Inspired by the sequence-to-sequence NLP models [57], [58], we fuse h and $\{f_{w1}, ..., f_{wL}\}$ to obtain the final global representation of S with attention mechanism of

$$F_T = \mathcal{K}(f_w \oplus h), f_w = \sum_{i=1,...,L} (f_{w_i} \times R_i), \tag{12}$$

where F_T is the text feature in Fig. 2, and $\mathcal{K}$ is one-layer MLP with input size 512 and output size 256.

Moreover, the image encoder E_I also produces two types of features. The first one is the image patch feature F_{I_P} as shown in Fig. 5(a), whose shape is $H \times W \times C$. H and W are height and width of the feature; C is the number of feature channel. The second one is the feature F_I in Fig. 2, which is obtained as the final output of E_I, the global representation of the input image. F_I and F_T have the same shape. Besides, $\mathcal{L}_{W_{sim}}$ is computed between F_{I_P} and F_{T_W}, and $\mathcal{L}_{S_{sim}}$ is computed between F_I and F_T [31].

Additionally, for each dataset, we split its training data into n categories based on IDs of persons in the corresponding dataset, and the discriminator $\mathcal{D}$ has the number of output channels as n. Each channel responses to adversarial learning of one category.

TABLE 1
Loss weights for our framework on different datasets.

	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
PEDES [15]	30	100	100	1	50	1
CelebA [16]	2	20	5	1	10	1

Algorithm 1 Training Procedure

Parameter: Consistent training tuple (S, I, P) where the text S correctly describes all concerned attributes in image I , and pose input P is the pose information for I.

1: **while** not converged **do**
2: Compute $\mathcal{L}_{Rec}$, $\mathcal{L}_{Adv}$ and $\mathcal{L}_{Mat}$ based on (S, I, P), and update E_I, E_T, E_S, D, $\mathcal{M}$ based on them.
3: Compute $\mathcal{L}_{Sim}$ and update E_I and E_T.
4: Apply text augmentation (as shown in Fig. 4 of our manuscript) on (S, I, P) to obtain incomplete text, as well as negative text. Then compute $\mathcal{L}_{Aug}$ based on them, and update E_I, E_T, E_S, D, $\mathcal{M}$, $\mathcal{A}$.
5: Compute $L_{\mathcal{D}}$ and update $\mathcal{D}$.
6: **end while**

3.5 Training Details

The details of loss weights for different datasets are listed in Table 1. We use Adam optimizer [59] to train our framework, with β_1 and β_2 set as 0.5 and 0.999 respectively. The learning rate is set as 10^{-4}. We summarize the training detail of our framework in Algorithm 1. It is trained on an Intel 2.60GHz CPU and one TITAN X GPU.

4 Experiments

4.1 Datasets

In our experiments, CUHK-PEDES [15] and CelebA [16] are utilized to validate the effectiveness of our apprach.

TABLE 2
VQA-score (VQAs), reconstruction $L1$ error and FID of different settings in the ablation study.

Method	CUHK-PEDES			CelebA		
	VQAs	L_1	FID	VQAs	L_1	FID
$w/o(Pos)$	0.608	0.105	81.95	0.618	0.112	19.68
$w/o(Neg)$	0.094	0.099	60.39	0.138	0.089	22.20
$w/o(Pix_{\mathcal{I}})$	0.590	0.104	78.68	0.702	0.117	17.64
$w/o(Per_{\mathcal{I}})$	0.596	0.106	80.57	0.742	0.093	17.93
$w/o(Adv_{\mathcal{I}})$	0.592	0.098	82.00	0.704	0.103	18.61
$w/o(Pix_{\mathcal{T}})$	0.520	0.102	77.94	0.592	0.091	17.06
$w/o(Per_{\mathcal{T}})$	0.648	0.105	76.70	0.692	0.097	18.02
$w/o(Adv_{\mathcal{T}})$	0.618	0.100	81.99	0.740	0.098	34.35
$w/o(Sim)$	0.550	0.103	80.63	0.678	0.100	20.77
$w/o(Pix)$	0.616	0.137	63.00	0.688	0.116	21.26
$w/o(Per)$	0.606	0.115	80.15	0.592	0.097	18.66
$w/o(Adv)$	0.624	0.097	80.42	0.740	0.092	31.76
$with(GAN)$	0.640	0.104	60.23	0.762	0.096	17.24
Ours	**0.668**	**0.092**	**55.18**	**0.792**	**0.083**	**15.64**

CUHK-PEDES CUHK-PEDES dataset [15] is a caption-annotated pedestrian image dataset. This dataset contains 40,206 images of 13,003 persons collected from five person re-identification datasets, which include CUHK03 [60], Market-1501 [61], SSM [62], VIPER [63], and CUHK01 [64]. Especially, we use 80% of images in this dataset for training and 20% for testing. The size of input image in this dataset is 128×64 for experiments.

Moreover, each image in CUHK-PEDES is annotated with descriptions by crowd-sourcing. Given a set of pre-defined semantic attributes, the workers on Amazon Mechanical Turk (AMT) write sentences to describe the attributes in this defined set for each image. The sentence pattern is "subject-predicate-(adjective)-object", where the "object" is obtained from the name space of attributes. Besides, all sentences are consistent with the corresponding images.

CelebA CelebA dataset [16] contains 202,599 face images, each annotated with 40 attributes. To construct the textual description for each image, we first select a set of attributes and construct a positive and complete text with the basic sentence pattern, i.e., "subject-predicate-(adjective)-object". Especially, we use the attribute values for each image to provide the adjectives. In this way, we provide the complete text description for each image. We adopt the train-test split provided in this dataset. The size of input image in this dataset is 128×128 for experiments.

4.2 Evaluation Metrics

To illustrate the effect of various manipulation approaches, we adopt three quantitative metrics for evaluation.

- **VQA score** We follow [5] to use VQA score to measure the accuracy of manipulation. Given a manipulated image, a question is designed according to the attribute to manipulate. It is then passed to a trained VQA model [65]. The VQA score is defined by the accuracy that the predicted answer confirms the manipulation target. A higher VQA score means the model performs manipulation more correctly, which is better.
- **FID Score** To quantify the level of realism of manipulated images, we adopt Frechet Inception Distance (FID) [66], which measures the distance of the distribution between manipulated and real images. Lower FID scores indicate better quality of generated images.
- **Reconstruction Error** We also follow [1] to compute L_1 reconstruction error by editing images with positive text inputs, whose descriptions are consistent with images, to measure the ability of preserving irrelevant attributes during editing. When the content of text is consistent with the input image, the synthesis result should be identical with the input. Thus following [1], we use L_1 to quantify the reconstruction error. Lower L_1 loss is better.

4.3 Ablation study

In this section, we perform ablation study to show the effectiveness of each loss term in our framework.

4.3.1 *Loss for module* $G_{\mathcal{M}}(S, I, P)$

$\mathcal{L}_{\mathcal{M}_{pos}}$ and $\mathcal{L}_{\mathcal{M}_{neg}}$ play important roles in the training of the manipulation module. Results without them are $w/o(Pos)$ and $w/o(Neg)$. They are listed in Rows 1 and 2 of Table 2.

It shows that removing any of them causes decreasing VQA scores and lowering manipulation accuracy. Especially, under the setting of "$w/o(Neg)$", there are no negative text inputs during the training, while text inputs could be negative during the testing. Without training with negative texts, the model could totally fail on generating new appearances. Thus, the VQA score will decrease a lot. Besides, deleting $\mathcal{L}_{\mathcal{M}_{pos}}$ or $\mathcal{L}_{\mathcal{M}_{neg}}$ also leads to the increasement of FID and L_1 error, indicating degradation of image quality. These results demonstrate the effectiveness of $\mathcal{L}_{\mathcal{M}_{pos}}$ and $\mathcal{L}_{\mathcal{M}_{neg}}$.

4.3.2 *Loss for module* $G_I(I, P)$

The loss to train this module includes the reconstruction loss in pixel-level, feature-level and the GAN loss, which are defined in Eqs. (4), (5) and (6). The results without them are called $w/o(Pix_{\mathcal{I}})$, $w/o(Per_{\mathcal{I}})$, and $w/o(Adv_{\mathcal{I}})$ respectively. They are listed from Rows 3 to 5 in Table 2. It is clear that removing any of these loss terms causes obvious decrease on VQA score and increase on FID and L_1 error.

When deleting the GAN loss from the image-based generation module G_I, the text-based generation module $G_{\mathcal{T}}$ can still produce pseudo label. However, due to the lack of the GAN loss, the quality of synthesized images will be decreased. This results in the decrease of VQA score and the increase of FID and L_1 error.

4.3.3 *Loss for module* $G_{\mathcal{T}}(S, P)$

The loss to train this module also includes the reconstruction loss in pixel-level, feature-level and the GAN loss, which are defined in Eqs. (7), (8) and (9). The results without them are recorded from Rows 6 to 8 in Table 2, denoted as $w/o(Pix_{\mathcal{T}})$, $w/o(Per_{\mathcal{T}})$, and $w/o(Adv_{\mathcal{T}})$ separately.

As shown, deleting any of these loss terms causes the deterioration of the manipulation effect. Especially, under the setting of $w/o(Adv_{\mathcal{T}})$, the FID increases significantly,

TABLE 3
Comparison with existing methods (with pose input) on the same testing set.

Methods	CUHK-PEDES			CelebA		
	VQAs	L_1	FID	VQAs	L_1	FID
SIS [2]	0.280	0.297	62.90	0.570	0.289	39.82
TAGAN [1]	0.518	0.136	80.23	0.622	0.150	23.88
MainGAN [3]	0.604	0.154	66.34	0.724	0.136	26.16
Prada [4]	0.446	0.273	91.36	–	–	–
Ours	**0.668**	**0.092**	**55.18**	**0.792**	**0.083**	**15.64**

TABLE 4
Comparison with existing methods (with their original configurations) on the same testing set.

Methods	CUHK-PEDES			CelebA		
	VQAs	L_1	FID	VQAs	L_1	FID
SIS [2]	0.326	0.254	66.32	0.598	0.257	36.32
TAGAN [1]	0.538	0.154	76.30	0.654	0.168	21.64
MainGAN [3]	0.622	0.128	61.34	0.752	0.153	30.48
Prada [4]	0.475	0.241	86.74	–	–	–
Ours	**0.668**	**0.092**	**55.18**	**0.792**	**0.083**	**15.64**

TABLE 5
Human evaluation for "quality and accuracy" and "usability", between our framework and all baselines.

Methods	Quality and Accuracy		Usability	
	PEDES	CelebA	PEDES	CelebA
SIS [2]	0.017	0.018	0.000	0.033
TAGAN [1]	0.023	0.030	0.033	0.067
MainGAN [3]	0.067	0.087	0.033	0.033
Prada [4]	0.013	–	0.000	–
Ours	**0.880**	**0.865**	**0.933**	**0.867**

while the VQA score decreases slightly. This indicates that the GAN loss in the training of $G_{\mathcal{T}}$ is critical for the image quality, but does not play the key role to guide the generation from text to image. As the text description and images are paired in our training set, the consistency between text and generated images are ensured with $\mathcal{L}_{\mathcal{T}_{rec}}$ (Eq. (7)) instead of the GAN loss in the training of $G_{\mathcal{T}}$.

Meanwhile, removing $\mathcal{L}_{Sim}$ reduces the performance as shown in Row 9 of Table 2. This is because this loss enhances the relation between the feature of image and text in the shared space, which covers identical information.

4.3.4 More Ablation Settings

$G_I(I, P)$ and $G_{\mathcal{T}}(S, P)$ learn image-text shared space to help training of $G_{\mathcal{M}}(S, I, P)$. They adopt the same form of loss. We analyze the effect if one class of loss term is deleted from both modules simultaneously. We delete the reconstruction loss with pixel level in $\mathcal{L}_{\mathcal{I}_{rec}}$ and $\mathcal{L}_{\mathcal{T}_{rec}}$, and list the results in Row 10 of Table 2. We also delete the reconstruction loss with feature level in $\mathcal{L}_{\mathcal{I}_{rec}}$ and $\mathcal{L}_{\mathcal{T}_{rec}}$ and show results in Row 11 of Table 2. The results of deleting GAN loss (including $\mathcal{L}_{Adv}$ and $\mathcal{L}_{\mathcal{D}}$) are listed in Row 12 of Table 2. It is clearly that all these three ablation settings cause VQA score dropping and increase of L_1 error as well as FID. This phenomenon proves the usefulness of our loss terms again.

In addition, we also analyze the effect after adding GAN loss in the training of $G_{\mathcal{M}}(S, I, P)$. We add the GAN loss with the following setting: in each iteration, for an input image I and a text input, we use the modified image from $G_{\mathcal{M}}$ as the fake sample, and set the input image I as the real sample. The results are listed in Row 13 of Table 2 marked as "with (GAN)" and demonstrate this ablation setting also causes deterioration of performance. This is because such GAN loss may impede the modification of the content, since the generator will try to make the content of the fake sample be the same as the real sample, causing the decrease of the VQA score (i.e., for a negative text input, the modification is more likely to fail compared with the setting of "Ours"). As for the positive input, GAN encourages the model to generate textures that make the manipulated outputs look sharper. However, the generated textures may not be exact the same as the ground truth, which leads to the increase of L_1 error.

4.4 Comparison with Related Methods

We compare our framework with state-of-the-art text-guided image manipulation approaches. We choose methods that take text input and have official codes for fairness in comparison. TAGAN [1], Semantic Image Synthesis (SIS) [2], MainGAN [3] and Prada [4] are four state-of-the-art text-guided manipulation methods. We take pose input to their manipulation modules by sending concatenation of image and pose input to their convolutional encoders of images. We only compare with Prada [4] on CUHK-PEDES, since it is not applicable to the manipulation of facial images.

The quantitative results are recorded in Table 3. Our method achieves the highest VQA score, and the lowest L_1 loss. This suggests that our framework manipulates content of images accurately while preserving original image information that should not be manipulated. Besides, our method yields the lowest FID, since the manipulated images from our method are natural. Moreover, we compare against MainGAN, TAGAN, SIS and Prada with their original configuration (i.e., without pose input for their convolutional encoders of images). The results are shown in Table 4.

4.5 Visual Comparison

Visual examples are in Figs. 6 and 7 for comparison between our framework and the baselines (without pose input). The text input is listed on the top of each target image.

For example, for the target image in the first row of Fig. 6, "This man has (black $\rightarrow$ gray) pants" means that black pants turn to gray. As shown in Fig. 6, our method accomplishes higher manipulation accuracy and quality compared with these baselines, on CUHK-PEDES dataset. TAGAN performs better than SIS, while it still changes the irrelevant content during manipulation. Moreover, although Prada can achieve editing effect, its quality and accuracy is lower. MainGAN produces better results than TAGAN and SIS. But the result quality still has room to improve. It tends to modify content that is not mentioned in the text, and possibly fails to manipulate the target. Fig. 7 also shows face result comparison.

4.6 Human Evaluation

We conduct user study for results on both CUHK-PEDES and CelebA. We show each participant the input image

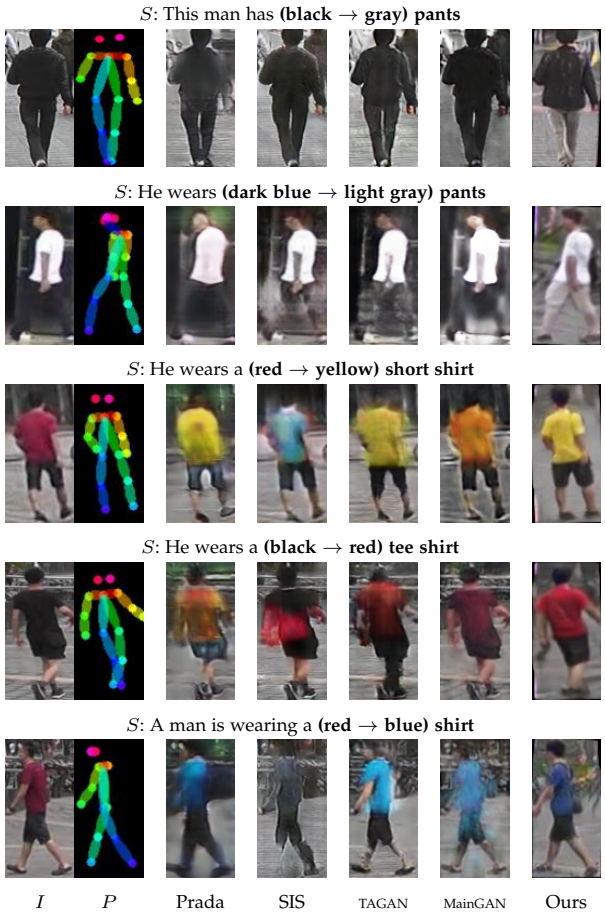


Fig. 6. Visual comparison with baselines on CUHK-PEDES dataset.

TABLE 6
Comparison with all baselines for the task that edits both appearance and spatial structure on CUHK-PEDES dataset.

Method	VQAs	FID
SIS [2]	0.302	75.24
TAGAN [1]	0.492	84.76
MainGAN [3]	0.575	68.32
Prada [4]	0.413	104.64
Ours	**0.624**	**60.22**

and text, as well as the manipulation results from different methods. Then we ask him/her to choose the one with the highest quality and accuracy. For each participant, we raise 20 questions, and the number of participants is 30. The results in Table 5 show that 88.0% and 86.5% of all participants choose our results on the datasets of CUHK-PEDES and CelebA respectively.

In another user study, we show each participant the manipulation process and function of different methods, since our framework obtains sequential results and control pose simultaneously. We ask them to choose the best one that satisfies their preference. For all 30 participants, 93.3% and 91.5% of them agree that the usability of our framework is the best on CUHK-PEDES and CelebA respectively as shown in Table 5.

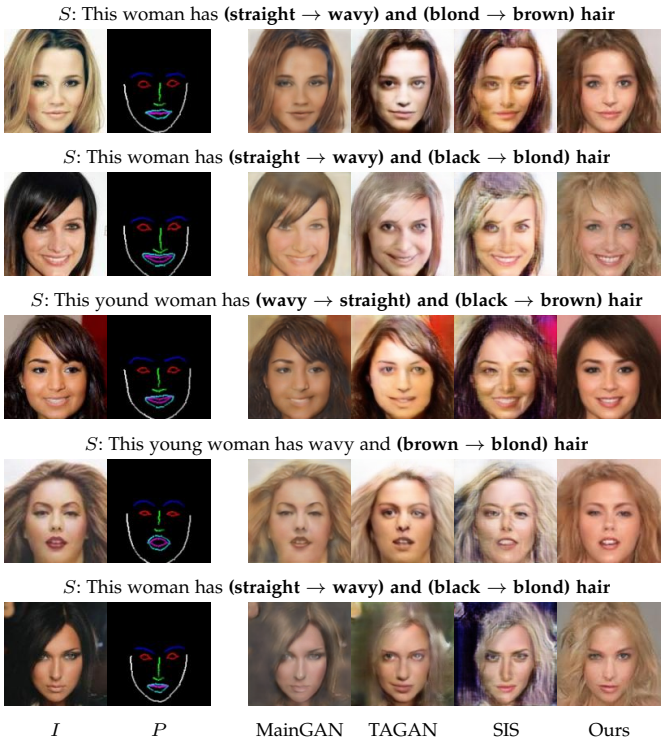


Fig. 7. Visual comparison with baselines on CelebA dataset.

TABLE 7
Experiments to evaluate the ability of identity preservation on CelebA dataset.

Method	Proportion of Identity Preservation
SIS [2]	0.447
TAGAN [1]	0.585
MainGAN [3]	0.762
Ours	**0.804**

4.7 Editing Both Appearance and Spatial Structure

Different from existing approaches that only focus on editing appearance or spatial structure, our framework manipulates both simultaneously. One may argue that this effect can be achieved by applying spatial editing after appearance manipulation. In this section, we show that this two-step strategy is not optimal.

First, MainGAN, TAGAN, SIS and Prada all implement editing of appearance. We finetune the pre-trained model of [38] on the correspondingly edited samples, and apply the finetuning model to spatial editing. For fairness, the target poses are the same for all methods during spatial editing. The results are shown in Table 6. Our method yields the highest VQA score and lowest FID.

4.8 Ability of Identity Preservation

Since the face data has identity information, we need to consider the ability of identity preservation during manipulation. To verify the ability of identity preservation in our framework, we set an experiment.

First, we train a network to classify the identity of different faces in CelebA dataset. The network structure takes ResNet101 [56] as backbone. It yields accuracy of 91.05%. Then, we use this trained classifier to compute the identity

TABLE 8
Performance on the CUHK-PEDES dataset, with different forms of structural input.

Method	VQAs	L_1	FID
Keypoints	0.612	0.115	58.63
Skeletons	**0.668**	**0.092**	**55.18**

TABLE 9
The comparison with existing methods (with pose input) on the same testing set, under the evaluation setting with the perturbations in the pose inputs.

Methods	CUHK-PEDES			CelebA		
	VQAs	L_1	FID	VQAs	L_1	FID
SIS [2]	0.268	0.306	66.13	0.558	0.291	42.35
TAGAN [1]	0.503	0.151	82.42	0.611	0.173	25.94
MainGAN [3]	0.582	0.178	68.89	0.705	0.162	30.01
Prada [4]	0.451	0.291	96.48	–	–	–
Ours	**0.655**	**0.113**	**57.32**	**0.778**	**0.105**	**18.25**

of testing images, by applying image manipulation methods (includes TAGAN [1], SIS [2], MainGAN [3] and ours). We report the proportion of images, whose classification results are identical, before and after editing. The results are shown in Table 7. Our framework effectively preserves identity information during image manipulation.

To explain it, in our framework, although we drop the spatial information of input images, pose input can adequately replenish the spatial information. Also, features of appearance from input images ensure manipulation output due to unchanged appearance, except for the portion to edit. Thus, the identity information is preserved in our AdaIN-based framework, as suggested by [67].

4.9 Robustness of Structural Input

In this section, we first prove the robustness of our framework when changing the form of structural input. Especially, we use keypoints to replace skeletons for experiments on CUHK-PEDES, and the results are listed in Table 8. VQAs, L_1 loss and FID are comparable, when using skeletons as structural input. Thus, using keypoints in our framework performs comparably.

To further study our framework's performance with more inaccurate structural input, we conduct an additional experiment as follows. We use the same models and evaluation protocols as in Table 3, while we add perturbations to the coordinates of the key-points in the pose during evaluation. For an image with a size of 128×64 and 128×128, the perturbation size is set as $[0, 12]$. As shown in Table 9, our method still gives superior results over baselines with the perturbations in the pose. This indicates that in most cases, the automatically predicted structural input is sufficiently good for our model.

4.10 Interpolation between Image and Text Features

In this section, we show linear interpolation between image and text features in our learned shared space. Existing manipulation methods perform feature interpolation for generation [1], [7] on the same types of features. Our framework, contrarily, can perform interpolation between image and text features. To verify it, we denote the image feature as F_I and the text feature as F_T as shown in Fig. 2. The interpolated feature becomes $F = F_I \times \alpha + F_T \times (1 - \alpha)$. We visualize these features through the decoder D. If the image and text features are mapped into the same space, these interpolated features should produce images, whose distribution is the same as the output when we use only image or text feature for the decoder D.

Results of interpolation on CUHK-PEDES and CelebA datasets are shown in Figs. 8 and 9. Interpolated features produce reasonable intermediates. Moreover, the results when $\alpha = 0$ illustrate that $G_{\mathcal{T}}(S, P)$ can produce required pseudo labels to guide formulation of $\mathcal{L}_{\mathcal{M}_{neg}}$.

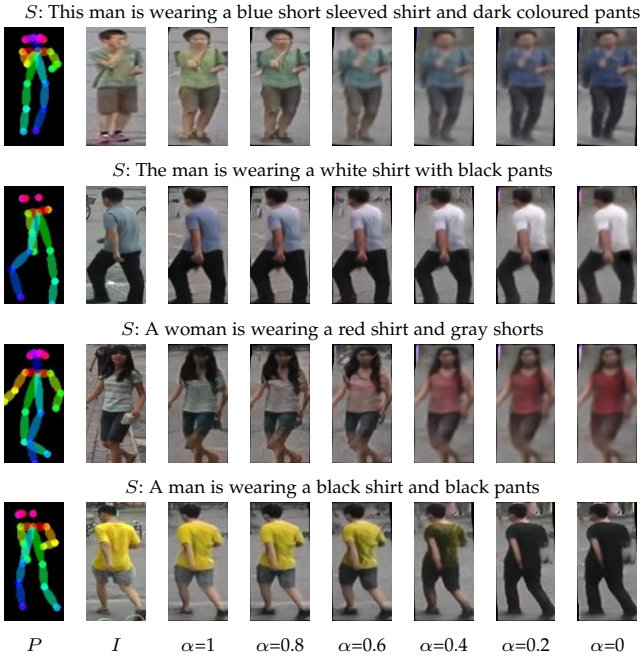


Fig. 8. Interpolation results on CUHK-PEDES dataset, which indicate that image and text features are mapped in the same space. When $\alpha = 1$, only image feature is used; when $\alpha = 0$, only text feature is used.

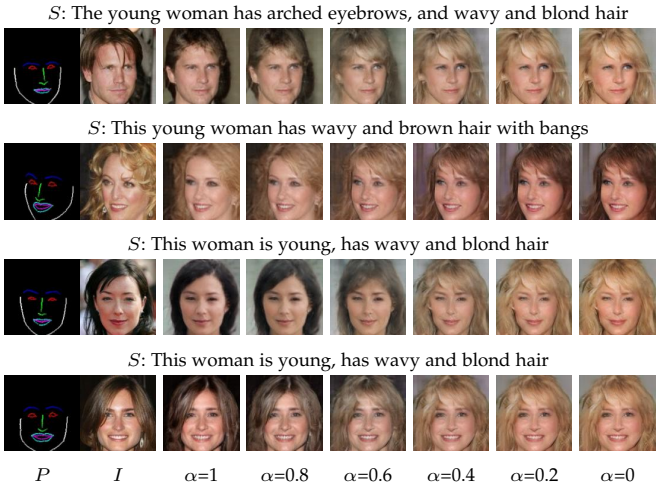


Fig. 9. Interpolation results on CelebA dataset, which indicate that image and text features are mapped in the same space.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3085339, IEEE Transactions on Pattern Analysis and Machine Intelligence

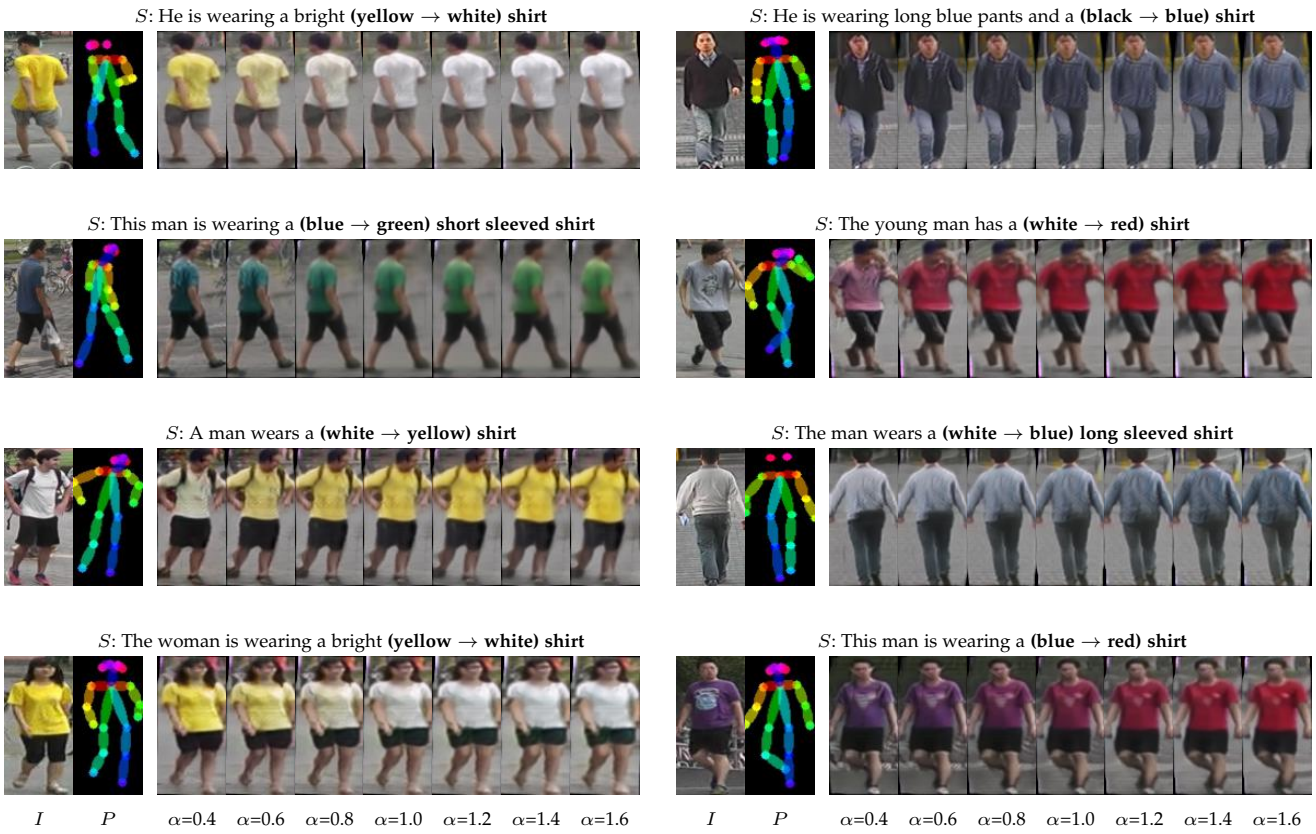


Fig. 10. Visual illustration on CUHK-PEDES dataset. Our method not only manipulates the image with text input, but also controls manipulation strength.

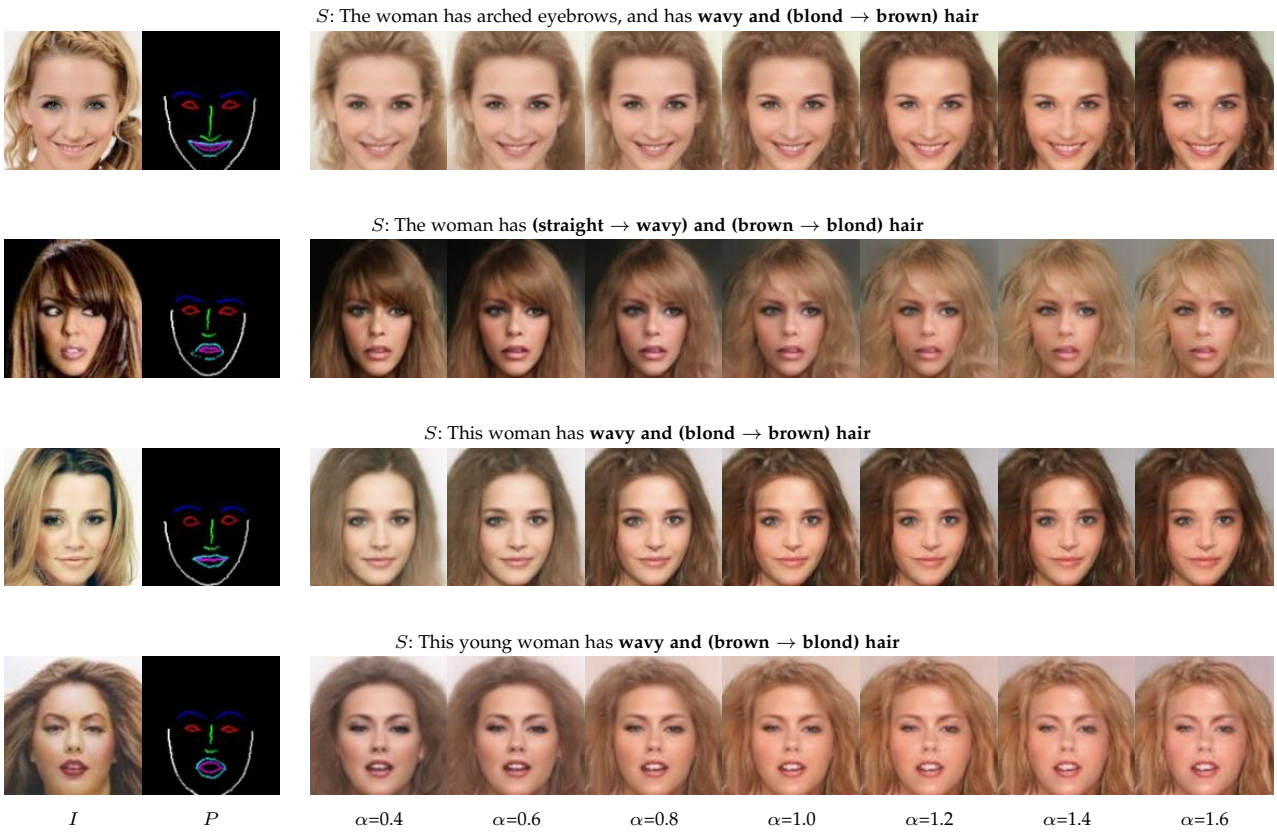


Fig. 11. Visual illustration on CelebA dataset. Our method not only manipulates the image with text input, but also controls manipulation strength.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3085339, IEEE Transactions on Pattern Analysis and Machine Intelligence

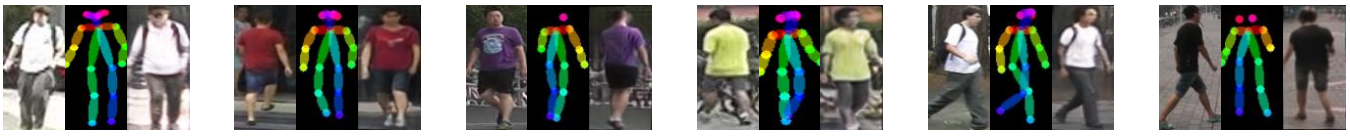


Fig. 12. Visual examples of pose control on CUHK-PEDES.

Fig. 13. Visual examples of pose control on CelebA dataset.

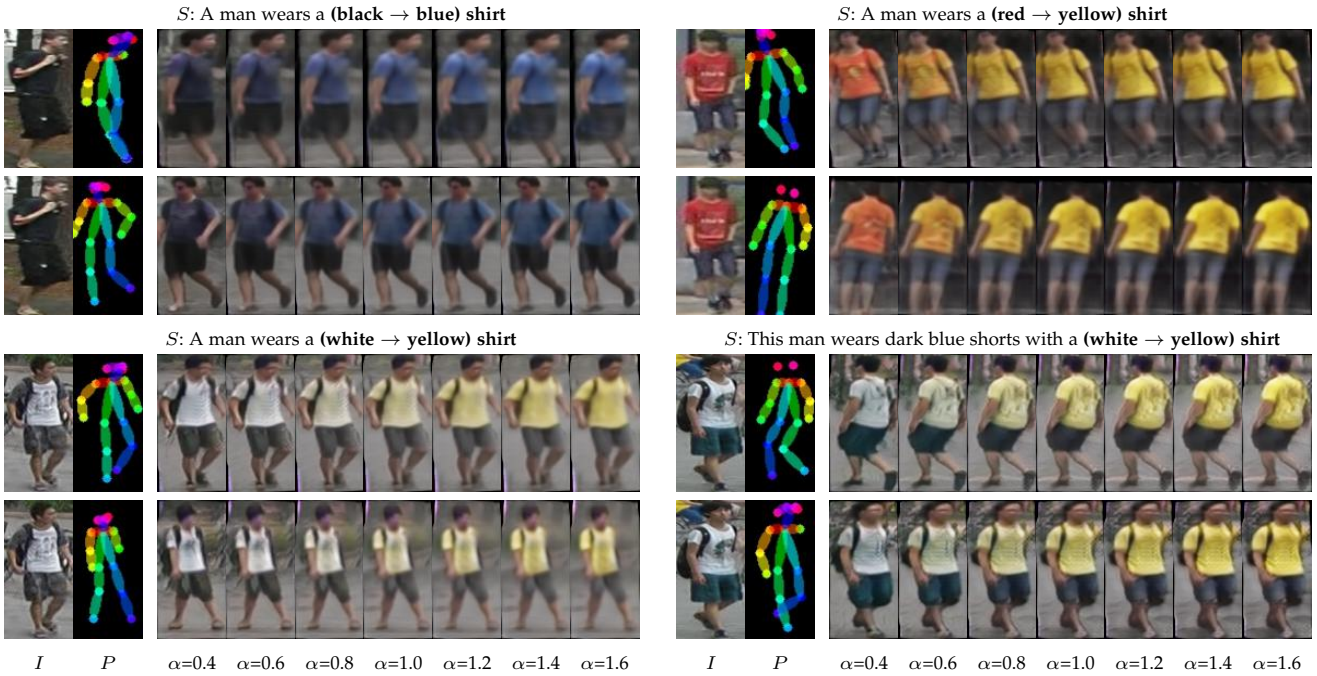


Fig. 14. Visual illustration on CUHK-PEDES dataset. Our method implements pose control, and yields controllable manipulation strength.

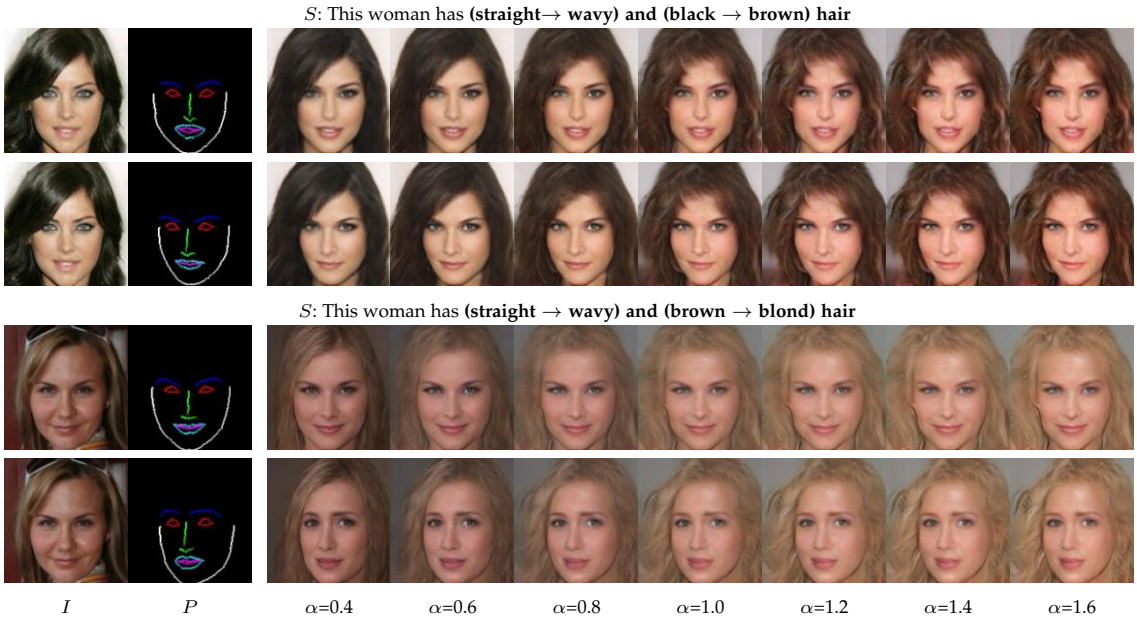


Fig. 15. Visual illustration on CelebA dataset. Our method implements pose control, and yields controllable manipulation strength.

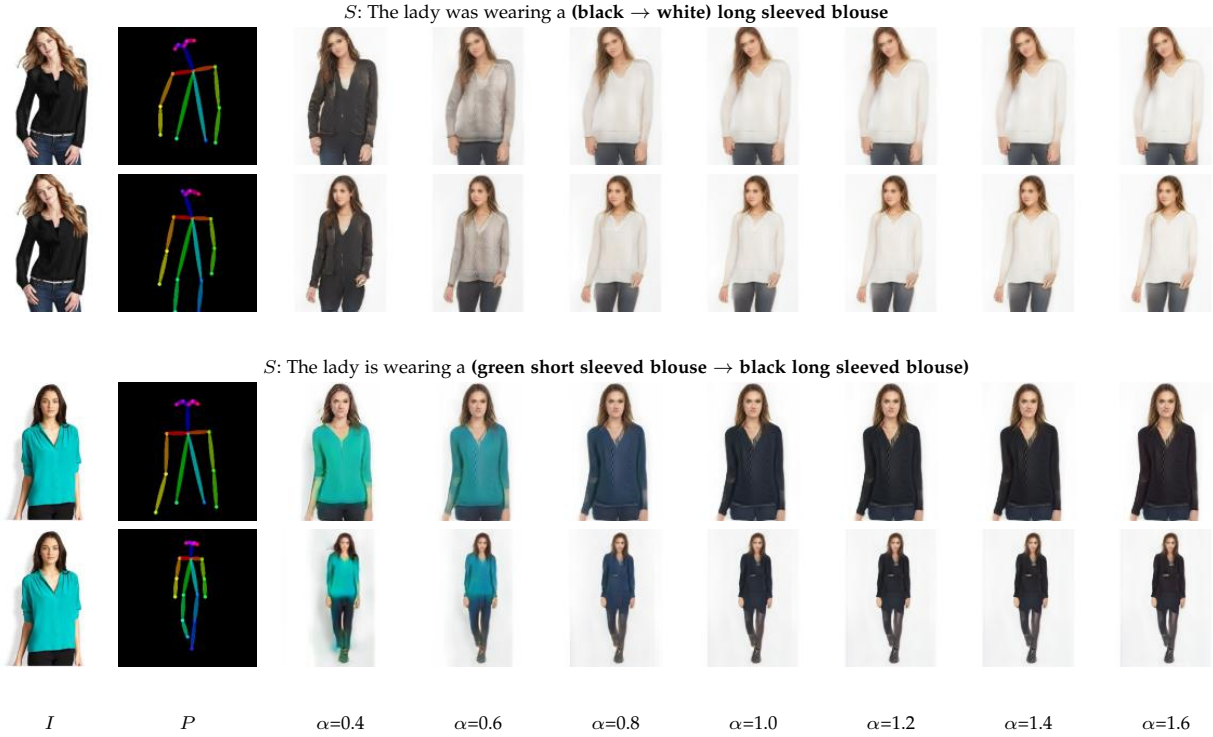


Fig. 16. Visual illustration on Deepfashion dataset.

4.11 Interactive Manipulation

Our framework further allows interactive manipulation. Users can control pose of output interactively, and tune edit strength. Figs. 10 and 11 show the results of edit strength control. As our framework implements manipulation by moving image feature along the direction of attribute vector, we control the edit strength by using different step sizes α. As shown in the bottom row of Fig. 10, by varying α from 0.6 to 1.0, the color of shirt changes from "light red" to "bright red". In the second row of Fig. 11, α from 0.8 to 1.4 alters hair from "unconspicuous blond" to "conspicuous blond".

Figs. 12 and 13 present illustration of pose control. Note that pose extraction is an independent module. Users can adjust joints or landmarks of the extracted pose to obtain desired pose, before passing temporary results to our framework. Using pose input that is not consistent with the spatial structure of the images to manipulate, we get corresponding results as shown in Figs. 12 and 13.

Figs. 14 and 15 indicate that the pose control, by using different pose input for identical image and text, can be realized along with edit strength control. Moreover, expression editing for faces can be achieved by manipulating the corresponding landmarks, as shown in Figs. 1 and 15.

4.12 Interactive Manipulation on Fashion Images

Interactive human image manipulation can greatly contribute to online fashion [20], [68], [69]. Here, we apply our framework to a fashion dataset to illustrate its generality. We conduct experiments of interactive manipulation on DeepFashion dataset [70] by adopting input pose with form of skeleton. There are annotations of text, which describes the corresponding images. Thus, this dataset meets the requirement for training our framework.

The results are shown in Fig. 16. Our framework simultaneously controls the pose and edit strength during manipulation. It produces modification and good-quality images on this fashion dataset.

5 Conclusion

In this paper, through analyzing the characteristic of textual instructions, we have proposed a novel framework for text-guide human image manipulation. Compared with existing approaches, our framework enables more interactive manipulation, where users can control appearance and spatial editing simultaneously. Moreover, by encoding both text and image features to the shared space, our method achieves accurate manipulation by moving source images along certain attribute vectors in the shared space. Our framework also allows for changing editing strength by generating a series of synthesis results. Extensive experiments on different datasets have manifested the effectiveness and generality of our framework.

References

[1] S. Nam, Y. Kim, and S. J. Kim, "Text-adaptive generative adversarial networks: manipulating images with natural language," in *NIPS*, 2018.
[2] H. Dong, S. Yu, C. Wu, and Y. Guo, "Semantic image synthesis via adversarial learning," in *ICCV*, 2017.
[3] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Manigan: Text-guided image manipulation," in *CVPR*, 2020.
[4] S. Zhu, R. Urtasun, S. Fidler, D. Lin, and C. Change Loy, "Be your own prada: Fashion synthesis with structural coherence," in *ICCV*, 2017.
[5] X. Zhou, S. Huang, B. Li, Y. Li, J. Li, and Z. Zhang, "Text guided person image synthesis," in *CVPR*, 2019.
[6] J. Chen, Y. Shen, J. Gao, J. Liu, and X. Liu, "Language-based image editing with recurrent attentive models," in *CVPR*, 2018.

[7] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," in *CVPR*, 2017.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[9] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *arXiv:1411.2539*, 2014.

[10] I. Laina, C. Rupprecht, and N. Navab, "Towards unsupervised image captioning with shared multimodal embeddings," in *ICCV*, 2019.

[11] X. Hou, L. Shen, K. Sun, and G. Qiu, "Deep feature consistent variational autoencoder," in *WACV*, 2017.

[12] Y.-C. Chen, X. Xu, Z. Tian, and J. Jia, "Homomorphic latent space interpolation for unpaired image-to-image translation," in *CVPR*, 2019.

[13] Y.-C. Chen, H. Lin, M. Shu, R. Li, X. Tao, X. Shen, Y. Ye, and J. Jia, "Facelet-bank for fast portrait manipulation," in *CVPR*, 2018.

[14] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," in *CVPR*, 2020.

[15] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang, "Person search with natural language description," in *CVPR*, 2017.

[16] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.

[17] R. Manuvinakurike, J. Brixey, T. Bui, W. Chang, D. S. Kim, R. Artstein, and K. Georgila, "Edit me: A corpus and a framework for understanding natural language image editing," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018.

[18] R. Manuvinakurike, T. Bui, W. Chang, and K. Georgila, "Conversational image editing: Incremental intent identification in a new dialogue task," in *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 2018.

[19] H. Wang, J. D. Williams, and S. Kang, "Learning to globally edit images with textual description," *arXiv:1810.05786*, 2018.

[20] M. Günel, E. Erdem, and A. Erdem, "Language guided fashion image manipulation with feature-wise transformations," *arXiv:1808.04000*, 2018.

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

[22] X. Mao, Y. Chen, Y. Li, T. Xiong, Y. He, and H. Xue, "Bilinear representation for language-based image editing using conditional generative adversarial networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[23] A. El-Nouby, S. Sharma, H. Schulz, D. Hjelm, L. E. Asri, S. E. Kahou, Y. Bengio, and G. W. Taylor, "Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction," in *ICCV*, 2019.

[24] S. Shinagawa, K. Yoshino, S. Sakti, Y. Suzuki, and S. Nakamura, "Interactive image manipulation with natural language instruction commands," *arXiv:1802.08645*, 2018.

[25] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *CVPR*, 2017.

[26] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv:1605.05396*, 2016.

[27] Z. Zhang, Y. Xie, and L. Yang, "Photographic text-to-image synthesis with a hierarchically-nested adversarial network," in *CVPR*, 2018.

[28] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Mirrorgan: Learning text-to-image generation by redescription," in *CVPR*, 2019.

[29] B. Li, X. Qi, T. Lukasiewicz, and P. H. Torr, "Controllable text-to-image generation," *NIPS*, 2019.

[30] S. Hong, D. Yang, J. Choi, and H. Lee, "Inferring semantic layout for hierarchical text-to-image synthesis," in *CVPR*, 2018.

[31] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He, "Attngan: Fine-grained text to image generation with attentional generative adversarial networks," in *CVPR*, 2018.

[32] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *NIPS*, 2016.

[33] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *CVPR*, 2017.

[34] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *CVPR*, 2020.

[35] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.

[36] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, "Few-shot unsupervised image-to-image translation," *arXiv:1905.01723*, 2019.

[37] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, "Few-shot video-to-video synthesis," *arXiv:1910.12713*, 2019.

[38] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, "Progressive pose attention transfer for person image generation," in *CVPR*, 2019.

[39] Y. Men, Y. Mao, Y. Jiang, W.-Y. Ma, and Z. Lian, "Controllable person image synthesis with attribute-decomposed gan," in *CVPR*, 2020.

[40] Y. Ren, X. Yu, J. Chen, T. H. Li, and G. Li, "Deep image spatial transformation for person image generation," in *CVPR*, 2020.

[41] L. Yang, P. Wang, X. Zhang, S. Wang, Z. Gao, P. Ren, X. Xie, S. Ma, and W. Gao, "Region-adaptive texture enhancement for detailed person image synthesis," in *IEEE International Conference on Multimedia and Expo*, 2020.

[42] Y. Li, C. Huang, and C. C. Loy, "Dense intrinsic appearance flow for human pose transfer," in *CVPR*, 2019.

[43] N. Neverova, R. Alp Guler, and I. Kokkinos, "Dense pose transfer," in *ECCV*, 2018.

[44] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[45] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *CVPR*, 2014.

[46] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.

[47] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016.

[48] L. Gao, D. Chen, J. Song, X. Xu, D. Zhang, and H. T. Shen, "Perceptual pyramid adversarial networks for text-to-image synthesis," in *AAAI*, 2019.

[49] M. Wang, G.-Y. Yang, R. Li, R.-Z. Liang, S.-H. Zhang, P. M. Hall, and S.-M. Hu, "Example-guided style-consistent image synthesis from semantic labeling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1495–1504.

[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[51] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *CVPR*, 2017.

[52] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, 2018.

[53] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *CVPR*, 2019.

[54] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in *ICCV*, 2019.

[55] X. Liu, G. Yin, J. Shao, X. Wang *et al.*, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," in *NIPS*, 2019.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[57] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv:1409.0473*, 2014.

[58] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv:1508.04025*, 2015.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[60] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014.

[61] L. Zheng, L. Shen, L. Tian, S. Wang, J. Bu, and Q. Tian, "Person re-identification meets image search," *arXiv:1502.02171*, 2015.

[62] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "End-to-end deep learning for person search," *arXiv:1604.01850*, 2016.

[63] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *IEEE international workshop on performance evaluation for tracking and surveillance*, 2007.

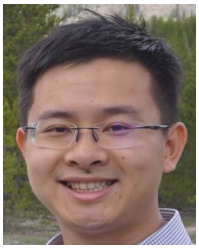
[64] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *ACCV*, 2012.

[65] A. Singh, V. Natarajan, Y. Jiang, X. Chen, M. Shah, M. Rohrbach, D. Batra, and D. Parikh, "Pythia-a platform for vision & language research," in *SysML Workshop, NeurIPS*, 2018.

[66] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NIPS*, 2017.
[67] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *arXiv:1905.08233*, 2019.
[68] K. E. Ak, J. H. Lim, J. Y. Tham, and A. A. Kassim, "Attribute manipulation generative adversarial networks for fashion images," in *ICCV*, 2019.
[69] X. Han, Z. Wu, W. Huang, M. R. Scott, and L. S. Davis, "Compatible and diverse fashion image inpainting," *arXiv:1902.01096*, 2019.
[70] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *CVPR*, 2016.

Jiaya Jia received the PhD degree in Computer Science from Hong Kong University of Science and Technology in 2004 and is currently a full professor in Department of Computer Science and Engineering at the Chinese University of Hong Kong (CUHK). He was a visiting scholar at Microsoft Research Asia from March 2004 to August 2005 and conducted collaborative research at Adobe Systems in 2007. He is Associate Editor-in-Chief of IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) and is in editorial board of International Journal of Computer Vision (IJCV). He continuously served as area chairs for ICCV, CVPR, AAAI, ECCV, and several other conferences for organization. He was on program committees of major conferences in graphics and computational imaging, including ICCP, SIGGRAPH, and SIGGRAPH Asia. He received the Young Researcher Award 2008 and Research Excellence Award 2009 from CUHK. He is a Fellow of the IEEE.

Xiaogang Xu is currently a third-year PhD student in the Chinese University of Hong Kong. He received his bachelor degree from Zhejiang University. He obtained the Hong Kong PhD Fellowship in 2018. He serves as a reviewer for CVPR and ECCV. His research interest includes deep learning, image manipulation, generative adversarial networks, etc.



Ying-Cong Chen received his PhD degree from the Chinese University of Hong Kong. He is currently a postdoctoral associate at the Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology. He obtained the Hong Kong PhD Fellowship in 2016. He serves as a reviewer for IJCV, TIP, CVPR, ICCV, ECCV, BMVC, IJCAI, AAAI, etc. His research interest includes deep learning, image generation and editing, generative adversarial networks, etc.

Xin Tao received the BS degree from the Honors Programme, School of Electronic Information and Electrical Engineering (SEIEE), Shanghai Jiao Tong University (SJTU) in 2013, and the PhD degree from the Department of Computer Science and Engineering, the Chinese University of Hong Kong (CUHK) in 2018. He worked as Senior Researcher in YouTu X-Lab, Tencent from 2018 to 2020. He joined Kuaishou Technology since 2020. He served as reviewers for TPAMI, ToG, IJCV, CVPR, ICCV, ECCV, ACCV, etc. His research interests currently lie in image/video restoration, editing and various generation tasks.