Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Where you edit is what you get: Text-guided image editing with region-based attention

Changming Xiao[a], Qi Yang[a], Xiaoqiang Xu[b], Jianwei Zhang[c], Feng Zhou[b], Changshui Zhang[a,*]

[a] *Institute for Artificial Intelligence, Tsinghua University (THUAI); Beijing National Research Center for Information Science and Technologies (BNRist); Department of Automation, Tsinghua University, Beijing, 100084, P.R.China*
[b] *Algorithm Research, Aibee Inc., Beijing, P.R.China*
[c] *Department of Informatics, Universitt Hamburg, Institute of Technical Aspects of Multimodal Systems (TAMS), Hamburg, Germany*

ARTICLE INFO

Article history:
Received 17 September 2022
Revised 21 November 2022
Accepted 20 February 2023
Available online 24 February 2023

Keywords:
Generative adversarial networks
Text-guided image editing
Spatial disentanglement

ABSTRACT

Leveraging the abundant knowledge learned from pre-trained multi-modal models like CLIP has recently proved to be effective for text-guided image editing. Though convincing results have been made when combining the image generator StyleGAN with CLIP, most methods need to train separate models for different prompts, and irrelevant regions are often changed after editing due to the lack of spatial disentanglement. We propose a novel framework that can edit different images according to different prompts in **one** model. Besides, an innovative region-based spatial attention mechanism is adopted to explicitly guarantee the locality of editing. Experiments mainly in the face domain verify the feasibility of our framework and show that when multi-text editing and local editing are accomplishable, our method can complete practical applications like sequential editing and regional style transfer.

1. Introduction

In recent years, we have witnessed rapid progress of generative models in various domains ranging from image [1], natural language [2] to speech [3]. Particularly on image synthesis, the pioneering work of StyleGAN [4–6] can generate high-resolution photo-realism images on specific domains like faces and cars. Moreover, the latent space of StyleGAN is highly structured, indicating that multiple visual attributes have their corresponding directions in the latent space. Recent works [7–10] thus focused on analyzing the latent space and finding specific directions for image editing.

However, these methods usually need large quantities of annotated data to find one specific direction or can only passively discover directions with thorough manual examinations. To develop a more user-friendly interface, StyleCLIP [11] originally proposes to take natural language into account. Regarding Contrastive Language-Image Pre-training (CLIP) model [12] as a universal attribute classifier, StyleCLIP manipulates the image in a freer manner. Along this line of thought, several works [13,14] seek for different ways to leverage the power of CLIP to enable text-guided image manipulation or generation. As CLIP provides precise image-text semantic similarity, the main idea of these works is to optimize the parameters of the generator to improve feature alignments between generated images and given texts.

Though compelling image editing results have been made, recent works have their limitations. First of all, most contemporary works are optimization-based, that is, per-prompt or per-sample optimization is needed when encountering a new text or even a new image, which limits their scalability. Secondly, different from image generation, the purpose of image editing is to semantically manipulate the relevant regions while keeping the rest unchanged. However, most current works just ensure that the edited image has the target attribute but unmentioned attributes are often changed simultaneously. Although the disentanglement of StyleGAN's latent space is well studied [7], it is implicit and statistical, which means its validity is not guaranteed in all cases. Thus these methods often lead to unwanted changes.

In this paper, we aim to resolve these limitations. First, similar to pre-CLIP period text-to-image generation works [15,16], we use a mapping module conditional on the text embedding to edit the original latent code of the image generator. We adopt a similar structure as in LAFITE [14]. Unlike image generation, our editing task does not need to model the real image distribution. Thus we embrace a lightweight version of LAFITE which excludes the discriminator. Treating text information as conditional input, we can

[*] Corresponding author.
E-mail address: zcs@mail.tsinghua.edu.cn (C. Zhang).

https://doi.org/10.1016/j.patcog.2023.109458
0031-3203/© 2023 Elsevier Ltd. All rights reserved.

use one model to edit different images with multiple manipulations, which expands the applicability of our method.

Second, we propose to use spatial attention maps to specify the editing region. Different from previous works [7,10] which detect style channels for specific semantic regions, we explicitly demand spatial disentanglement. With spatial-wise feature blending, our approach is strictly localized while channel-wise methods have their failure cases. The most related work to us is FEAT [17], which also uses spatial attention maps. However, they need to train separated models for different prompts. As spatial attention maps induce additional optimization degrees of freedom, it is difficult to choose the hyper-parameters to balance different losses, and different prompts usually require different recipes to stabilize training and produce sensible results (See Fig. 7). Thus their framework can not be directly extended to the multi-text setting that uses only one model to handle different prompts.

To alleviate the difficulty of optimization, we introduce region-based attention mechanism, which clusters the feature map of StyleGAN into meaningful semantic regions [10] and allocates attention to these regions instead of pixels. Specifically, since spatial attention aims to restrict the changeable area size of the image, local attributes, such as those related to eyes, are naturally more favorable than global attributes, such as those related to expression. Therefore, we adopt a region-based attention mechanism to eliminate the effect of area size by averaging the attention value over the regional area. With above designs, we explicitly realize spatial disentanglement, which is conducive to sequential editing [9] and regional style transfer [10].

We summarize our contributions as follows: First, we propose a novel framework that enables stable training of multi-text image editing within one model. Second, we adopt a region-based attention mechanism to ensure spatially-localized editing, in which we utilize the semantic properties of StyleGAN's latent space. Finally, we show that several practical applications become feasible due to these designs, some of which can not be achieved by existing works to the best of our knowledge.

The rest of the paper is organized as follows. Section 2 introduces related works on text-guided image editing and enhancing spatial disentanglement. Section 3 describes our method and focuses on two main components. Section 4 reports experimental results and analyzes our attention mechanism. In the end, Section 5 summarizes the paper.

2. Related works

StyleGAN-based Image Editing. StyleGAN [4–6] is one of the most popular Generative Adversarial Networks (GANs) [18] in the image domain. It is designed to have a disentangled latent space [7], which contributes to its ability of editing. A common practice in this field is to find semantic directions in StyleGAN's latent space, and image editing is carried out via moving latent codes along these directions. Methods on how to find these directions can be grouped as supervised ones [8,19] and unsupervised ones [20,21]. In a supervised manner, a large number of images with the target attribute are usually needed to train an attribute classifier. The unsupervised approaches, on the other hand, have adopted classical unsupervised learning methods to find salient transformation directions in the latent space. Though annotated samples are not needed in unsupervised methods, the discovered directions are restricted and considerable human efforts are needed to verify the semantics of each found direction. Our approach is most related to supervised methods, but we treat the pre-trained CLIP as a universal attribute classifier, thus we can conduct manipulation freely without extra data annotation.

CLIP for Image Generating and Editing. With the development of contrastive learning [22] and the introduction of attention mechanism into the vision-language field [23], a new multi-modal representation learning model CLIP [12] is proposed recently. CLIP is trained on 400 million collected text-image pairs, and it aligns the features of paired texts and images. StyleCLIP [11] first combines the powerful representations of CLIP with image generators and enables text-guided image editing. The main idea of StyleCLIP is to optimize the generator to increase the similarity, measured by CLIP, between the generated image and the given text. Compared to auto-regressive methods like DALL-E [24], this optimization-based method needs fewer computation costs [11,14]. Later works employing CLIP mostly follow this paradigm [13,25]. Though high fidelity results can be achieved for a diverse set of manipulations, they often need per-prompt or even per-sample optimization. Recently, LAFITE [14] is proposed for text-to-image generation. Its generation process is conditional on text information, thus only one model is needed for multiple texts. Our work adopts its structure of the mapping module, but as we perform editing, our training objectives are different. Besides, we introduce spatial attention maps to enable local editing.

Local Editing. Spatial disentanglement is vital for image editing, as the essence of editing is to alter the related regions while keeping the unrelated regions unchanged. Several works have focused on editing local regions, and they can be distinguished into two flavors: implicit channel-wise intervention [7,9,10] and explicit spatial-wise blending [17,26–29]. The former takes the advantage of the disentanglement of StyleGAN's latent space, as it is found that certain channels in the latent space consistently correlate with specific semantic regions, thus modifying particular channels can lead to local changes. These methods need additional segmentation networks to detect specific channels [7,9] or human supervision to evaluate found channels [10]. Moreover, this spatial disentanglement is statistical, and may not be true for rare cases. As for explicit methods, they usually rely on user-specified masks [26] or need additional attribute classifiers [27]. As a combination of these two flavors, GAN Dissection [30] detects the channel for certain visual concepts and operates spatially in the channel to insert the concepts locally. Most related to our work is FEAT [17], which only uses text descriptions to obtain the explicit spatial-wise attention map. Although no additional annotations are needed and the locality can be guaranteed, FEAT needs to train an individual model for each text, and FEAT is found hard to train stably. As improvements, our work enables multi-text editing within one model and introduces a novel way to simplify the optimization process.

3. Method

3.1. Overview

We use StyleGAN2 [5] as the backbone image generator, and we introduce the text information in its StyleSpace [5], which is shown to be more suitable for editing [7,11]. The text input is first encoded by the text encoder of CLIP [12]. Then we propose a mapping module to transform style codes of the original image into shifted ones conditioned on the text embedding. To enable local editing, a region-based attention module is adopted to provide an attention map according to correlations between the image region and the text. The attention map blends the original features of the generator with the shifted ones. Lastly, the blended features are put forward through the generator to obtain the edited image. During training, the StyleGAN2 generator and the CLIP encoder both remain frozen. An illustration of our method is shown in Fig. 1.

3.2. The mapping module

We design the mapping module `Map` following LAFITE [14], which constructs an effective mapping from the CLIP embedding

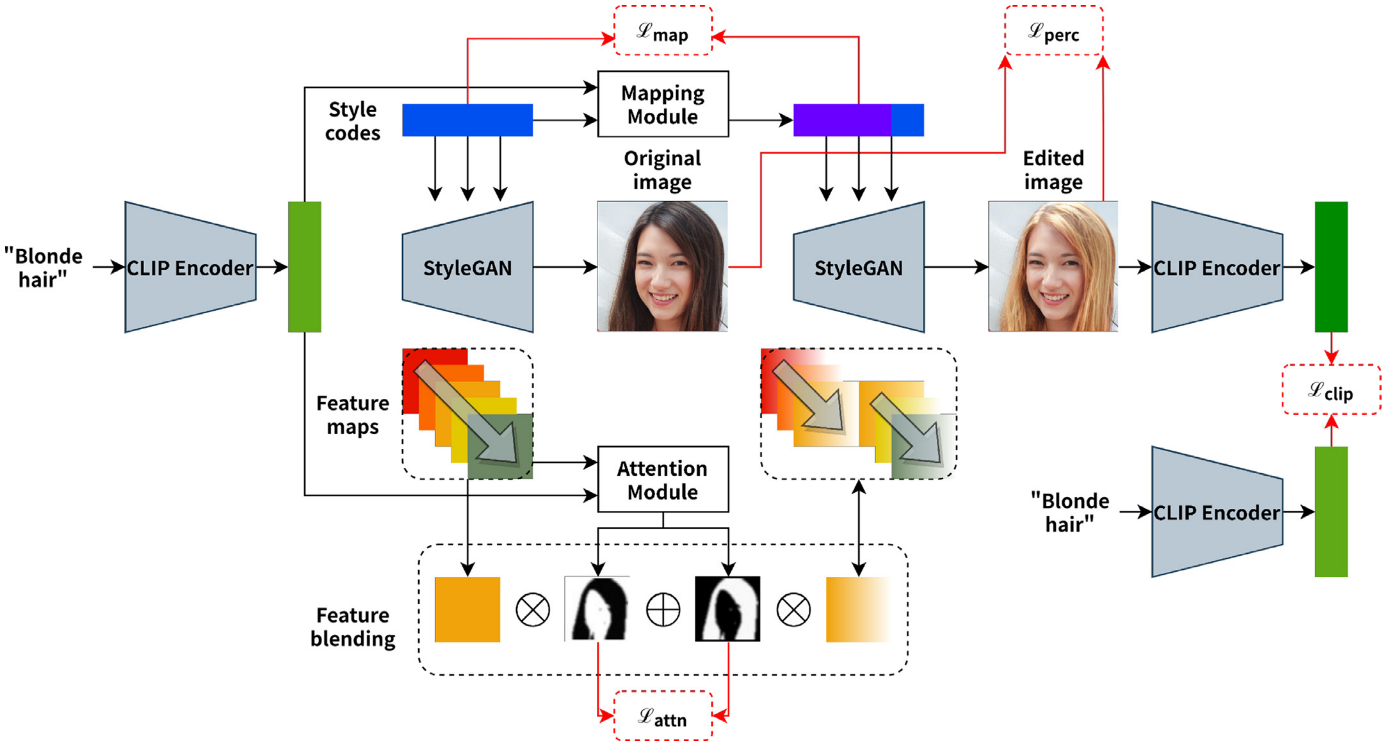

Fig. 1. An overview of our method. Given a text prompt and the original image, the mapping module is used to modify the style codes while the attention module processes the feature maps of the generator to attain the spatial attention map. The attention map is used to fuse the original features and the altered ones in some specific layers. At last, the blended features are used to generate the edited image. Best viewed in color.

space to the StyleGAN2 StyleSpace. Specifically, we define the CLIP encoding of the input prompt as h, and the style code from the ith layer of StyleGAN2 as s_i. The mapping module aims to obtain the modified style codes $u_i = \mathtt{Map}_i(s_i, h)$.

We first transform h into the condition code c via a two-layer fully-connected (FC) network F_i. We also add a single-layer FC network A_i^1 to convert s_i to the same space and obtain s_i^t. Next we concatenate s_i^t and c and transform them to the modified style codes u_i^{init} using another single-layer FC network A_i^2. Finally, we adopt the truncation trick [5] to prevent the style codes from deviating too far:

$$u_i = s_i + \alpha(u_i^{init} - s_i), \tag{1}$$

where α is the hyper-parameter of truncation that controls the editing degree.

The trainable parts of the mapping module $\mathtt{Map}_i$ are F_i, A_i^1 and A_i^2. To sum up, the calculation of layer i's mapping module is as follows:

$$u_i = \mathtt{Map}_i(s_i, h) = s_i + \alpha(A_i^2(\text{concat}[A_i^1(s_i), F_i(h)]) - s_i). \tag{2}$$

3.3. The attention module

3.3.1. The overall calculation process

The attention module $\mathtt{Attn}$ takes the features of StyleGAN2 layers as inputs and outputs a single-channel feature map as the spatial attention. More specifically, we define the feature of the ith layer of the generator as $f_i \in \mathbb{R}^{C_i \times H_i \times W_i}$, where C_i represents the number of channels and $H_i \times W_i$ represents the resolution. A 1 × 1 modulated convolution layer M_i is applied for each layer to transform the feature map. The modulated convolution scales each input channel of the original standard convolution separately according to modulation weights. And we introduce the text information by using a learned affine transformation to convert h, the CLIP encoding of the input text, to the modulation weights. Thus we get $a_i = M_i(f_i, h)$, where $a_i \in \mathbb{R}^{C \times H_i \times W_i}$ is the output layer-wise attention map, and C is the number of output channels same for different layers.

To integrate the layer-wise attention maps together, we interpolate each a_i to a certain resolution $H \times W$ to get $a_i^* \in \mathbb{R}^{C \times H \times W}$. Then a_i^* from different layers can be concatenated along the channel dimension, and they are fed to the final 1 × 1 modulated convolution layer M^{fin} followed by a sigmoid activation layer to produce the initial spatial attention map $a \in \mathbb{R}^{1 \times H \times W}$.

After acquiring the attention map a, we apply it to blend the features from specific layers of the generator as in [31]. The details of this operation can be found in the Appendix, and it is worth noting that it needs a careful scheme to guarantee the locality as StyleGAN2 has skip connections.

3.3.2. The region-based attention mechanism

The above implementation is just a multi-text extension of FEAT [17], and our preliminary experiments have shown that training is usually unstable, even for single-text. We attribute it to the additional optimization degree of freedom brought by attention. The model now has two choices to reduce the loss: to change the style codes or to change the editing regions, and it is usually tough for the model to find equilibrium among them.

Instead of starting from the entire image, we propose a way to select from the structured regions, which makes it easier for the model to find a good optimization route without going back and forth on various options. The structured regions are acquired by applying k-means clustering [32] to the activation vectors from a given layer of the generator, which is also adopted by various works like [10,33] to make training more effective. More precisely, the feature map of one certain layer has dimensionality of $C \times H \times W$, where C is the number of channels, and $H \times W$ is the spatial dimension. Collecting N samples and flattening the features,

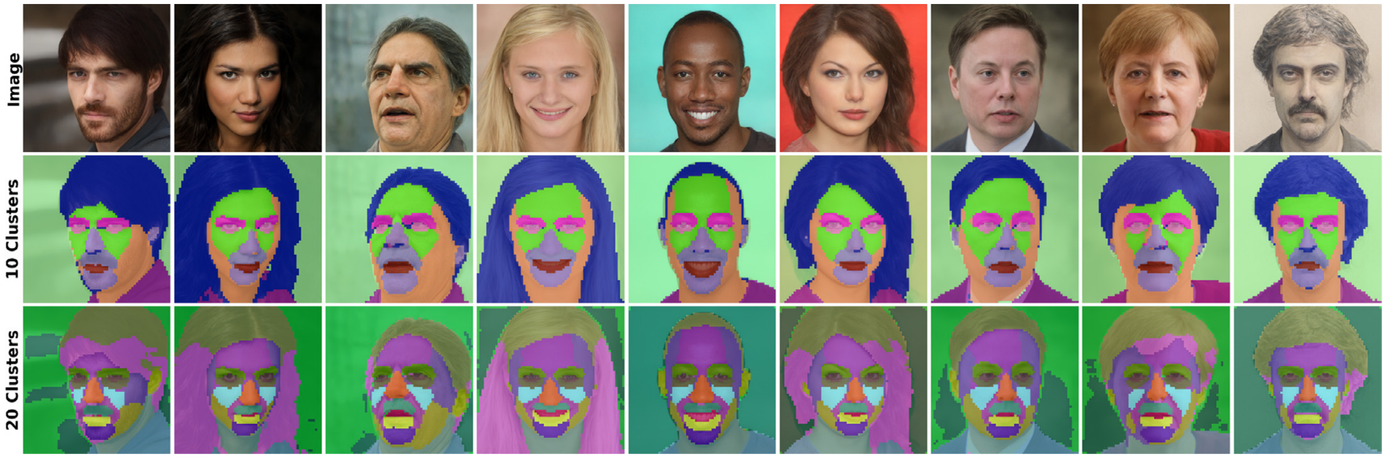

Fig. 2. *k*-means clustering results. The 1st row shows the image. The 2nd row and the 3rd row show clustering results for $k = 10$ and $k = 20$ respectively. Best viewed in color.

we can get $N \times H \times W$ C-dimensional vectors and k-means algorithm is applied on these vectors.

We conduct a pilot study in the face domain. After applying k-means to the activation vectors at the 13th layer of the pre-trained StyleGAN2 generator, we obtain semantic segmentation of faces. As shown in Fig. 2, clustered regions are semantically meaningful and the discovered clusters are generalizable at the same time. This property may come from a good scene understanding required for generation. As for our approach, we cluster the features using 100 samples in advance to obtain the clusters. When processing new images, semantic regions can be segmented according to the stored cluster centers, and we choose the editing region from multiple semantic areas rather than the whole pixel space. We calculate the average attention value of one semantic region and take it as the new attention value of all locations in this region. Then the clustered spatial attention map $\bar{a} \in \mathbb{R}^{1 \times H \times W}$ is obtained from the initial attention map a via the above Clus operation.

For a generator with n layers, the trainable parts of the attention module `Attn` are $M_i, for\ \ i = 1, 2, \cdots, n,$ and M^{fin}. To sum up, the calculation of our attention module can be encapsulated as follows:

$$\bar{a} = \mathrm{Clus}(\mathrm{Sigmoid}(M^{fin}(\mathrm{concat}_{i=1}^{n}[\mathrm{Interpolate}(M_i(f_i, h))], h))), \tag{3}$$

which can be abbreviated as $\bar{a} = \mathtt{Attn}(f_{i=1}^{n}, h)$.

3.4. Training objectives

Our goal is to edit images based on different text inputs while keeping irrelevant regions unchanged. For one image x with m different text inputs, each text input is encoded by CLIP to form text embeddings $\left\{h^j\right\}_{j=1}^{m}$.

First, we aim to reduce the distance between the generated image and the conditional prompts h^j in the latent space of CLIP. Donate x_{edit}^{j} and h_{edit}^{j} as the edited image and its corresponding CLIP feature respectively, we define a contrastive loss $\mathcal{L}_{clip}$ as follows:

$$\mathcal{L}_{clip} = -\sum_{j=1}^{m} \log \frac{\exp(\cos(h_{edit}^{j}, h^j)/\tau)}{\sum_{i=1}^{m} \exp(\cos(h_{edit}^{i}, h^j)/\tau)}, \tag{4}$$

where $\cos(\cdot)$ represents cosine similarity and τ is the temperature hyper-parameter. As observed in [22], minimizing this objective maximizes a lower bound on the mutual information between $\{h_{edit}^{j}\}$ and $\{h^j\}$, which encourages the editing process to make full use of the text information.

Second, we regularize the region-based attention map to encourage editing on a more compact region. At the same time, we encourage each pixel value of the initial attention map to be close to the mean value of its belonging region, which is conducive to stable training. The attention regularization loss $\mathcal{L}_{attn}$ is defined as follows:

$$\mathcal{L}_{attn} = \lambda_{a1} \sum_{j=1}^{m} \sum_{l=1}^{k} \left\| \bar{a}_l^j \right\|_1 + \lambda_{a2} \sum_{j=1}^{m} \left\| a^j - \bar{a}^j \right\|_2^2, \tag{5}$$

where the clustered spatial attention map $\bar{a}^j$ for prompt j is defined in Eq. (3), $\bar{a}_l^j$ is the mean attention value on region l, and a^j is the initial attention map before Clus process. Note that the k regions are clustered based on the feature vectors from the image generator as stated in Section 3.3.2. And λ_{a1} and λ_{a2} are hyper-parameters to balance different losses.

Third, to further keep the unrelated attributes preserved and obtain more natural results after editing, we regularize the modification strength, which is defined as the Euclidean distance between original style codes and edited ones. The mapping regularization loss $\mathcal{L}_{map}$ is defined as:

$$\mathcal{L}_{map} = \sum_{j=1}^{m} \sum_{i=1}^{n} \left\| u_i^j - s_i \right\|_2^2, \tag{6}$$

where s_i is the style code of x at the ith layer of the n-layer generator, and u_i^j is the corresponding edited style code for prompt j defined in Eq. (2). For real images, style codes can be obtained via inversion methods like *e4e* [34].

Last, the perceptual distance [35] between images x and x_{edit}^{j} is adopted as the regularization target to get smooth editing results:

$$\mathcal{L}_{perc} = \sum_{j=1}^{m} \mathrm{perc}(x_{edit}^{j}, x), \tag{7}$$

where $\mathrm{perc}(\cdot)$ represents the calculation of perceptual loss.

To sum up, our full objective can be written as:

$$\mathcal{L} = \mathcal{L}_{clip} + \mathcal{L}_{attn} + \lambda_m \mathcal{L}_{map} + \lambda_p \mathcal{L}_{perc}, \tag{8}$$

where λ_m and λ_p are hyper-parameters to balance different losses.

3.5. Inference phase

After training, text-guided image editing can be conducted with the pre-trained image generator, text encoder, and the trained

Algorithm 1 Text-Guided Image Editing with Region-Based Attention.

```
Model: image generator $S$ with $n$ layers, text encoder $E_{text}$; mapping module Map, attention module Attn; blending layer $k$.
Input: style codes $\{s_i\}_{i=1}^{n}$ of original image $x$, editing prompt $t$.
Output: edited image $x_{edit}$.
1: Encode the editing prompt: $h = E_{text}(t)$;
2: Get the modified style codes $u_i$ using Equation (2) for $i = 1, 2, \ldots, n$;
3: Get $S'$s features maps $f_i$ and $\widehat{f_i}$ corresponding to $s_i$ and $u_i$ respectively for $i = 1, 2, \ldots, n$;
4: Get the attention map $\bar{a}$ using Equation (3);
5: Get the blended feature $\tilde{f}_k$ of $f_k$ and $\widehat{f_k}$ at the $kth$ layer of $S$ using $\bar{a}$ and apply $S$ on $\tilde{f}_k$ to get the edited image $x_{edit}$;
6: return $x_{edit}$.
```

mapping module, attention module. This inference process is summarized in Algorithm 1. The 5th step (the blending operation) is further elaborated in the Appendix.

4. Experiments and discussions

As our method has versatile applications and it is hard to compare under different settings, we first conduct experiments under the conventional text-guided image editing setting, thus qualitative and quantitative comparisons can be made with previous works. Then we verify the effectiveness of our designed components. Finally, we explore some novel utilizations of our method. We train and evaluate our method mainly in the face domain which has practical application scenarios, but it can be easily extended to other domains such as cars, and we put the results in the Appendix. More implementation details are also provided in the Appendix. The source code is available at https://github.com/Big-Brother-Pikachu/Where2edit.

4.1. Experimental settings

Datasets

We use the StyleGAN2 model pre-trained on the Flickr-Faces-HQ Dataset (FFHQ) [4] as our generator. FFHQ contains high quality 1024 × 1024 images of human faces. We use the textual descriptions from Multi-Modal-CelebA-HQ (MM CelebA-HQ) [36] as our prompt corpus. There are 300,000 depictions in total, and they are generated automatically based on the facial attributes of real CelebAMask-HQ [37] images.

Evaluation metrics

We evaluate the generated images from three aspects: visual quality, alignment with text, and attribute preservation. Fréchet Inception Distance (FID) [38] and Inception Score (IS) [39] are used to measure the naturalness of edited images. The cosine distance between the image and the conditional text before and after editing in the CLIP latent space is compared. And we record the ratio of getting closer after editing (Success Rate). Next, we take up a face recognition model FaceNet [40] to measure the identity variation after editing using the cosine similarity between the extracted features (ID). Moreover, PSNR and SSIM are calculated in the intersection region of non-hair regions before and after editing when hair-related prompts are used. ID, PSNR, and SSIM all focus on attribute preservation.

4.2. Standard text-guided editing

For the standard text-guided editing task, we compare against three closely related baselines: TediGAN [36], StyleCLIP [11], and FEAT [17]. Besides, a specialized hair editing method HairCLIP [13] is also considered. TediGAN optimizes the latent codes of StyleGAN2 through multi-modal alignment to obtain editing results. It is time-consuming in the inference phase and it lacks spatial disentanglement. StyleCLIP explores three techniques that combine CLIP with StyleGAN2. We focus on the **mapper** approach, which is closer to our framework. This approach trains separate models for different texts, and it also lacks spatial disentanglement. FEAT employs an attention network to explicitly encourage changes only in the intended regions. Though spatial disentanglement is achieved, it is found hard to balance the training objectives when optimizing directly from the entire area. To make it worse, different models are needed for multiple prompts. HairCLIP can conduct multi-text editing through one model. However, it is a specialized model for hair editing and it needs an extra parsing model to encourage attribute preservation. Moreover, the spatial disentanglement of HairCLIP is not guaranteed, as it only implicitly takes not changing irrelevant attribute areas as an optimization objective. Different from the above approaches, our method trains a unified model for different descriptions and we adopt a novel attention mechanism to achieve explicit spatial disentanglement.

Table 1
Quantitative comparisons when using generic prompts. ↑ indicates that higher is better while ↓ indicates that lower is better. Numbers in **bold** indicate the best results.

Methods	FID (↓)	Success Rate (↑)	ID (↑)	Model used (↓)
TediGAN [36]	21.70	**100.0%**[a]	0.59	**1**
StyleCLIP [11]	11.40	99.29%	0.82	5
FEAT [17]	8.11	100.0%[b]	0.84	5
Ours	**7.22**	98.72%	**0.86**	**1**

[a] We only generate 1,000 samples to calculate the Success Rate for [36], as [36] needs instance-level optimization, which is time-consuming.
[b] It is the assumed best results for [17], as [17] have not provided their codes yet.

4.2.1. Quantitative and qualitative comparisons using generic prompts

Following FEAT [17], we evaluate different methods on 10,000 randomly generated samples of each prompt and average the results of 5 generic editing prompts used in FEAT. As shown in Table 1, our method can edit with different prompts through one model, and the success rate is still at a high level. TediGAN [36], which is based on instance-level optimization, has a 100% success rate, but it is time-consuming. Compared with the real-time editing ability of other methods, TediGAN needs around 30 s to edit an image. Additionally, our method shows higher visual quality (lower FID value) and better attribute preservation (higher ID value) compared with previous methods, which is the advantage our novel attention module brings. As for statistical tests, the null hypothesis that our ID results are not better can be rejected as the p-value is smaller than 10^{-10}.

The metrics used above can not fully reveal the performance of each method, which is a common issue in the image generation field, thus we conduct qualitative comparisons. As shown in Fig. 3, all methods modify the face according to the prompts, but TediGAN and StyleCLIP tend to change unrelated regions. For instance, TediGAN changes the expression when editing the hair, while it alters the hair when manipulating the expression. As for StyleCLIP, it changes the cloth color to purple in the “Purple Hair” case. In contrast, our method can locate the edited region precisely. More qualitative results of our method are shown in Fig. 4. It is worth noting that all our edits are done through one model with only one forward calculation.

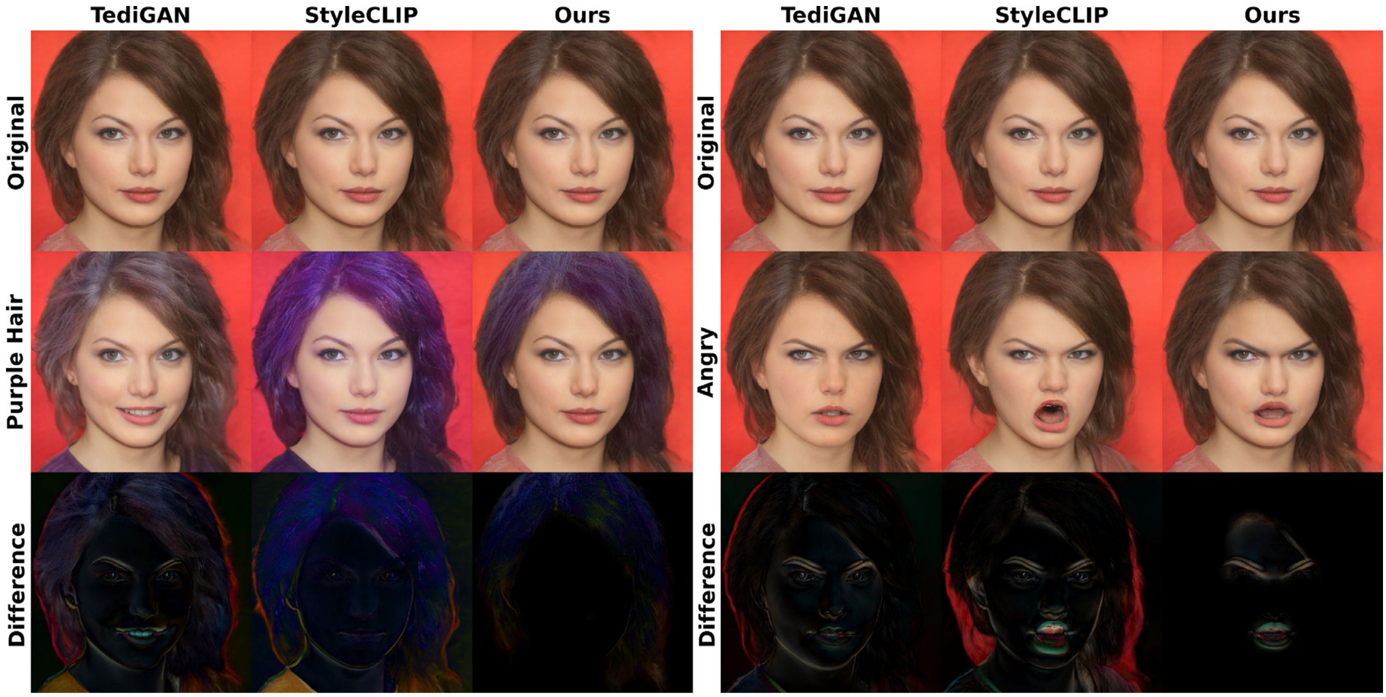

Fig. 3. Qualitative comparisons with TediGAN and StyleCLIP. The 1st row shows the original images, and the 2nd row shows the edited results. The differences in pixel values before and after editing are shown in the last row. Best viewed in color.

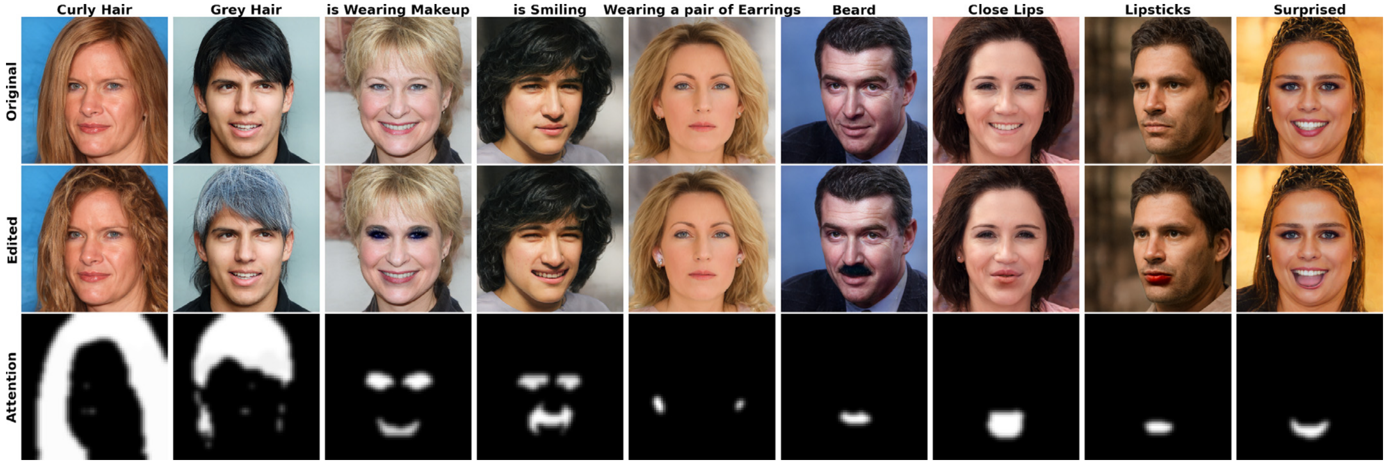

Fig. 4. More qualitative results of our method. The original images in the 1st row are edited by the prompts above, and the corresponding edited results are shown in the 2nd row. The text-guided attention maps are shown in the last row. Best viewed in color.

Table 2
Quantitative comparisons when using hair prompts. ↑ indicates that higher is better while ↓ indicates that lower is better. Numbers in **bold** indicate the best results.

Methods	PSNR (↑)	SSIM (↑)	ID (↑)	Model used (↓)
TediGAN [36]	24.1	0.79	0.17	**1**
StyleCLIP [11]	23.2	0.87	0.79	10
HairCLIP [13]	27.8	0.92	0.83	**1**
Ours	**31.2**	**0.98**	**0.84**	**1**

4.2.2. Quantitative and qualitative comparisons using hair prompts

Following HairCLIP [13], we evaluate different methods on the CelebA-HQ test set and average the results of 10 hair editing prompts used in HairCLIP. As shown in Table 2, our method better preserves irrelevant attributes. Note that though we did not add identity consistency loss to our objective function while HairCLIP did, we still perform better on ID. As for statistical tests, the null hypothesis that our results are not better can be rejected as the p-value is smaller than 10^{-10}. Regarding visual results, our method can edit the hair with comparable quality as shown in Fig. 5. Besides, we explicitly ensure changes do not occur in the non-hair region. For example, other methods change the mouth and the collar in the “Bowlcut Hairstyle” case, while we do not. Though our model has not been specially trained for hair editing, it can locate the hair region precisely with the attention module and change the hair according to different prompts properly with the mapping module.

4.2.3. Out-of-domain results

We manifest the generalization capability of our method when applying it to out-of-domain data from MetFaces [5]. We use the StyleGAN2 model pre-trained on FFHQ and train the mapping module and the attention module with face description corpus and

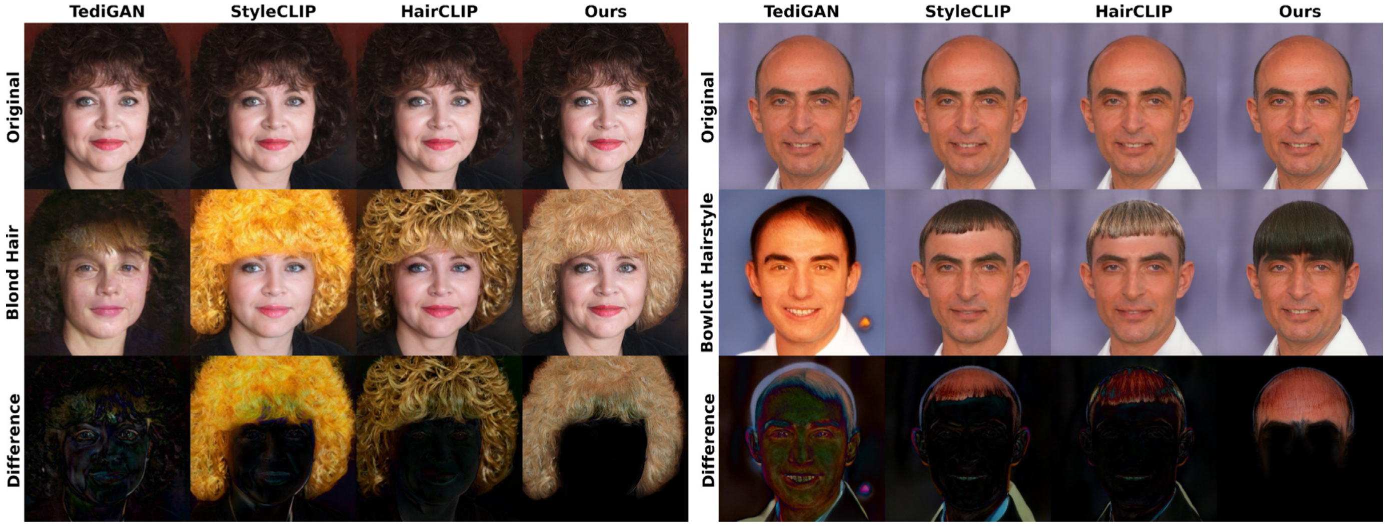

Fig. 5. Qualitative comparisons with TediGAN, StyleCLIP, and HairCLIP. The last row shows the difference between original images and edited images in pixel space. Best viewed in color.

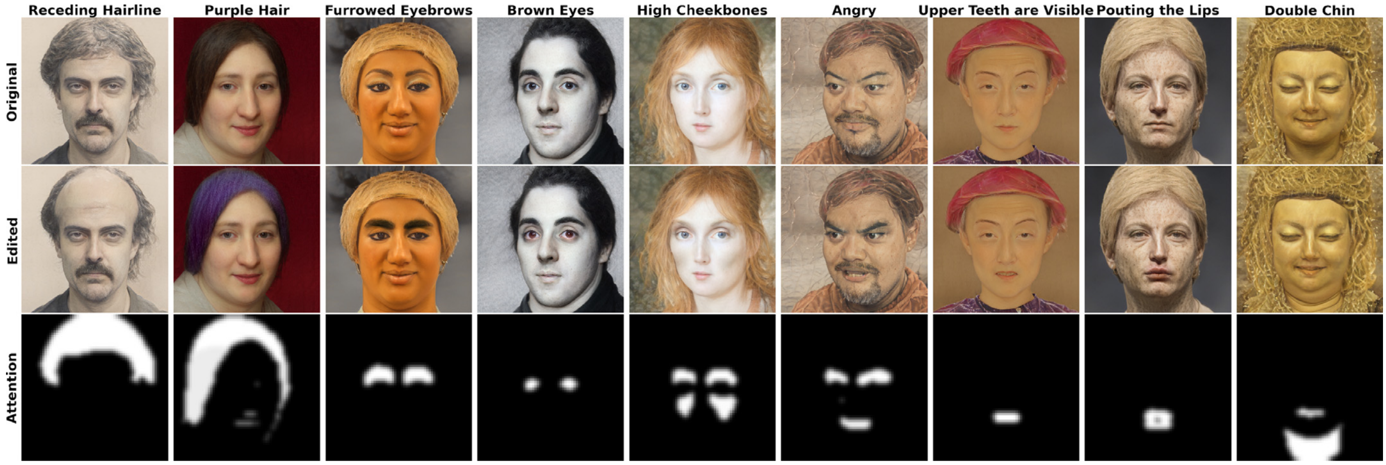

Fig. 6. The editing results on out-of-domain images. The texts above each column are the editing prompts. The 1st row shows the original images including portraits, sculptures, and sketches. The 2nd row shows the edited images while the 3rd row shows the corresponding attention masks. Best viewed in color.

synthesis face images. The *e4e* model trained for FFHQ is used to inverse the images from MetFaces. As shown in Fig. 6, our method works surprisingly well on the out-of-domain data. Not only does the mapping module complete the right modifications, but also does the attention module focus on the correct regions. This phenomenon indicates the nice properties of our method.

4.3. Attention mechanism

In this section, we verify the effectiveness of our attention mechanism. We perform an ablation study regarding different ways of introducing attention.

We compare our implementation to one variant that mutes the attention module (W/O Attn) and two variants that eliminate the clustering process (W/O Clus). In the absence of the attention module, irrelevant attributes are often altered, which is undesirable for image editing. Besides, when eliminating the clustering process as the framework proposed in FEAT, the attention region is less accurate. We speculate that when it is harder to select the region of interest, the optimization process will be less stable, thus accurate results cannot be obtained. In addition, different facial regions have different area sizes, which makes it difficult to decide the regularization hyper-parameters. We try different recipes of hyper-parameters, where “W/O Clus, small λ” represents a small effect of attention regularization, while “W/O Clus, large λ” in contrast represents a large effect of attention regularization. They both have their own drawbacks as the former recipe usually leads to full face attention, while the latter one often partially edits large semantic regions. Our attention mechanism which regularizes the average attention value of different clustered regions fends off these problems.

We first show quantitative evaluation results for the ablation study in Table 3. We choose prompts with different sizes of regions of interest, which are roughly ordered as “Chubby”, “Rosy cheeks”, “Goatee”, “Sculpted eyebrows”, and “Pouting the lips”. We randomly generate 10,000 samples for each prompt to calculate the evaluation values. “W/O Attn” and “W/O Clus, small λ” change human identity to a great extent. On the other hand, “W/O Clus, large λ” can not handle prompts with more global attributes (“Chubby”, “Rosy cheeks”) as the learned attention masks just cover small regions. Our proposed attention mechanism can produce more precise edits for prompts with different relevant region sizes. As for statistical tests, the null hypothesis that our ID results are not better can be rejected as the p-value is smaller than 10^{-10}.

The qualitative comparisons are shown in Fig. 7, “W/O Attn” alters almost the whole region while attention-based variants only

Table 3
Quantitative comparisons with different attention mechanism variants. ↑ indicates that higher is better while ↓ indicates that lower is better. Numbers in **bold** indicate the best results. “1” represents “W/O Attn”, “2” represents “W/O Clus, small λ”, and “3” represents “W/O Clus, large λ”.

Prompts	FID (↓)				ID (↑)				Success Rate (↑)			
	1	2	3	Ours	1	2	3	Ours	1	2	3	Ours
Chubby	5.03	4.29	**0.06**	0.57	0.68	0.67	**0.97**	0.95	99.87%	**99.88%**	88.26%	96.49%
Rosy cheeks	9.30	8.35	**0.25**	6.92	0.79	0.78	**0.90**	0.90	**100.0%**	**100.0%**	98.59%	99.99%
Goatee	**9.64**	13.43	26.29	18.42	0.62	0.62	**0.72**	0.69	99.52%	**99.70%**	99.34%	99.10%
Sculpted eyebrows	6.02	5.73	2.64	**2.47**	0.71	0.70	0.79	**0.85**	99.76%	**99.77%**	99.60%	98.71%
Pouting the lips	17.20	16.44	14.75	**11.42**	0.71	0.64	0.78	**0.79**	**99.99%**	99.98%	99.98%	**99.99%**
Average	9.44	9.65	8.80	**7.96**	0.70	0.68	0.83	**0.84**	99.83%	**99.87%**	97.15%	98.86%

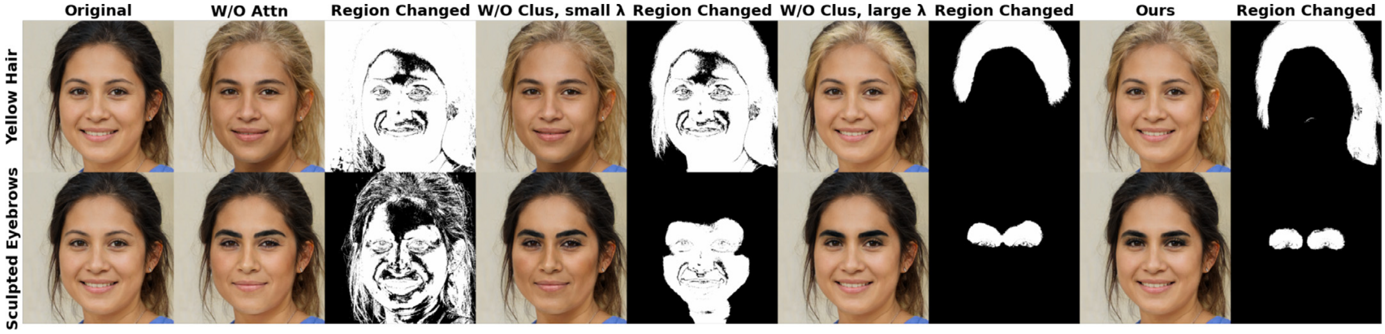

Fig. 7. Qualitative comparisons with different variants of the attention mechanism. The region-changed diagram indicates the region where pixel value changes relative to the original image exceed a certain threshold. Best viewed in color.

Table 4
Running time of different methods. The training time is in minutes and is recorded when good performance is achieved. The per-image inference time is in seconds.

Phase	TediGAN [36]	StyleCLIP [11]	HairCLIP [13]	Ours	W/O Attn	W/O Clus
Training (min)	-	67	124	132=20+112	99	108
Inference (sec/img)	35.80	0.22	0.32	0.29	0.26	0.28

affect the interior of attention areas. More specifically, “W/O Attn” greatly changes the identity of the edited person. For hair editing, the mouth, the eyes, and the skin color are changed, while for eyebrow editing, the mouth, the eyes are changed. “W/O Clus, small λ” also leads to full face modification. In addition, “W/O Clus, large λ” only partially edits large semantic regions like hair, which leads to unnatural results as we seldom have half blond and half black hair like this. Our proposed attention mechanism decides where to edit more precisely, which is conducive to improving editing quality.

4.4. Running time analysis

We conduct running time analysis for different methods. As algorithms all operate on an individual sample, the training time increases linearly with the number of training steps while the inference time increases linearly with the number of samples. Although our method adopts a contrastive loss for training, it is calculated within a batch, thus the linearity remains. We evaluate training time using 7 NVIDIA GTX 2080Ti GPUs and we exclude the training time for pre-trained models. Our method needs 20 min for prior clustering and TediGAN has no parameters to train. The inference time is evaluated using 1 NVIDIA GTX 2080Ti GPU. We record the time to edit 50 images and average the results. As shown in Table 4, the training time and inference time of our method are at the same level as those of StyleCLIP and HairCLIP. It is worth noting that we can all edit images in real time. And we are all much faster than the instance-level optimization method TediGAN. Besides, StyleCLIP needs to train separate models for different editing prompts and the result shown here is for training one prompt. As for different variants of our method, the running time has not much difference.

4.5. Applications

Since multi-text editing and local editing are enabled, a wide range of tasks can be completed through our method. Some of them are shown in this section to demonstrate the extensibility of our method.

4.5.1. Sequential editing

One requirement of a practical image editing application is the ability to edit sequentially. Spatial disentanglement is beneficial for sequential editing, as subsequent manipulations will not affect previous ones if different regions have participated. We first compare our method with TediGAN [36] and StyleCLIP [11], which do not take spatial disentanglement into account. As shown in Fig. 8, our method only alters relevant regions, and after multi-turn manipulations, our image quality and editing precision remain at a high level. For instance, TediGAN gradually turns the man into a woman as the editing progresses on the left side of the 1st row. Moreover, the angry woman generated by StyleCLIP on the right side of the 2nd row looks unnatural. We speculate that this is due to the deviation of the feature map from its true distribution caused by multiple global changes. As for our method, each manipulation only alters the feature map at the corresponding location thanks to the attention mechanism. Thus our feature map will not be dramatically modified due to sequential editing as in these competing methods.

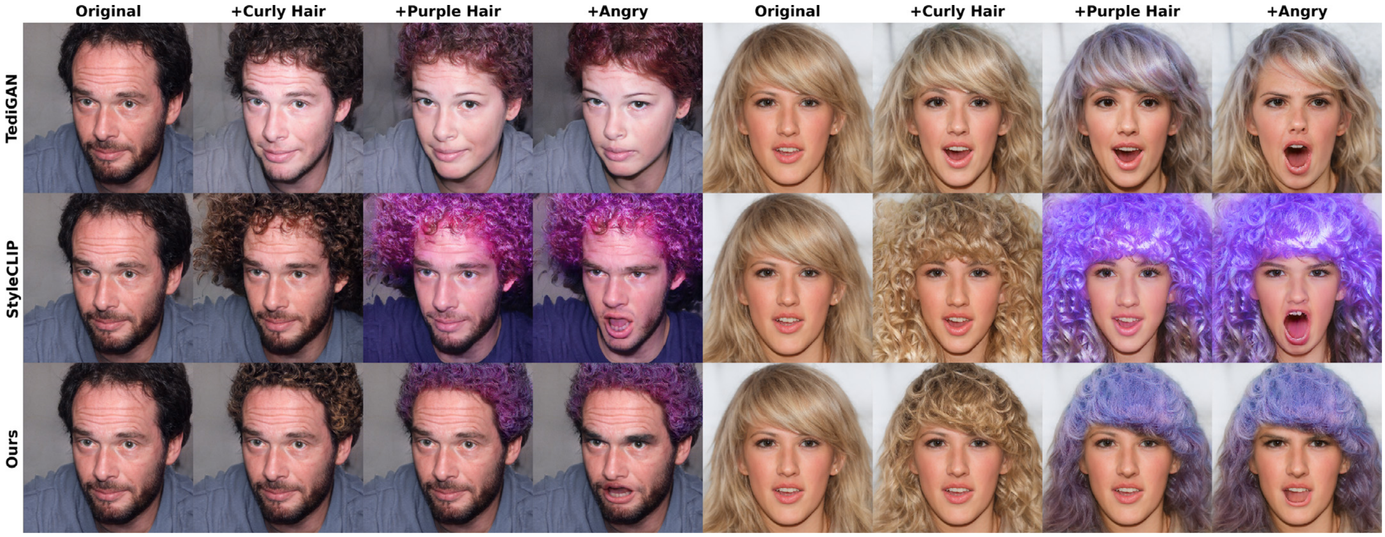

Fig. 8. Qualitative comparisons under sequential editing setting with methods that do not consider spatial disentanglement. Best viewed in color.

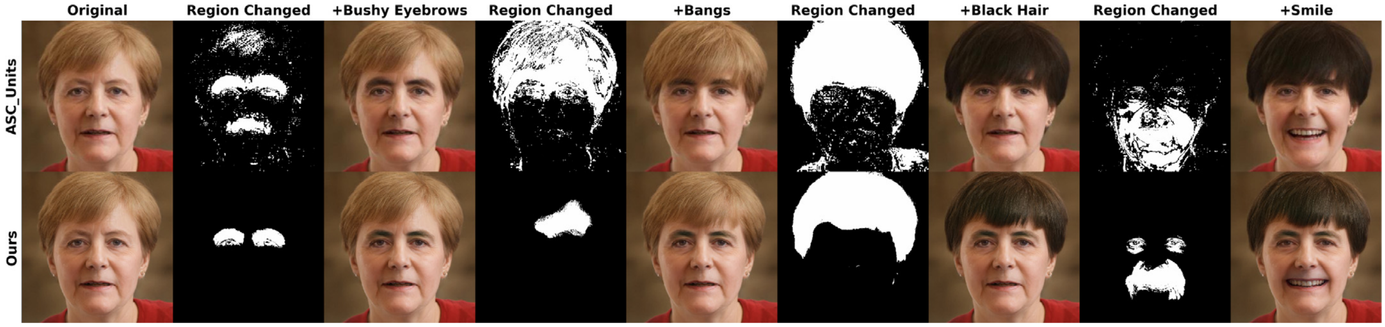

Fig. 9. Qualitative comparisons under sequential editing setting with a local editing method. The region-changed diagram indicates the region where pixel value changes relative to the previous image exceed a certain threshold. Best viewed in color.

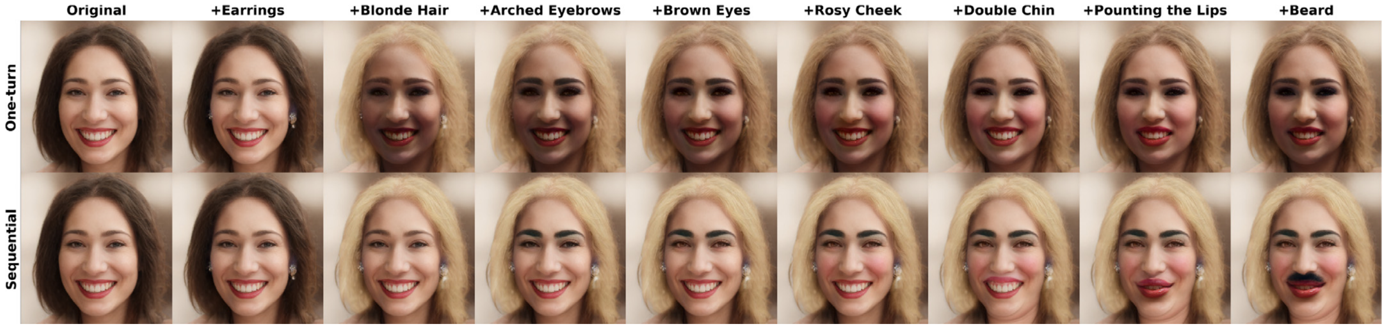

Fig. 10. Handle long prompts using sequential editing. The 1st row shows the one-turn editing results while the 2nd row shows the sequential editing results. Best viewed in color.

Besides, we compare our method with ASC_Units [9], which considers spatial disentanglement. More specifically, ASC_Units is an implicit approach that detects attribute-specific channels for local transformations. As shown in Fig. 9, our explicit spatial attention map restricts the editing region more precisely, and multiple manipulations only have minimal interactions.

Finally, we demonstrate that with sequential editing, long prompts can be processed more accurately. When the editing prompt contains a large quantity of attributes, it is difficult for the editing model to handle all of them. One solution is to divide the face description into multiple single attributes and add them to the face one by one. As shown in Fig. 10, we use long sentences to include all the previous attributes for one-turn editing. And when the prompt includes too many attributes, the editing result will not reflect all the attributes. On the contrary, the results at the 2nd row cover the entire attributes well when attributes are added sequentially.

4.5.2. Regional style transfer

Performing semantic part transformation from a reference image constitutes an interesting image editing tool. Our framework has the ability to complete this task. We use a facial description to determine the editing region through the attention module, and use the CLIP feature of the reference image as the condition of the mapping module. As shown in Fig. 11, we can adaptively change the style of one specific semantic part of the target image according to the reference image without affecting other semantic parts. Compared with simple feature blending [26], our framework can

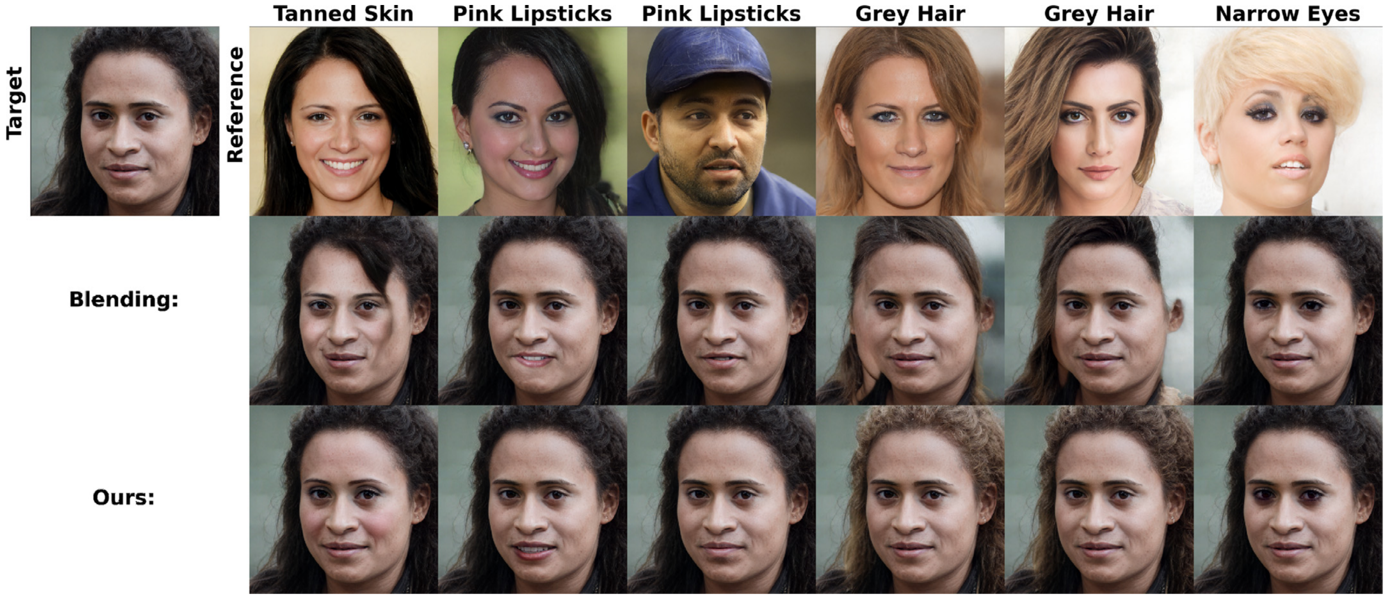

Fig. 11. Qualitative comparisons under regional style transfer setting. The target face is on the top left and reference images are on the right of the 1st row. Facial descriptions above each column are used to determine semantic parts of interest. Best viewed in color.

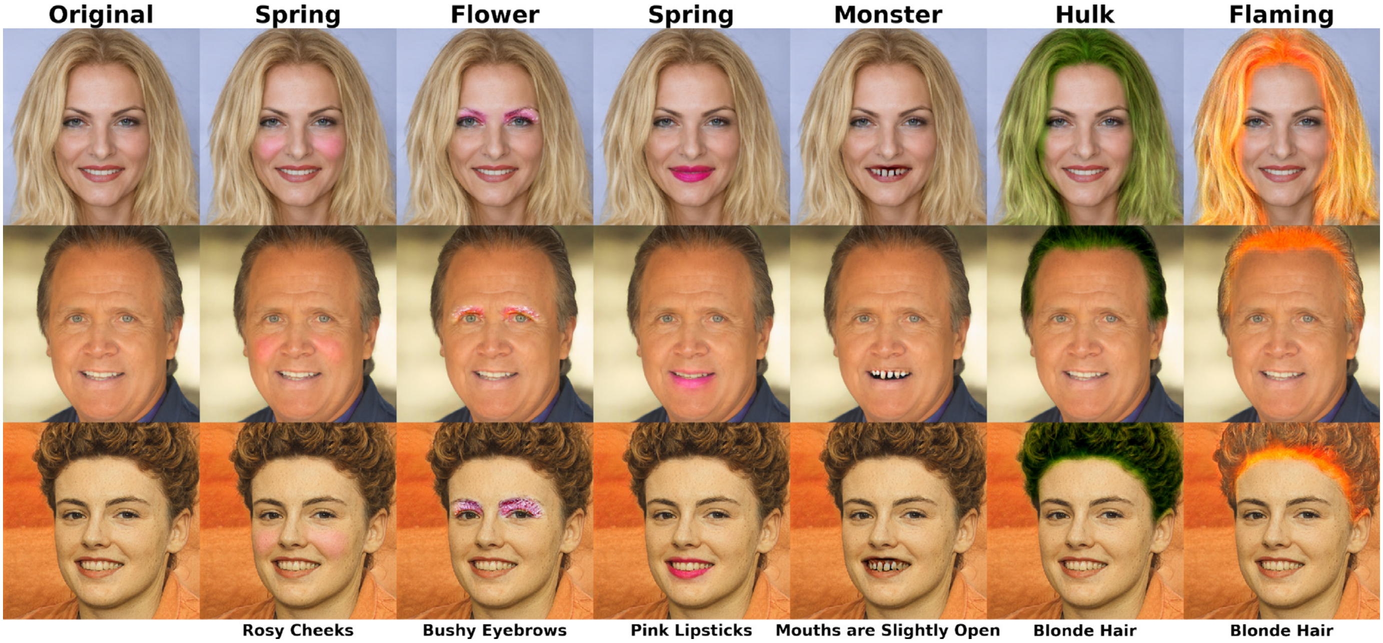

Fig. 12. Editing specific facial regions with innovative prompts. The original images are in the 1st column. Appearance prompts are shown above each column while attention prompts are shown below. Best viewed in color.

better handle the misalignment between the reference face and the target face. For instance, the simple blending results of hair style transfer produce unnatural artifacts while our method maintains the hair shape.

4.5.3. Two-prompt editing

At last, we show a more artistic application inspired by [17]. Same as Section 4.5.2, we use usual facial descriptions to decide the editing region but use more imaginative descriptions for the appearance. More specifically, we use the attention maps derived from the attention prompts to determine the editing region. Then, the appearance prompts modify the style codes through the mapping module. At last, we can get more free-form images as shown in Fig. 12. As we can see, the “Spring” prompt makes the face more splendid. The prompt “Monster” for the mouth area can lead to bucktooth. The prompt “Hulk”, a movie character, can lead to a green appearance. And the prompt “Flaming” makes the hair “Fire”.

4.6. Discussions

Recent work about local editing favors the implicit method [7,10], we develop along another route: the explicit method, which preserves irrelevant attributes better than implicit methods in principle. But current explicit methods usually require user-provided masks. We instead decide where to edit completely according to the prompts, so as to make the editing process more automatic and reduce the efforts of users. Experiment comparisons in Section 4.5.1 demonstrate the irrelevant attribute preservation ability of our method. And as our framework provides a more intuitive interface for explicit methods, it may stimulate further research in this direction.

Besides, our framework may contribute to video generation tasks as our method has a good insight into what to change and retain when synthesizing new images. Frames about a person changing expressions can be easily generated using our method, which is impossible for typical image synthesis methods as temporal continuity should be considered for videos.

Finally, local editing requires a deeper understanding of the synthesis process, as it needs to establish a finer correspondence between semantics and images. Our method can help to reveal how the large generator locates different attributes, which may inspire interpretability research.

5. Conclusion

In this paper, we propose a new approach for text-guided image editing. With a novel mapping module, our method is more flexible, as we can edit different images according to different prompts using one model in real-time. Moreover, the difference between manipulation and generation lies in the change of irrelevant attributes, as editing requires unrelated attributes well preserved. Therefore we have to choose where to edit in addition to how to change. We propose a novel region-based attention mechanism, which explicitly restricts the editing region with the spatial attention map related to the prompt. To stabilize training, we utilize the spatial information learned in the generator to simplify the localization process. Experiments mainly in the face domain demonstrate the effectiveness of our framework as well as our attention mechanism. Additionally, our method has a broad range of applications, some of which have not been well solved by previous methods.

As for the weakness of our method, we rely on the clustering result of StyleGAN2 features, which sets an upper bound on the accuracy of the attention map. Some clustering regions may be composed of several semantic parts, thus we will take trainable clustering centers into account later. Another issue relates to the editing of real images, we adopt the GAN-inversion method to obtain the style codes of real images. But the reconstruction is far from perfect in general, thus edited images usually change globally compared with the original real images. In future work, we will consider new generative models with better reconstruction quality, such as the diffusion model [25].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Key Research and Development Program (2020AAA0107800), the Natural Science Fundation of China (NSFC) and the German Research Foundation (DFG) in Project Crossmodal Learning, NSFC 62061136001/ DFG TRR-169.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patcog.2023.109458.

References

[1] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, D.N. Metaxas, StackGAN++: realistic image synthesis with stacked generative adversarial networks, IEEE Trans. Pattern Anal. Mach. Intell. 41 (8) (2019) 1947–1962.

[2] W. Li, R. Peng, Y. Wang, Z. Yan, Knowledge graph based natural language generation with adapted pointer-generator networks, Neurocomputing 382 (2020) 174–187.

[3] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A.W. Senior, K. Kavukcuoglu, WaveNet: a generative model for raw audio, in: ISCA, 2016, p. 125.

[4] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: CVPR, 2019, pp. 4401–4410.

[5] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in: CVPR, 2020, pp. 8107–8116.

[6] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, T. Aila, Alias-free generative adversarial networks, in: NeurIPS, 2021, pp. 852–863.

[7] Z. Wu, D. Lischinski, E. Shechtman, StyleSpace analysis: disentangled controls for stylegan image generation, in: CVPR, 2021, pp. 12863–12872.

[8] Y. Shen, C. Yang, X. Tang, B. Zhou, InterFaceGAN: interpreting the disentangled face representation learned by GANs, IEEE Trans. Pattern Anal. Mach. Intell. 44 (4) (2022) 2004–2018.

[9] R. Wang, J. Chen, G. Yu, L. Sun, C. Yu, C. Gao, N. Sang, Attribute-specific control units in stylegan for fine-grained image manipulation, in: ACM MM, 2021, pp. 926–934.

[10] E. Collins, R. Bala, B. Price, S. Süsstrunk, Editing in style: uncovering the local semantics of GANs, in: CVPR, 2020, pp. 5770–5779.

[11] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, D. Lischinski, StyleCLIP: text-driven manipulation of stylegan imagery, in: ICCV, 2021, pp. 2065–2074.

[12] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, in: ICML, 2021, pp. 8748–8763.

[13] T. Wei, D. Chen, W. Zhou, J. Liao, Z. Tan, L. Yuan, W. Zhang, N. Yu, HairCLIP: design your hair by text and reference image, in: CVPR, 2022, pp. 18072–18081.

[14] Y. Zhou, R. Zhang, C. Chen, C. Li, C. Tensmeyer, T. Yu, J. Gu, J. Xu, T. Sun, Towards language-free training for text-to-image generation, in: CVPR, 2022, pp. 17907–17917.

[15] T. Hinz, S. Heinrich, S. Wermter, Semantic object accuracy for generative text-to-image synthesis, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1552–1565.

[16] Y. Dong, Y. Zhang, L. Ma, Z. Wang, J. Luo, Unsupervised text-to-image synthesis, Pattern Recognit. 110 (2021) 107573.

[17] X. Hou, L. Shen, O. Patashnik, D. Cohen-Or, H. Huang, FEAT: face editing with attention, CoRR (2022) 2202.02713.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, NeurIPS, Vol. 27, 2014.

[19] R. Abdal, P. Zhu, N.J. Mitra, P. Wonka, StyleFlow: attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows, ACM Trans. Graph. 40 (3) (2021) 21:1–21:21.

[20] Y. Shen, B. Zhou, Closed-form factorization of latent semantics in GANs, in: CVPR, 2021, pp. 1532–1540.

[21] A. Voynov, A. Babenko, Unsupervised discovery of interpretable directions in the GAN latent space, in: ICML, 2020, pp. 9786–9796.

[22] T. Chen, S. Kornblith, M. Norouzi, G.E. Hinton, A simple framework for contrastive learning of visual representations, in: ICML, 2020, pp. 1597–1607.

[23] G. KV, A. Nambiar, K.S. Srinivas, A. Mittal, Linguistically-aware attention for reducing the semantic gap in vision-language tasks, Pattern Recognit. 112 (2021) 107812.

[24] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever, Zero-shot text-to-image generation, in: ICML, 2021, pp. 8821–8831.

[25] G. Kim, T. Kwon, J.C. Ye, DiffusionCLIP: text-guided diffusion models for robust image manipulation, in: CVPR, 2022, pp. 2416–2425.

[26] R. Suzuki, M. Koyama, T. Miyato, T. Yonetsuji, Collaging on internal representations: an intuitive approach for semantic transfiguration, CoRR (2018) 1811.10153.

[27] J. Kwak, D.K. Han, H. Ko, CAFE-GAN: arbitrary face attribute editing with complementary attention feature, in: ECCV, 2020, pp. 524–540.

[28] L. Gao, D. Chen, Z. Zhao, J. Shao, H.T. Shen, Lightweight dynamic conditional GAN with pyramid attention for text-to-image synthesis, Pattern Recognit. 110 (2021) 107384.

[29] Runpeng Cui, Zhong Cao, Weishen Pan, Changshui Zhang, Jianqiang Wang, Deep Gesture Video Generation with Learning on Regions of Interest, IEEE Transactions on Multimedia 22 (10) (Oct, 2020) 2551–2563.

[30] B. Zhou, D. Bau, A. Oliva, A. Torralba, Interpreting deep visual representations via network dissection, IEEE Trans. Pattern Anal. Mach. Intell. 41 (9) (2019) 2131–2145.

[31] S. Wu, H. Tang, X.-Y. Jing, J. Qian, N. Sebe, Y. Yan, Q. Zhang, Cross-view panorama image synthesis with progressive attention GANs, Pattern Recognit. 131 (2022) 108884.

[32] S.P. Lloyd, Least squares quantization in PCM, IEEE Trans. Inf. Theory 28 (2) (1982) 129–136.

[33] S. Li, L. Liu, J. Liu, W. Song, A. Hao, H. Qin, SC-GAN: subspace clustering based GAN for automatic expression manipulation, Pattern Recognit. 134 (2023) 109072.

[34] O. Tov, Y. Alaluf, Y. Nitzan, O. Patashnik, D. Cohen-Or, Designing an encoder for stylegan image manipulation, ACM Trans. Graph. 40 (4) (2021) 133:1–133:14.

[35] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: ECCV, 2016, pp. 694–711.

[36] W. Xia, Y. Yang, J. Xue, B. Wu, TediGAN: text-guided diverse face image generation and manipulation, in: CVPR, 2021, pp. 2256–2265.

[37] C. Lee, Z. Liu, L. Wu, P. Luo, MaskGAN: towards diverse and interactive facial image manipulation, in: CVPR, 2020, pp. 5548–5557.
[38] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium, in: NeurIPS, 2017, pp. 6626–6637.
[39] T. Salimans, I.J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, in: NeurIPS, 2016, pp. 2226–2234.
[40] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: a unified embedding for face recognition and clustering, in: CVPR, 2015, pp. 815–823.

Changming Xiao is a PhD student at the Department of Automation, Tsinghua University. His current interests include multi-modal learning and generative model.

Qi Yang is a PhD student at the Department of Automation, Tsinghua University. His research interests include generative model and 3D understanding.

Xiaoqiang Xu is an engineer in Algorithm Research, Aibee Inc. His research interests include face recognition and 3D reconstruction.

Jianwei Zhang is a Professor and the Director of the Group TAMS, Department of Informatics, University of Hamburg. He received the Bachelor of Engineering (Hons.) and Master of Engineering degrees from the Department of Computer Science, Tsinghua University, in 1986 and 1989, respectively, the PhD degree from the Department of Computer Science, Institute of Real-Time Computer Systems and Robotics, University of Karlsruhe, Germany, in 1994, and the Habilitation degree from the Faculty of Technology, University of Bielefeld, Germany, in 2000. His research interests include sensor fusion, intelligent robotics, and multimodal machine learning.

Feng Zhou is a senior engineer in Algorithm Research, Aibee Inc.. He received the BS degree in computer science from the Zhejiang University in 2005, the MS degree in computer science from the Shanghai Jiao Tong University in 2008, and the PhD degree in robotics from Carnegie Mellon University in 2014. His research interests include machine learning and computer vision.

Changshui Zhang is a Professor at the Department of Automation, Tsinghua University. He received the BS degree in mathematics from Peking University in 1986 and the PhD degree from the Department of Automation, Tsinghua University in 1992. His current research interests include artificial intelligence, image processing, pattern recognition, machine learning, and evolutionary computation.