Open-Edit: Open-Domain Image Manipulation with Open-Vocabulary Instructions

Xihui Liu[1⋆][0000−0003−1831−9952], Zhe Lin[2][0000−0003−1154−9907], Jianming Zhang[2], Handong Zhao[2], Quan Tran[2], Xiaogang Wang[1], and Hongsheng Li[1][0000−0002−2664−7975]

[1] The Chinese University of Hong Kong {xihuiliu,xgwang,hsli}@ee.cuhk.edu.hk
[2] Adobe Research {zlin,jianmzha,hazhao,qtran}@adobe.com

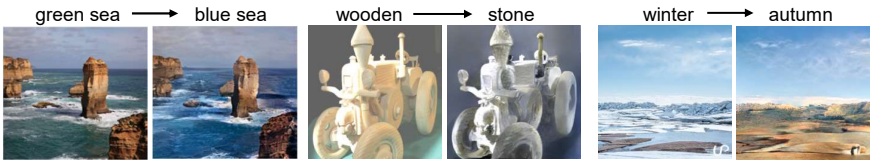

Fig. 1: Examples of open-edit. The editing instruction (top), source image (left), and manipulated image (right) are shown for each example. Our approach edits open-vocabulary color, texture, and semantic attributes of open-domain images.

Abstract. We propose a novel algorithm, named Open-Edit, which is the first attempt on open-domain image manipulation with open-vocabulary instructions. It is a challenging task considering the large variation of image domains and the lack of training supervision. Our approach takes advantage of the unified visual-semantic embedding space pretrained on a general image-caption dataset, and manipulates the embedded visual features by applying text-guided vector arithmetic on the image feature maps. A structure-preserving image decoder then generates the manipulated images from the manipulated feature maps. We further propose an on-the-fly sample-specific optimization approach with cycle-consistency constraints to regularize the manipulated images and force them to preserve details of the source images. Our approach shows promising results in manipulating open-vocabulary color, texture, and high-level attributes for various scenarios of open-domain images.[3]

1 Introduction

Automatic image editing, aiming at manipulating images based on the user instructions, is a challenging problem with extensive applications. It helps users to edit photographs and create art works with higher efficiency.

⋆ This work was done during Xihui Liu's internship at Adobe.
[3] Code is released at https://github.com/xh-liu/Open-Edit.

Several directions have been explored towards image editing with generative models. Image-to-image translation [19, 49, 12] translates an image from a source domain to a target domain. But it is restricted to the predefined domains, and cannot be generalized to manipulating images with arbitrary instructions. GAN Dissection [5] and GANPaint [4] are able to add or remove certain objects by manipulating related units in the latent space. However, they are limited to editing a small number of pre-defined objects and stuff that can be identified by semantic segmentation and can be disentangled in the latent space.

Most relevant to our problem setting is language-based image editing [51, 13, 31, 16, 28]. Some previous work [14, 11, 8] annotates the manipulation instructions and ground-truth manipulated images for limited images and scenarios. But it is infeasible to obtain such annotations for large-scale datasets. To avoid using ground-truth manipulated images, other work [51, 13, 31, 16, 28] only use images and caption annotations as training data. Given an image A and a mismatched caption B, the model is required to edit A to match B. The manipulated images are encouraged to be realistic and to match the manipulation instructions, without requiring ground-truth manipulated images as training supervision. However, it is assumed that any randomly sampled caption is a feasible manipulation instruction for the image. For example, given an image of a red flower, we can use "a yellow flower" as the manipulation instruction. But it is meaningless to use "a blue bird" as the manipulation instruction for the image of a red flower. So this approach is restricted to datasets from a specific domain (*e.g.*, flowers or birds in previous work [30]) with human-annotated fine-grained descriptions for each image, and cannot generalize to open-domain images.

In this work, we aim to manipulate open-domain images by open-vocabulary instructions with minimal supervision, which is a challenging task and has not been explored in previous work. We propose *Open-Edit*, which manipulates the visual feature maps of source images based on the open-vocabulary instructions, and generates the manipulated images from the manipulated visual feature maps. It takes advantages of the universal visual-semantic embedding pretrained on a large-scale image-caption dataset, Conceptual Captions [35]. The visual-semantic embedding model encodes any open-domain images and open-vocabulary instructions into a joint embedding space. Features within the joint embedding space can be used for localizing instruction-related regions of the input images and for manipulating the related visual features. The manipulations are performed by vector arithmetic operations between the visual feature maps and the textual features, *e.g.*, visual embedding of green apple = visual embedding of red apple - textual embedding of "*red apple*" + textual embedding of "*green apple*". Then a structure-preserving image decoder generates the manipulated images based on the manipulated visual feature maps. The image generator is trained with image reconstruction supervision and does not require any paired manipulation instruction for training. So our approach naturally handles open-vocabulary open-domain image manipulations with minimal supervision.

Moreover, to better preserve details and regularize the manipulated images, we introduce *sample-specific optimization* to optimize the image decoder with

the specific input image and manipulation instruction. Since we cannot apply direct supervisions on the manipulated images, we adopt reconstruction and cycle-consistency constraints to optimize the small perturbations added to the intermediate decoder layers. The reconstruction constraint forces the image decoder to reconstruct the source images from their visual feature maps; The cycle-consistency constraint performs a cycle manipulation (*e.g.*, red apple → green apple → red apple) and forces the final image to be similar to the original ones.

Our proposed framework, Open-Edit, is the first attempt for open-domain image manipulation with open-vocabulary instructions, with several unique advantages: (1) Unlike previous approaches that require single-domain images and fine-grained human-annotated descriptions, we only use noisy image-captions pairs harvested from the web for training. Results in Fig. 1 demonstrates that our model is able to manipulate open-vocabulary colors, textures, and semantic attributes of open-domain images. (2) By controlling the coefficients of the vector arithmetic operation, we can smoothly control the manipulation strength and achieve visual appearances with interpolated attributes. (3) The sample-specific optimization with cycle-consistency constraints further regularizes the manipulated images and preserves details of the source images. Our results achieve better visual quality than previous language-based image editing approaches.

2 Related Work

Image Manipulation with Generative Models. Zhu *et al*. [48] to defines coloring, sketching, and warping brush as editing operations and used constrained optimization to update images. Similarly, Andrew *et al*. [6] proposes Introspective Adversarial Network (IAN) which optimizes the latent space to generate manipulated images according to the input images and user brush inputs. GAN-Paint [4] manipulates the latent space of the input image guided by GAN Dissection [5], which relies on a segmentation model to identify latent units related to specific objects. This approach therefore is mainly suitable for adding or removing specific types of objects from images. Another line of work focuses on face or fashion attribute manipulation with predefined attributes and labeled images on face or fashion datasets [33, 37, 43, 36, 1, 10, 39]. In contrast, our approach aims to handle open-vocabulary image manipulation on arbitrary colors, textures, and high-level attributes without attribute annotations for training.

Language-based Image Editing. The interaction between language and vision has been studied for various applications [40, 25, 7, 26, 42]. Language-based image editing enables user-friendly control for image editing by free-form sentences or phrases as the manipulation instructions. [14, 11, 8] collects paired data (*i.e.* original images, manipulation queries, and images after manipulation) for training. However, collecting such data is time-consuming and infeasible for most editing scenarios. Other works [51, 13, 31, 16, 28, 45] only require image-caption pairs for training, but those methods are restricted to specific image domains with fine-grained descriptions such as flowers or birds. Our work extends the problem setting to open-domain images. Moreover, our approach does not rely

on fine-grained accurate captions. Instead, we use Conceptual Captions dataset, where the images and captions are harvested from the web. Concurrent work [23] conducts language-based image editing on COCO dataset. But it takes a trade-off between reconstructing and editing, restricting the model from achieving both high-quality images and effective editing at the same time.

Image-to-image Translation. Supervised image-to-image translation [19, 9, 41, 32, 27] translates images between different domains with paired training data. [38, 24, 34, 44, 49, 21] focus on unsupervised translation with unpaired training data. Consequent works focus on multi-domain [12] or multi-modal [50, 2, 18, 22]. However, one have to define domains and collect domain-specific images for image-to-image translation, which is not able to tackle arbitrary manipulation instructions. On the contrary, our approach performs open-vocabulary image manipulation without defining domains and collecting domain-specific images.

3 Method

Our goal of open-vocabulary open-domain image manipulation is to edit an arbitrary image based on an open-vocabulary manipulation instruction. The manipulation instructions should indicate the source objects or attributes to be edited as well as the target objects or attributes to be added. For example, the manipulation instruction could be "red apple $\rightarrow$ green apple".

There are several challenges for open-vocabulary open-domain image manipulation: (1) It is difficult to obtain a plausible manipulation instruction for each training image. And it is infeasible to collect large-scale ground-truth manipulated images for fully supervised training. (2) The open-domain images are of high variations, compared with previous work which only consider single-domain images like flowers or birds. (3) The manipulated images may fail to preserve all details of the source images. Previous work on language-guided image editing uses other images' captions as the manipulation instruction for an image to train the model. However, it assumes that all images are from the same domain, while cannot handle open-domain images, *e.g.*, a caption for a flower image cannot be used as the manipulation instruction for a bird image.

To achieve open-vocabulary open-domain image manipulation, we propose a simple but effective pipeline, named Open-Edit, as shown in Fig. 2. It exploits the visual-semantic joint embedding space to manipulate visual features by textual features, and then decodes the images from the manipulated feature maps. It is composed of visual-semantic embedding, text-guided visual feature manipulation, structure-preserving image decoding, and sample-specific optimization.

There are two stages for training. In the first stage, we pretrain the *visual-semantic embedding* (VSE) model on a large-scale image-caption dataset to embed any images and texts into latent codes in the visual-semantic embedding space (Fig. 2(a)). Once trained, the VSE model is fixed to provide image and text embeddings. In the second stage, the *structure-preserving image decoder* is trained to reconstruct the images from the visual feature maps encoded by the VSE model, as shown in Fig. 2(b). The whole training process only requires the

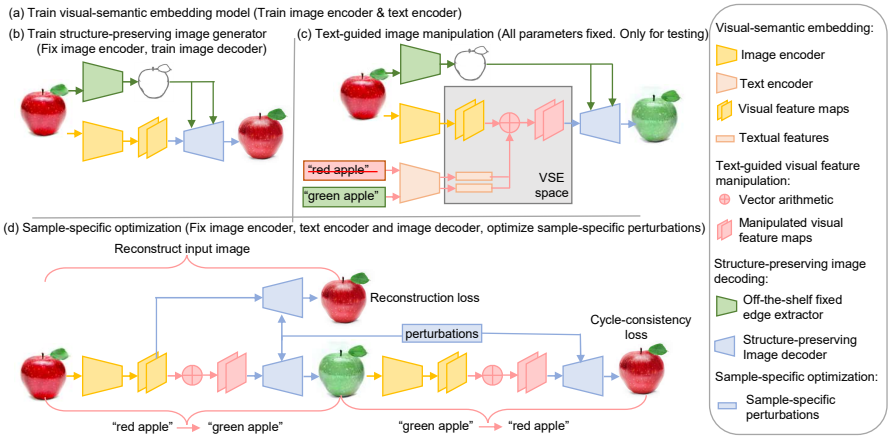

Fig. 2: The pipeline of our Open-Edit framework. (a) and (b) show the training process. (c) and (d) illustrate the testing process. To simplify the demonstration, edge extractor and text encoder are omitted in (d).

images and noisy captions harvested from the web, and does not need any human annotated manipulation instructions or ground-truth manipulated images.

During inference (Fig. 2(c)), the visual-semantic embedding model encodes the input images and manipulation instructions into visual feature maps and textual features in the joint embedding space. Then *text-guided visual feature manipulation* is performed to ground the manipulation instructions on the visual feature maps and manipulate the corresponding regions of the visual feature maps with the provided textual features. Next, the structure-preserving image decoder generates the manipulated images from the manipulated feature maps.

Furthermore, in order to regularize the manipulated images and preserve details of the source images, we introduce small sample-specific perturbations added to the intermediate layers of the image decoder, and propose a *sample-specific optimization* approach to optimize the perturbations based on the input image and instruction, shown in Fig. 2(d). For a specific image and manipulation instruction, we put constraint on both the reconstructed images and the images generated by a pair of cycle manipulations (*e.g.*, red apple $\rightarrow$ green apple $\rightarrow$ red apple). In this way, we adapt the image generator to the specific input image and instruction and achieve higher quality image manipulations.

3.1 A Revisit of Visual-Semantic Embedding

To handle open-vocabulary instructions and open-domain images, we use a large-scale image-caption dataset to learn a universal visual-semantic embedding space. Convolutional neural networks (CNN) and long short-term memory networks (LSTM) are used as encoders to transform images and captions into

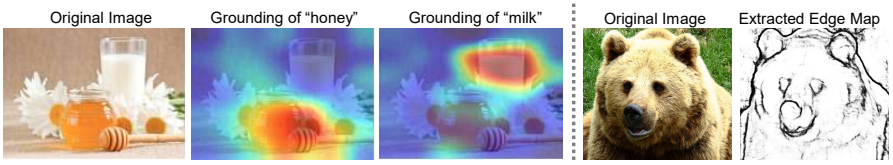

Fig. 3: Left: an example of grounding results by visual-semantic embedding. Right: an example of edge map extracted by off-the-shelf edge detector.

visual and textual feature vectors. A triplet ranking loss with hardest negatives, as shown below, is applied to train the visual and textual encoders [15].

$$\mathcal{L}(\mathbf{v}, \mathbf{t}) = \max_{\hat{\mathbf{t}}}[m + \langle \mathbf{v}, \hat{\mathbf{t}} \rangle - \langle \mathbf{v}, \mathbf{t} \rangle]_+ + \max_{\hat{\mathbf{v}}}[m + \langle \hat{\mathbf{v}}, \mathbf{t} \rangle - \langle \mathbf{v}, \mathbf{t} \rangle]_+ \quad (1)$$

where $\mathbf{v}$ and $\mathbf{t}$ denote the visual and textual feature vectors of a positive image-caption pair. $\hat{\mathbf{v}}$ and $\hat{\mathbf{t}}$ are the negative image and caption features in the mini-batch. $[x]_+ = \max(0, x)$, and m is the constant margin for the ranking loss. $\langle \mathbf{v}, \mathbf{t} \rangle$ denotes the dot product to measure the similarity between the visual and textual features. With the trained VSE model, the visual feature maps before average pooling $\mathbf{V} \in \mathbb{R}^{1024 \times 7 \times 7}$ is also embedded into the VSE space.

3.2 Text-guided Visual Feature Manipulation

The universal visual-semantic embedding space enables us to manipulate the visual feature maps with the text instructions by vector arithmetic operations, similar to that of word embeddings (*e.g.*, "king" - "man" + "woman" = "queen") [29]. When manipulating certain objects or attributes, we would like to only modify specific regions while keeping other regions unchanged. So instead of editing the global visual feature vector, we conduct vector arithmetic operations between the visual feature maps $\mathbf{V} \in \mathbb{R}^{1024 \times 7 \times 7}$ and textual feature vectors.

We first identify which regions in the feature map to manipulate, *i.e.*, ground the manipulation instructions on the spatial feature map. The VSE model provides us a soft grounding for textual queries by a weighted summation of the image feature maps, similar to class activation maps (CAM) [47]. We use the textual feature vector $\mathbf{t} \in \mathbb{R}^{1024 \times 1}$ as weights to compute the weighted summation of the image feature maps $\mathbf{g} = \mathbf{t}^\top \mathbf{V}$. This scheme gives us a soft grounding map $\mathbf{g} \in \mathbb{R}^{7 \times 7}$, which is able to roughly localize corresponding regions in the visual feature maps related to the textual instruction. Examples of the text-guided soft grounding results are shown in Fig. 3 (left). We adopt the grounding map as location-adaptive coefficients to control the manipulation strength at different locations. We further adopt a coefficient α to control the global manipulation strength, which enables continuous transitions between source images and the manipulated ones. The visual feature vector at spatial location (i, j) (where $i, j \in \{0, 1, ...6\}$) in the visual feature map $\mathbf{V} \in \mathbb{R}^{1024 \times 7 \times 7}$, is denoted

as $\mathbf{v}^{i,j} \in \mathbb{R}^{1024}$. We define the following types of manipulations by vector arithmetics weighted by the soft grounding map and the coefficient α.

Changing Attributes. Changing object attributes or global attributes is one of the most common manipulations. The textual feature embeddings of the source and target concepts are denoted as $\mathbf{t}_1$ and $\mathbf{t}_2$. respectively. For example, if we want to change a "red apple" into a "green apple", $\mathbf{t}_1$ would be the textual embedding of phrase "red apple" and $\mathbf{t}_2$ would be the embedding of phrase "green apple". The manipulation of image feature vector $\mathbf{v}^{i,j}$ at location (i, j) is,

$$\mathbf{v}_m^{i,j} = \mathbf{v}^{i,j} - \alpha\langle\mathbf{v}^{i,j}, \mathbf{t}_1\rangle\mathbf{t}_1 + \alpha\langle\mathbf{v}^{i,j}, \mathbf{t}_1\rangle\mathbf{t}_2, \tag{2}$$

where $i, j \in \{0, 1, ...6\}$, and $\mathbf{v}_m^{i,j}$ is the manipulated visual feature vector at location (i, j) of the 7×7 feature map. We remove the source features $\mathbf{t}_1$ and add the target features $\mathbf{t}_2$ to each visual feature vector $\mathbf{v}^{i,j}$. $\langle\mathbf{v}^{i,j}, \mathbf{t}_1\rangle$ is the value of the soft grounding map at location (i, j), calculated as the dot product of the image feature vector and the source textual features. We can also interpret the dot product as the projection of the visual embedding $\mathbf{v}^{i,j}$ onto the direction of the textual embedding $\mathbf{t}_1$. It serves as a location-adaptive manipulation strength to control which regions in the image should be edited. α is a hyper-parameter that controls the image-level manipulation strength. By smoothly increasing α, we can achieve smooth transitions from source to target attributes.

Removing Concepts. In certain scenarios, objects, stuff or attributes need to be removed, *e.g.*, remove the beard from a face. Denote the semantic embedding of the concept we would like to remove as $\mathbf{t}$. The removing operation is

$$\mathbf{v}_m^{i,j} = \mathbf{v}^{i,j} - \alpha\langle\mathbf{v}^{i,j}, \mathbf{t}\rangle\mathbf{t}. \tag{3}$$

Relative Attributes. Our framework also handles relative attribute manipulation, such as making a red apple less red or tuning the image to be brighter. Denote the semantic embedding of the relative attribute as $\mathbf{t}$. The strength of the relative attribute is controlled by the hyper-parameter α. By smoothly adjusting α, we can gradually strengthen or weaken the relative attribute as

$$\mathbf{v}_m^{i,j} = \mathbf{v}^{i,j} \pm \alpha\langle\mathbf{v}^{i,j}, \mathbf{t}\rangle\mathbf{t}. \tag{4}$$

3.3 Structure-Preserving Image Decoding

After deriving the manipulated feature map $\mathbf{V}_m \in \mathbb{R}^{1024\times7\times7}$, an image decoder takes $\mathbf{V}_m$ as input and generates the manipulated images.

Since we do not have paired data for training and it is difficult to generate plausible manipulation instructions for each image, we train the image decoder with only the reconstruction supervisions, as shown in Fig. 2(b). Specifically, we fix the VSE model to transform an image $\mathbf{I}$ into the feature maps $\mathbf{V}$ in the joint embedding space, and train a generative adversarial network to reconstruct the input image from $\mathbf{V}$. The generator is trained with the hinge-based adversarial

loss, discriminator feature matching loss, and perceptual loss.

$$\begin{aligned}\mathcal{L}_G = &- \mathbb{E}[D(G(\mathbf{V}))] + \lambda_{VGG}\mathbb{E}[\sum_{k=1}^{N}\frac{1}{n_k}||F_k(G(\mathbf{V})) - F_k(\mathbf{I})||_1] \\ &+ \lambda_{FM}\mathbb{E}[\sum_{k=1}^{N}\frac{1}{m_k}||D_k(G(\mathbf{V})) - D_k(\mathbf{I})||_1], \\ \mathcal{L}_D = &-\mathbb{E}[\min(0, -1 + D(\mathbf{I}))] - \mathbb{E}[min(0, -1 - D(G(\mathbf{V})))], \end{aligned} \tag{5}$$

where $\mathcal{L}_D$ and the first term of $\mathcal{L}_G$ are the hinge-based adversarial loss. The second term of $\mathcal{L}_G$ is the perceptual loss, calculated as the VGG feature distance between the reconstructed image and the input image. The third term of $\mathcal{L}_G$ is the discriminator feature matching loss, which matches the intermediate features of the discriminator between the reconstructed image and the input image. n_k and m_k are the number of elements in the k-th layer of the VGG network and discriminator, respectively. λ_{VGG} and λ_{FM} are the loss weights. Although not being trained on manipulated feature maps, the image decoder learns a general image prior. So during inference, the decoder is able to generate manipulated images when given the manipulated feature maps as input.

Furthermore, we incorporate edge constraints into the image decoder to preserve the structure information when editing image appearances. We adopt an off-the-shelf CNN edge detector [17] to extract edges from the input images. The extracted edges, as shown in Fig. 3 (right), are fed into intermediate layers of the image decoder by spatially-adaptive normalization [32]. Specifically, we use the edge maps to predict the spatially-adaptive scale and bias parameters of batch-normalization layers. We denote the edge map as $\mathcal{E}$. Denote the feature map value of the n-th image in the mini-batch at channel c and location (h, w) as $f_{n,c,h,w}$. Denote the mean and standard deviation of the feature maps at channel c as μ_c and σ_c, respectively. The spatially-adaptive normalization is

$$\gamma_{c,h,w}(\mathcal{E})\frac{f_{n,c,h,w} - \mu_c}{\sigma_c} + \beta_{c,h,w}(\mathcal{E}), \tag{6}$$

where γ and β are two-layer convolutions to predict spatially-adaptive scale and bias for BN layers. With the edge constraints, the decoder is able to preserve the structures and edges of the source images when editing the image appearances.

3.4 Sample-Specific Optimization with Cycle-Consistency Constraints

The vector arithmetic manipulation operations may not be precise enough, because some attributes might be entangled and the visual-semantic embedding space may not be strictly linear. Moreover, the image decoder trained with only reconstruction supervision is not perfect and might not be able to reconstruct all details of the source image. To mitigate those problems, we adopt a sample-specific optimization approach to adapt the decoder to the specific input image and manipulation instruction.

For each image and manipulation instruction (*e.g.*, "red apple" → "green apple"), we apply a pair of cycle manipulations to exchange the attributes forth and back (*e.g.*, "red apple" → "green apple" → "red apple"). The corresponding source and manipulated images are denoted as $\mathbf{I} \rightarrow \mathbf{I}_m \rightarrow \mathbf{I}_c$. We incorporate a cycle-consistency loss to optimize the decoder to adapt to the specific image and manipulation instruction. In this way, we can regularize the manipulated image and complete the details missed during encoding and generating. We also adopt a reconstruction loss to force the optimized decoder to reconstruct the source image without manipulating the latent visual features. The reconstruction loss $\mathcal{L}_{rec}$ and cycle-consistency loss $\mathcal{L}_{cyc}$ are the summation of L_1 loss and perceptual loss, computed between the source image $\mathbf{I}$ and the reconstructed $\mathbf{I}_r$ or the cycle manipulated image $\mathbf{I}_c$,

$$\mathcal{L}_{cyc} = ||\mathbf{I}_c - \mathbf{I}||_1 + \lambda \sum_{k=1}^{N} \frac{1}{n_k} ||F_k(\mathbf{I}_c) - F_k(\mathbf{I})||_1, \tag{7}$$

$$\mathcal{L}_{rec} = ||\mathbf{I}_r - \mathbf{I}||_1 + \lambda \sum_{k=1}^{N} \frac{1}{n_k} ||F_k(\mathbf{I}_r) - F_k(\mathbf{I})||_1, \tag{8}$$

where λ is the loss weight for perceptual loss and F_k is the k-th layer of the VGG network with n_k features.

However, directly finetuning the decoder parameters for a specific image and manipulation instruction would cause severe overfitting to the source image, and the finetuned decoder would not be able to generate the high-quality manipulated image. Alternatively, we fix the decoder parameters and only optimize a series of additive perturbations of the decoder network, as shown in Fig. 2(d). For each specific image and manipulation, the sample-specific perturbations are initialized as zeros and added to the intermediate layers of the decoder. The perturbation parameters are optimized with the manipulation cycle-consistency loss and reconstruction loss on that specific image and manipulation instruction. So when generating the manipulated images, the optimized perturbations can complete the high-frequency details of the source images, and regularize the manipulated images. Specifically, the image decoder is divided into several decoder blocks $G_1, G_2, \cdots, G_n$ ($n = 4$ in our implementation), and the perturbations are added to the decoder between the n blocks,

$$G'(\mathbf{V}) = G_n(G_{n-1}(\cdots(G_1(\mathbf{V}) + \mathbf{P_1})\cdots) + \mathbf{P_{n-1}}), \tag{9}$$

where $\mathbf{P_1}, \cdots, \mathbf{P_{n-1}}$ are the introduced perturbations. We optimize the perturbations by the summation of reconstructions loss, manipulation cycle-consistency loss, and a regularization loss $\mathcal{L}_{reg} = \sum_{i=1}^{n-1} ||\mathbf{P}_i||_2^2$.

Those optimization steps are conducted only during testing. We adapt the perturbations to the input image and manipulation instruction by the introduced optimization process. Therefore, the learned sample-specific perturbations models high-frequency details of the source images, and regularizes the manipulated images. In this way, the generator with optimized perturbations is able to generate photo-realistic and detail-preserving manipulated images.

4 Experiments

4.1 Datasets and Implementation Details

Our visual-semantic embedding model (including image encoder and text encoder) and image decoder are trained on Conceptual Captions dataset [35] with 3 million image-caption pairs harvested from the web. The images are from various domains and of various styles, including portrait, objects, scenes, and others. Instead of human-annotated fine-grained descriptions in other image captioning datasets, the captions of Conceptual Captions dataset are harvested from the Alt-text HTML attribute associated with web images. Although the images are of high variations and the captions are noisy, results show that with large datasets, the VSE model is able to learn an effective visual-semantic embedding space for image manipulation. The image decoder trained with images from Conceptual Captions dataset learns a general image prior for open-domain images.

The model structure and training process of the VSE model follow that of VSE++ [15]. The image decoder takes $1024 \times 7 \times 7$ feature maps as input, and is composed of 7 ResNet Blocks with upsampling layers in between, which generates 256×256 images. The discriminator is a Multi-scale Patch-based discriminator following [32]. The decoder is trained with GAN loss, perceptual loss, and discriminator feature matching loss. The edge extractor is an off-the-shelf bi-directional cascade network [17] trained on BSDS500 dataset [3].

4.2 Applications and Results

Our approach can achieve open-domain image manipulation with open-vocabulary instructions, which has various applications. We demonstrate several examples in Fig. 4, including changing color, texture, and global or local high-level attributes.

Results in the first row demonstrate that our model is able to change *color* for objects and stuff while preserving other details of the image. Moreover, it preserves the lighting conditions and relative color strengths very well when changing colors. Our model is also able to change *textures* of the images with language instructions, for example, editing object materials or changing sea to grass, as shown in the second row. Results indicate that the VSE model learns effective texture features in the joint embedding space, and that the generator is able to generate reasonable textures based on the manipulated features. Besides low-level attributes, our model is also able to handle *high-level semantic attributes*, such as removing lights, changing photos to paintings, sunny to cloudy, and transferring seasons, in the third and fourth rows.

Quantitative evaluation. Since ground-truth manipulated images are not available, we conduct evaluations by user study, L2 error, and LPIPS.

The user study is conducted to evaluate human perceptions of our approach. For each experiment, we randomly pick 60 images and manually choose the appropriate manipulation instructions for them. The images cover a wide variety of styles and the instructions range from color and texture manipulation to high-level attribute manipulation. 10 users are asked to score the 60 images for each

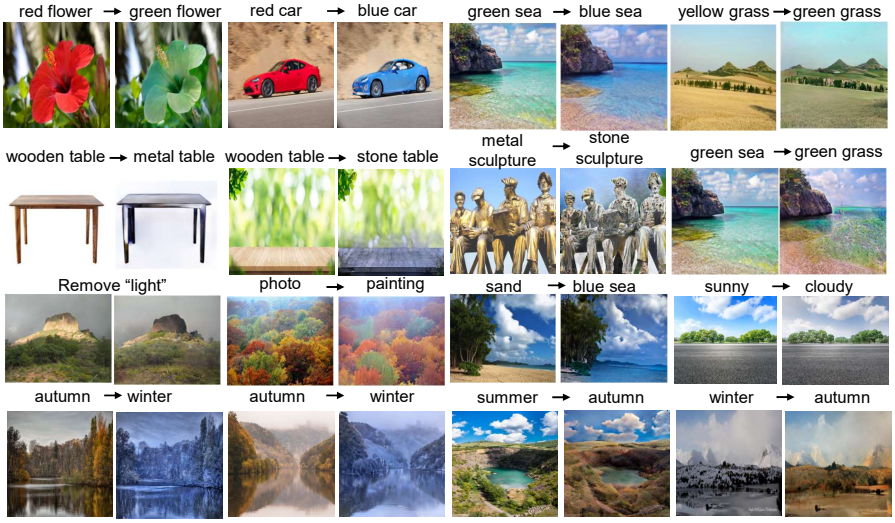

Fig. 4: Applications and manipulation results of our method.

experiment by three criteria, (1) visual quality, (2) how well the manipulated images preserve the details of the source image, and (3) how well the manipulation results match the instruction. The scores range from 1 (worst) to 5 (best), and we will analyze the results shown in Table. 1 in the following.

To evaluate the visual quality and content preservation, we calculate the L2 error and Perceptual similarity (LPIPS) [46] between the reconstructed images and input images, as shown in Table 2 and analyzed in the following.

Comparison with previous work. Since this is the first work to explore open-domain image manipulation with open-vocabulary instructions, our problem setting is much more challenging than previous approaches. Nonetheless, we compare with two representative approaches, CycleGAN [49] and TAGAN [31].

CycleGAN is designed for image-to-image translation, but we have to define domains and collect domain-specific images for training. So it is not able to tackle open-vocabulary instructions. To compare with CycleGAN, we train three CycleGAN models for translating between blue and red objects, translating between red and green objects, and translating between beach and sea, respectively. The images for training CycleGAN are retrieved from Conceptual Captions with our visual-semantic embedding model. Qualitative comparison are shown in Fig. 5, and user study are shown in Table 1. Results indicate that both our approach and CycleGAN is able to preserve details of the input images very well, but CycleGAN worse at transferring desired attributes in some scenarios.

State-of-the-art language-based image manipulation method TAGAN [31] uses mismatched image-caption pairs as training samples for manipulation. It is able to handle language instructions, but is limited to only one specific domain

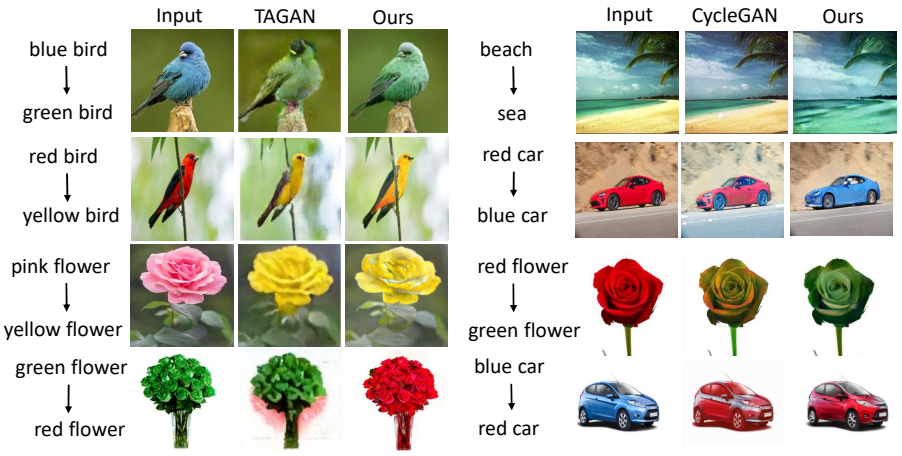

Fig. 5: Comparison between with previous language-based image editing method TAGAN [31] (left) and image-to-image translation method CycleGAN [49] (right).

such as flowers (Oxford-102) or birds (CUB). It also requires fine-grained human-annotated descriptions of each image in the dataset. While our approach handles open-domain images with noisy captions harvested from the web for training. Since TAGAN only has the models for manipulating bird or flower images, we compare our results with theirs on flower and bird image manipulation in Fig. 5. We also compare user evaluation in Table 1. Quantitative evaluation of L2 error and perceptual metric (LPIPS) between reconstructed images and original images are shown in Table 2. Our model is not trained on the Oxford-102 or CUB datasets, but still performs better than the TAGAN models specifically trained on those datasets. Moreover, the L2 reconstruction error also shows that our model has the potential to preserve the detailed contents of the source images.

4.3 Component Analysis

The effectiveness of instruction grounding. Our text-guided visual feature manipulation module uses the soft instruction grounding maps as location-adaptive manipulation coefficients to control the local manipulation strength at different locations. The instruction grounding map is very important when the manipulation instruction is related to local areas or objects in the image. Fig. 6(a) demonstrates the effectiveness of adopting grounding for manipulation, where we aim to change the green apple into a yellow apple and keep the red apple unchanged. The grounding map is able to roughly locate the green apple, and with the help of the soft grounding map, the model is able to change the color of the green apple while keeping the red apple and the background unchanged. On the contrary, the model without grounding changes not only the green apple, but also the red apple and the background.

Table 1: User study results on visual quality, how well the manipulated images preserve details, and how well the manipulated images match the instruction. In the table, "edge" represents edge constraints in the image decoder, and "opt" represents the sample-specific optimization.

	CycleGAN[49]	TAGAN[31]	w/o edge, w/o opt	w/ edge, w/o opt	**w/ edge, w/ opt**
Visual quality	4.0	3.1	1.3	4.1	**4.4**
Preserve details	4.2	2.7	1.2	3.7	**4.3**
Match instruction	1.9	4.2	4.0	4.5	**4.5**

Table 2: L2 error and LPIPS between reconstructed and original images of TAGAN and ablations of our approach. Lower L2 reconstruction error and LPIPS metric indicates that the reconstructed images preserve details of the source images better.

	TAGAN[31]	w/o edge, w/o opt	w/ edge, w/o opt	**w/ edge, w/ opt**
L2 error on Oxford-102 test set	0.11	0.19	0.10	**0.05**
L2 on Conceptual Captions val	N/A	0.20	0.12	**0.07**
LPIPS on Conceptual Captions val	N/A	0.33	0.17	**0.06**

Edge Constraints. Our structure-aware image decoder exploits edge constraints to preserve the structure information when generating the reconstructed and manipulated images. In Fig. 6(b), we show an example of image reconstruction and manipulation with and without edges. The image decoder is able to reconstruct and generate higher-quality images with clear structures and edges with edge constraints. User study results in Table 1 and quantitative evaluation in Table 2 also indicate that the generated images are of better visual quality and preserve details better with the structure-aware image decoder.

Adjusting coefficient α for smooth attribute transition. The hyper-parameter α controls the global attribute manipulation strength, which can be adjusted according to user requirements. By gradually increasing α, we obtain a smooth transition from the source images to the manipulated images with different manipulation strengths. Fig. 6(c)(d) illustrates the smooth transition of an image from dark to bright, and from red apple to green apple, respectively.

The effectiveness of sample-specific optimization and cycle-consistency constraints. Fig. 6(e)[4] demonstrates the effectiveness of sample-specific optimization and cycle-consistency constraints. The reconstructed image and manipulated image without sample-specific optimization miss some details such as the shape of the glasses. With the perturbation optimization by reconstruction loss, our model is able to generate better-quality reconstructed and manipulated images. Optimizing the perturbations with both reconstruction loss and manipulation cycle-consistency loss further improves the quality of the generated images, *e.g.*, the glasses are more realistic and the person identity appearance is better preserved. User study in Table 1 and quantitative evaluation in Table 2 indicate that the sample-specific optimization has the potential of enhancing details of the generated images.

[4] The decoder for Fig. 6(e) is trained on FFHQ [20] to learn the face image prior.

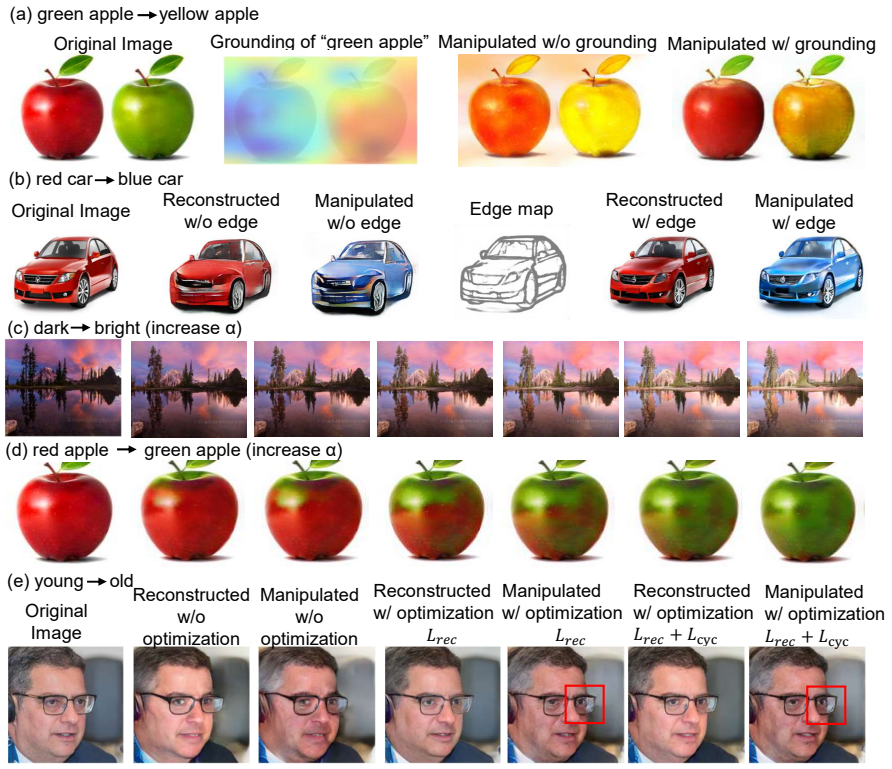

Fig. 6: Component analysis of our approach. The examples from top to bottom show analysis on grounding, edge constraints, adjusting coefficient, and the sample-specific optimization and cycle-consistency constraints, respectively.

5 Conclusion and Discussions

We propose Open-Edit, the first framework for open-vocabulary open-domain image manipulation with minimal training supervision. It takes advantage of the pretrained visual-semantic embedding, and manipulates visual features by vector arithmetic with textual embeddings in the joint embedding space. The sample-specific optimization further regularizes the manipulated images and encourages realistic and detail-preserving results. Impressive color, texture, and semantic attribute manipulation are shown on various types of images.

We believe that this is a challenging and promising direction towards more general and practical image editing, and that our attempt would inspire future work to enhance editing qualities and extend the application scenario. In this work we focus on editing appearance-related attributes without changing the structure of images. Further work can be done on more challenging structure-related editing and image editing with more complex sentence instructions.

References

1. Ak, K.E., Lim, J.H., Tham, J.Y., Kassim, A.A.: Attribute manipulation generative adversarial networks for fashion images. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10541–10550 (2019)
2. Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A.: Augmented cyclegan: Learning many-to-many mappings from unpaired data. arXiv preprint arXiv:1802.10151 (2018)
3. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence **33**(5), 898–916 (2010)
4. Bau, D., Strobelt, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J.Y., Torralba, A.: Semantic photo manipulation with a generative image prior. ACM Transactions on Graphics (TOG) **38**(4), 59 (2019)
5. Bau, D., Zhu, J.Y., Strobelt, H., Zhou, B., Tenenbaum, J.B., Freeman, W.T., Torralba, A.: Gan dissection: Visualizing and understanding generative adversarial networks. arXiv preprint arXiv:1811.10597 (2018)
6. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Neural photo editing with introspective adversarial networks. arXiv preprint arXiv:1609.07093 (2016)
7. Chen, D., Li, H., Liu, X., Shen, Y., Shao, J., Yuan, Z., Wang, X.: Improving deep visual representation for person re-identification by global and local image-language association. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 54–70 (2018)
8. Chen, J., Shen, Y., Gao, J., Liu, J., Liu, X.: Language-based image editing with recurrent attentive models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8721–8729 (2018)
9. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1511–1520 (2017)
10. Chen, Y.C., Shen, X., Lin, Z., Lu, X., Pao, I., Jia, J., et al.: Semantic component decomposition for face attribute manipulation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9859–9867 (2019)
11. Cheng, Y., Gan, Z., Li, Y., Liu, J., Gao, J.: Sequential attention gan for interactive image editing via dialogue. arXiv preprint arXiv:1812.08352 (2018)
12. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8789–8797 (2018)
13. Dong, H., Yu, S., Wu, C., Guo, Y.: Semantic image synthesis via adversarial learning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5706–5714 (2017)
14. El-Nouby, A., Sharma, S., Schulz, H., Hjelm, D., Asri, L.E., Kahou, S.E., Bengio, Y., Taylor, G.W.: Keep drawing it: Iterative language-based image generation and editing. arXiv preprint arXiv:1811.09845 (2018)
15. Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: Vse++: Improving visual-semantic embeddings with hard negatives. arXiv preprint arXiv:1707.05612 (2017)
16. Günel, M., Erdem, E., Erdem, A.: Language guided fashion image manipulation with feature-wise transformations. arXiv preprint arXiv:1808.04000 (2018)
17. He, J., Zhang, S., Yang, M., Shan, Y., Huang, T.: Bi-directional cascade network for perceptual edge detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3828–3837 (2019)

18. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 172–189 (2018)
19. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
20. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948 (2018)
21. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain relations with generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1857–1865. JMLR. org (2017)
22. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 35–51 (2018)
23. Li, B., Qi, X., Lukasiewicz, T., Torr, P.H.: Manigan: Text-guided image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7880–7889 (2020)
24. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: Advances in neural information processing systems. pp. 469–477 (2016)
25. Liu, X., Li, H., Shao, J., Chen, D., Wang, X.: Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 338–354 (2018)
26. Liu, X., Wang, Z., Shao, J., Wang, X., Li, H.: Improving referring expression grounding with cross-modal attention-guided erasing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1950–1959 (2019)
27. Liu, X., Yin, G., Shao, J., Wang, X., Li, H.: Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In: Advances in Neural Information Processing Systems. pp. 570–580 (2019)
28. Mao, X., Chen, Y., Li, Y., Xiong, T., He, Y., Xue, H.: Bilinear representation for language-based image editing using conditional generative adversarial networks. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2047–2051. IEEE (2019)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
30. Mo, S., Cho, M., Shin, J.: Instagan: Instance-aware image-to-image translation. arXiv preprint arXiv:1812.10889 (2018)
31. Nam, S., Kim, Y., Kim, S.J.: Text-adaptive generative adversarial networks: manipulating images with natural language. In: Advances in Neural Information Processing Systems. pp. 42–51 (2018)
32. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. arXiv preprint arXiv:1903.07291 (2019)
33. Perarnau, G., Van De Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355 (2016)
34. Royer, A., Bousmalis, K., Gouws, S., Bertsch, F., Mosseri, I., Cole, F., Murphy, K.: Xgan: Unsupervised image-to-image translation for many-to-many mappings. arXiv preprint arXiv:1711.05139 (2017)
35. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of ACL (2018)

36. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. arXiv preprint arXiv:1907.10786 (2019)
37. Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., Samaras, D.: Neural face editing with intrinsic image disentangling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5541–5550 (2017)
38. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. arXiv preprint arXiv:1611.02200 (2016)
39. Usman, B., Dufour, N., Saenko, K., Bregler, C.: Puppetgan: Cross-domain image manipulation by demonstration. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9450–9458 (2019)
40. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3156–3164 (2015)
41. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8798–8807 (2018)
42. Wang, Z., Liu, X., Li, H., Sheng, L., Yan, J., Wang, X., Shao, J.: Camp: Cross-modal adaptive message passing for text-image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5764–5773 (2019)
43. Xiao, T., Hong, J., Ma, J.: Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 168–184 (2018)
44. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. In: Proceedings of the IEEE international conference on computer vision. pp. 2849–2857 (2017)
45. Yu, X., Chen, Y., Liu, S., Li, T., Li, G.: Multi-mapping image-to-image translation via learning disentanglement. In: Advances in Neural Information Processing Systems. pp. 2994–3004 (2019)
46. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
47. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)
48. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision. pp. 597–613. Springer (2016)
49. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)
50. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: Advances in Neural Information Processing Systems. pp. 465–476 (2017)
51. Zhu, S., Urtasun, R., Fidler, S., Lin, D., Change Loy, C.: Be your own prada: Fashion synthesis with structural coherence. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1680–1688 (2017)