EDIT: Exemplar-Domain Aware Image-to-Image Translation

Yuanbin Fu
Tianjin University
fu199671@126.com

Jiayi Ma
Wuhan University
jyma2010@gmail.com

Lin Ma
Tencent AI Lab
forestlma@tencent.com

Xiaojie Guo
Tianjin University
xj.max.guo@gmail.com

Abstract

Image-to-image translation is to convert an image of the certain style to another of the target style with the content preserved. A desired translator should be capable to generate diverse results in a controllable (many-to-many) fashion. To this end, we design a novel generative adversarial network, namely exemplar-domain aware image-to-image translator (EDIT for short). The principle behind is that, for images from multiple domains, the content features can be obtained by a uniform extractor, while (re-)stylization is achieved by mapping the extracted features specifically to different purposes (domains and exemplars). The generator of our EDIT comprises of a part of blocks configured by shared parameters, and the rest by varied parameters exported by an exemplar-domain aware parameter network. In addition, a discriminator is equipped during the training phase to guarantee the output satisfying the distribution of the target domain. Our EDIT can flexibly and effectively work on multiple domains and arbitrary exemplars in a unified neat model. We conduct experiments to show the efficacy of our design, and reveal its advances over other state-of-the-art methods both quantitatively and qualitatively.

1. Introduction

A scene can be expressed in various manners using sketches, semantic maps, photographs, and painting artworks, to name just a few. Basically, the way that one portrays the scene and expresses his/her vision is so-called style, which can reflect the characteristic of either a class/domain or a specific case. Image-to-image translation (I2IT) [10][25][1][36][29] refers to the process of converting an image I of the certain style to another of the target style S_t with the content preserved. Formally, seeking a desired translator $\mathcal{T}$ can be written in the following form:

$$\min\ \mathcal{C}(I_t, I) + \mathcal{S}(I_t, S_t)\ \text{ with }\ I_t := \mathcal{T}(I, S_t), \qquad (1)$$

where $\mathcal{C}(I_t, I)$ is to measure the content difference between the translated I_t and the original I, while $\mathcal{S}(I_t, S_t)$ is to enforce the style of I_t following that indicated by S_t.

1.1. Previous Arts

With the emergence of deep techniques, a variety of I2IT strategies have been proposed with great progress made over the last decade. In what follows, we briefly review contemporary works along two main technical lines, *i.e.* one-to-one translation and many-to-many translation.

One-to-one Translation. Methods in this category aim at mapping images from a source domain to a target domain. Benefiting from the generative adversarial networks (GANs) [11], the style of translated results satisfies the distribution of the target domain $\mathbb{Y}$, achieved by $\mathcal{S}(I_t, S_t) := \mathcal{D}(I_t, \mathbb{Y})$, where $\mathcal{D}(I_t, \mathbb{Y})$ represents a discriminator to distinguish if I_t is real with respect to $\mathbb{Y}$. An early attempt by Isola *et al.* [17] uses conditional GANs to learn mappings between two domains. The content preservation is supervised by the paired data, *i.e.* $\mathcal{C}(I_t, I) := \mathcal{C}(I_t, I_t^{gt})$ with I_t^{gt} the ground-truth target. However, in real-world situations, acquiring such paired datasets, if not impossible, is impractical. To alleviate the pressure from data, inspired by the concept of cycle consistency, cycleGAN [37] in an unsupervised fashion was proposed, which adopts $\mathcal{C}(I_t, I) := \mathcal{C}(\mathcal{F}_{\mathbb{Y}\to\mathbb{X}}(\mathcal{F}_{\mathbb{X}\to\mathbb{Y}}(I)), I)$ with $\mathcal{F}_{\mathbb{X}\to\mathbb{Y}}$ the generator from domain $\mathbb{X}$ to $\mathbb{Y}$ and $\mathcal{F}_{\mathbb{Y}\to\mathbb{X}}$ the reverse one. Afterwards, StarGAN [8] further extends the translation between two domains to that cross multiple domains in a single model. Though the effectiveness of the mentioned methods has been witnessed by a wide spectrum of specific applications such as photo-caricature [3, 7], making up-makeup removal [4], and face manipulation [35], their main drawback comes from the nature of deterministic (uncontrollable) one-to-one mapping.

Many-to-many Translation. The goal of approaches in this class is to transfer the style controlled by an exemplar image to a source image with content maintained. Arguably, the most representative work goes to [9], which uses the pre-trained VGG16 network [33] to extract the content and style features, then transfer style information by minimizing the distance between Gram matrices constructed from the generated image and the exemplar E, say $\mathcal{S}(I_t, S_t) := \mathcal{S}(\text{Gram}(I_t), \text{Gram}(E))$. Since then, numerous applications on 3D scene [6], face swap [19], portrait

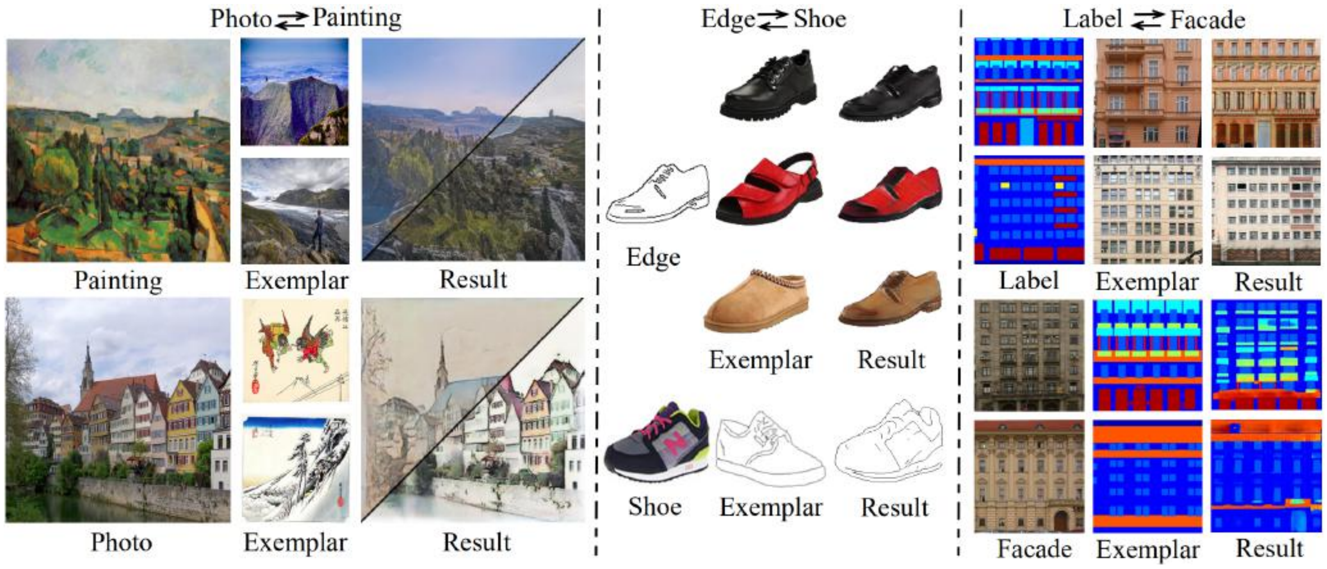

Figure 1: Several results by the proposed EDIT. Our EDIT is able to take arbitrary exemplars as reference for translating images across multiple domains including photo-painting, shoe-edge, and semantic map-facade in *one* model.

stylization [31] and font design [2] have been done. Furthermore, a number of schemes have also been developed towards relieving the limitations of [9] in terms of speed and flexibility. For example, to tackle the requirement of training for every new exemplar (style), Shen *et al.* [28] built a meta network, which takes in the style image and produces a corresponding image transformation network directly. Risser *et al.* [26] proposed the histogram loss to improve the training instability. Huang and Belongie [15] designed a more suitable normalization manner, *i.e.* AdaIN, for style transfer. Li *et al.* [22] replaced the Gram matrices with an alternative distribution alignment manner from the perspective of domain adaption. Johnson *et al.* [18] trained the network with a specific style image and multiple content images while keeping the parameters at the inference stage. Chen *et al.* [5] introduced a style-bank layer containing several filter-banks, each of which represents a specific style. Gu *et al.* [12] proposed a style loss to make parameterized and non-parameterized methods complement to each other. Huang *et al.* [14] designed a new temporal loss to ensure the style consistency between frames of a video. In addition, to mitigate the deterministic nature of one-to-one translation, several works, for instance [20], [23] and [16], advocated to separately take care of content $c(I)$ and style $s(I)$ subject to $I \simeq c(I) \circ s(I)$ with $\circ$ the combination operation. They manage to control the translated results by combining the content of an image with the style of target, *i.e.* $c(I) \circ s(E)$. Besides one domain pair requires one independent model, their performance, as observed from comparisons, is inferior to our method in visual quality, diversity, and style preservation. Please see Fig. 1 for examples produced by our method that handles multiple domains and arbitrary exemplars in a unified model.

1.2. Challenges & Motivations

Developing a practical I2I translator remains challenging, because the capabilities of preserving content information after translation, and handling multiple domains as well as arbitrary exemplars should be considered jointly. We list the challenges as follows:

- How to rationally disentangle the content and style representations of images from different domains in a unified fashion (multi-domain in one model)?
- How to effectively ensure the content of the translated result being consistent with that of the original image in an unsupervised manner (content preservation)?
- How to flexibly manipulate an image by considering both the style of a target domain and that of a specific exemplar (exemplar-domain style awareness)?

Our principle is that, for images from different domains, the content features can be obtained by a uniform extractor, while (re-)stylization is achieved by mapping the extracted features specifically to different purposes. This principle is rational: taking artwork composition for an example, given a fixed scene, the physical content is the same, but the styles of presentation can be much diverse by different artists. For the style factor, one may generally like the paintings by Monet (domain), and among so many pieces of art, a particular one, *e.g.* “Water Lilies” (exemplar), is his/her favorite. In other words, the domain-level and exemplar-level should

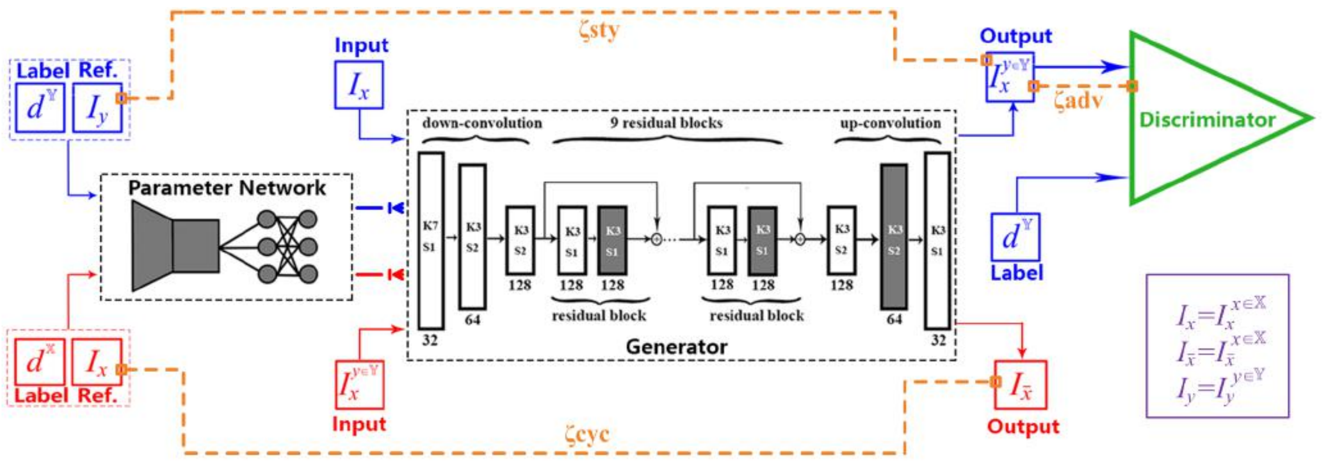

Figure 2: The model architecture of our EDIT. The procedure of mapping $\mathbb{X} \rightarrow \mathbb{Y}$ is in blue, while the reverse of mapping $\mathbb{Y} \rightarrow \mathbb{X}$ is in red. I_x and I_y are samples from domain $\mathbb{X}$ and $\mathbb{Y}$, respectively. The whole network comprises of a generator and a discriminator. The generator contains a part of blocks configured by shared parameters, and the rest by varied parameters exported by an exemplar-domain aware parameter network. The parameter network generates the specific parameters based on an exemplar and its domain label. The content is preserved by adopting the cycle consistency. The discriminator takes a generated result and its domain label as input to judge if the result is distinguishable from the target domain. K_k means that the kernel size is $k \times k$, while S_s represents that the stride is s. The number of channels is given below each block.

be simultaneously concerned during style transfer. Moreover, to maintain the content information after translation, the cycle consistency can be employed due to its effectiveness and simplicity. It is worth emphasizing that, a single generator instead of a pair, like cycleGAN [37], could be sufficient if the content and style are well-disentangled.

1.3. Contributions

Motivated by the above principle, we propose a novel network to overcome the mentioned challenges. Concretely, our primary contributions can be summarized as follows:

- We design a network, namely EDIT, to produce diverse results in an unsupervised controllable (many-to-many) fashion, which can flexibly and effectively work on multiple domains and arbitrary exemplars in a unified model.
- The generator of our EDIT comprises of a part of blocks configured by shared parameters to uniformly extract features for images from multiple domains, and the rest by varied parameters exported by an exemplar-domain aware parameter network to catch specific style information.
- To preserve the content between input and generated result, the cycle consistency is employed. Plus, a discriminator is equipped during the training phase to guarantee the output satisfying the distribution of the target domain.
- We conduct extensive experimental results to reveal the efficacy of our design, and demonstrate its advantages over other state-of-the-art methods both quantitatively and qualitatively.

Several previous works, with feature transform in [21], style decoration in [30], and feature normalization transfer in [24] as representatives, insert an extra step, say feature manipulation, **between** the trained encoder and decoder to achieve style transfer, the spirit of which is seemingly similar but much different to ours. We achieve the style transfer **in** the decoder dynamically generated. Even if there is no exemplar provided, the model still can produce results according to the target domain (by setting the exemplar to *e.g.* a black image), which is more flexible than traditional style transfer methods like [21, 30] having no domain information considered. Please notice that although the method [24] considers both the domain and exemplar knowledge, it requires to train different models for different domain pairs, while our method is able to embrace multiple domain pairs in one neat model.

2. Methodology

2.1. Problem Analysis

A desired translator should be capable to generate diverse results in a controllable (many-to-many) fashion. Again, we emphasize the core principle behind this work: *for images from different domains, the content features can* *be obtained by a uniform extractor, while (re-)stylization is achieved by mapping the extracted features specifically to different purposes*. In other words, we assume that the content $c(\cdot)$ and the style $s(\cdot)$ of an image are independent, *i.e.* $p(I) = p(c(I), s(I)) = p(c(I)) \cdot p(s(I))$. Suppose that the whole style space is $\bigcup_i \mathbb{S}_i$, where $\mathbb{S}_i$ is the style subspace corresponding to the domain i. Mathematically, the problem can be expressed and solved by maximizing the following probability:

$$\begin{aligned} p(I_x^y|I_x, I_y) &:= p(c(I_x^y), s(I_x^y)|c(I_x), s(I_x), c(I_y), s(I_y)) \\ &\propto p(c(I_x^y)|c(I_x)) \cdot p(s(I_x^y)|s(I_y)) \\ &= p(c(I_x^y)|c(I_x)) \cdot \sum_i p(s(I_x^y)|s(I_y), \mathbb{S}_i) \cdot p(\mathbb{S}_i) \\ &= \sum_i p(c(I_x^y)|c(I_x)) \cdot p(s(I_x^y)|s(I_y), \mathbb{S}_i) \cdot p(\mathbb{S}_i). \end{aligned} \tag{2}$$

The relationship of second row holds by the problem definition in Eq. (1) and the independence assumption (our core principle). Furthermore, the style of I_y may appear in more than one domains, for instance, a semantic map can also be a painting. This situation makes $p(s(I_x^y)|s(I_y))$ a mixture of $\sum_i p(s(I_x^y)|s(I_y), \mathbb{S}_i) \cdot p(\mathbb{S}_i)$ (the equality of 3rd row). Please see Figure 3 for evidence. Therefore, we specify the domain label to clear the mix-up. By doing so, the problem turns to maximize the following:

$$\begin{aligned} &p(I_x^{y\in\mathbb{S}_i}|I_x, I_y \in \mathbb{S}_i) \\ &\quad := p(c(I_x^y)|c(I_x)) \cdot p(s(I_x^y)|s(I_y), \mathbb{S}_i). \end{aligned} \tag{3}$$

As given in Eq. (3), the entire problem can thus be divided into two subproblems. The first component $p(c(I_x^y)|c(I_x))$ corresponds to the uniform content extractor, while the second term $p(s(I_x^y)|s(I_y), \mathbb{S}_i)$ yields the exemplar-domain aware style mapping.

2.2. Architecture Design

The blueprint of our EDIT is schematically illustrated in Figure 2, from which, we can see that the generator of EDIT $\mathcal{G}$ is composed by a part of blocks configured by shared parameters θ_s, and the rest by varied parameters θ_p exported by an exemplar-domain aware parameter auxiliary network. In addition, a discriminator $\mathcal{D}$ is equipped during the training phase to guarantee the output satisfying the distribution of the target domain.

The **generator** is to produce desired images through

$$I_x^{y\in\mathbb{S}_i} := \mathcal{G}(I_x, I_y \in \mathbb{S}_i; \theta), \tag{4}$$

where θ is the trainable parameters for the whole generator. The generator consists of three gradually down-sampled encoding blocks, followed by 9 residual blocks. Then, the decoder processes the feature maps gradually up-sampled to

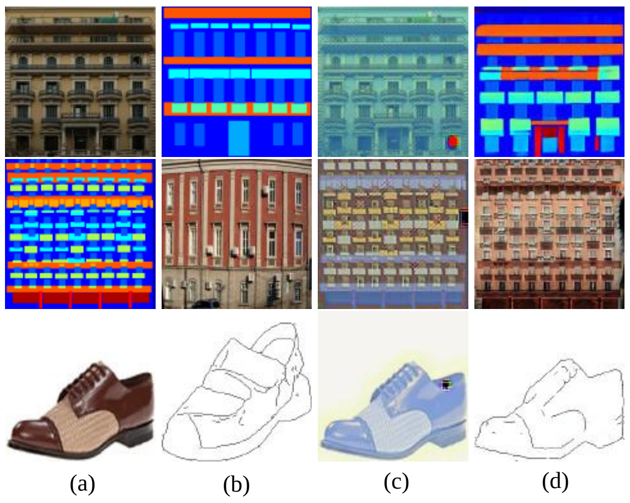

Figure 3: Visual results by EDIT with and without specifying the target domain. (a) and (b) contain the inputs I_x and exemplars I_y, respectively. (c) and (g) give the translated results without and with domain specification, respectively.

the same size as input. Each block performs in the manner of *Conv+InstanceNorm+ReLU*. As stated, a part of the generator should respond to extract features uniformly for images no matter what styles they are in. In other words, a number of blocks (in white as shown in Fig. 2, **uniform content extractor**) are shared across domains, the parameter set of which is denoted by θ_s. As for the rest blocks (in black as shown in Fig. 2) related to (re)-stylization (feature selection and reassembling). Inspired by [13, 28], the corresponding parameters can be dynamically generated by a parameter network, that is:

$$\theta_p := \mathcal{G}_P(I_y \in \mathbb{S}_i; \psi), \tag{5}$$

where ψ is its trainable parameters. Please notice that, our parameter network only covers a part, instead of all as [13, 28], of blocks in the generator, which significantly saves the resource. Specifically, the **parameter network** contains the *VGG16* network pre-trained on the ImageNet and fixed, followed by *one fully connected layer and one group fully connected layer*. Feeding an exemplar (style image) and its target domain label (a one-hot vector) into the parameter network gives the parameters required by the **exemplar-domain aware style mapping**. Now, we can express the generator as follows:

$$I_x^{y\in\mathbb{S}_i} := \mathcal{G}(I_x; \theta_p := \mathcal{G}_P(I_y \in \mathbb{S}_i; \psi), \theta_s), \tag{6}$$

where both θ_s and θ_p form θ.

Based on the analysis on domain specification, it is important to clear the style mix-up issue as revealed in Fig. 3. Merely providing the domain ID to the parameter network is insufficient to capture the domain characteristic, as it is blind to the distribution of the target domain. To guide the training process and produce high-quality images satisfying the distribution of the target domain, we further employ a **discriminator** built upon the 70×70 Patch-GAN architecture [17], which tries to determine whether each local image patch , rather than the whole image, is real or fake. More details about the discriminator can be found in the corresponding paper or at our website*. It is worth noting that the exemplar-domain aware style mapping is actually achieved by the dynamic part in the generator together with the discriminator.

One may wonder why inserting dynamic (black) blocks into fixed (white) blocks. First, considering the generation of dynamic parameters, the complexity of the fully connected layers will dramatically grow as the number of dynamic parameters (*e.g.* all the blocks in the decoder) required to generate increases. From another point of view, the style mapping can be viewed as a procedure of feature selection and reassembling. Some operations should be in common for features from different domains. Taking the above concerns, we adopt the organization fashion as shown in Fig. 2, which performs sufficiently well in practice and makes the volume of the parameter network compact. The primary merit of our EDIT is that it can handle arbitrary exemplars and be trained for multiple domains at the same time in *one* neat model. More details of EDIT are given in supplementary.

2.3. Loss Design

We adopt a combination of a cycle consistency loss, a style loss and an adversarial loss for training the network.
Cycle consistency loss. Taking a sample pair $I_x \in \mathbb{X}$ and $I_y \in \mathbb{Y}$ for an example, let $I_{\bar{x}}$ and $I_{\bar{y}}$ be $\mathcal{G}(\mathcal{G}(I_x, I_y \in \mathbb{Y}), I_x \in \mathbb{X})$ and $\mathcal{G}(\mathcal{G}(I_y, I_x \in \mathbb{X}), I_y \in \mathbb{Y})$, respectively. To preserve content between generated and original images, the cycle consistency loss is employed, which is written as:

$$\zeta_{cyc} := \|I_{\bar{x}} - I_x\|_1 + \|I_{\bar{y}} - I_y\|_1, \tag{7}$$

where $\|\cdot\|_1$ is the ℓ_1 norm.
Style loss. For allowing users to control the style by giving an exemplar, a measurement for style difference is required. As advocated in [22], the batch normalization statistics based loss is adopted instead of the Gram matrix based one, for ease of computation. By denoting $I_{\hat{x}} := \mathcal{G}(I_x, I_y \in \mathbb{Y})$ and $I_{\hat{y}} := \mathcal{G}(I_y, I_x \in \mathbb{X})$, we have:

$$\begin{aligned}\zeta_{sty} :=& \sum_{l=1}^{N_L}\sum_{m=1}^{M_l} \frac{\left((\mu_{\hat{y}}^{l,m} - \mu_{x}^{l,m})^2 + (\sigma_{\hat{y}}^{l,m} - \sigma_{x}^{l,m})^2\right)}{N_L \times M_l} \\ &+ \sum_{l=1}^{N_L}\sum_{m=1}^{M_l} \frac{\left((\mu_{\hat{x}}^{l,m} - \mu_{y}^{l,m})^2 + (\sigma_{\hat{x}}^{l,m} - \sigma_{y}^{l,m})^2\right)}{N_L \times M_l},\end{aligned} \tag{8}$$

where N_L and M_l are the number of involved layers (in this work, we use the $relu1\_2$, $relu2\_2$, $relu3\_3$, $relu4\_3$ and $relu5\_1$ layers in the VGG16) and that of feature maps in the l-th layer. In addition, μ and σ are the mean and the standard deviation of the corresponding feature map.
Adversarial loss. The adversarial loss is standard [11] as:

$$\begin{aligned}\zeta_{adv} :=& \log \mathcal{D}(I_x, \mathbb{X}) + \log(1 - \mathcal{D}(\mathcal{G}(I_x, I_y \in \mathbb{Y}), \mathbb{Y})) \\ &+ \log \mathcal{D}(I_y, \mathbb{Y}) + \log(1 - \mathcal{D}(\mathcal{G}(I_y, I_x \in \mathbb{X}), \mathbb{X})).\end{aligned} \tag{9}$$

Final objective. Our optimization is carried out on the total loss, *i.e.* the sum of the above losses, as follows:

$$\begin{aligned}&\min_{\mathcal{G}} \max_{\mathcal{D}} \ \mathbb{E}_{I_x \sim P_{data}(\mathbb{X})} \mathbb{E}_{I_y \sim P_{data}(\mathbb{Y})} \ \zeta_{total}, \\ &\text{where } \ \zeta_{total} := \zeta_{adv} + \lambda \zeta_{cyc} + \eta \zeta_{sty},\end{aligned} \tag{10}$$

where η and λ are coefficients to balance the loss terms. In order to keep the common features effectively, we set λ to a relatively large value 10. As for η, we observe that setting it in the range from 0.01 to 0.1 works well.

3. Experimental Validation

Implementation details. Our EDIT is implemented in PyTorch and performed on a GeForce RTX 2080 GPU. We use the strategy proposed in [32] to improve the training stability, which uses historic generated images to update the parameters of the discriminator. Our optimization adopts an Adam solver. The decays of the first and second order momentums are as default. The learning rate is set to 0.001 at the beginning and linearly decreased as the number of epochs grows. During the training phase, the input images are resized to 256×256 and augmented by random horizontal flip.

Competitors & Evaluation metrics. The competitors involved in comparisons contain neural style transfer (NST) [9], cycleGAN [37], metaNST [28], DRIT [20], MUNIT [16], WCT [21], EGSC-IT [24], and art2real [34]. The codes of the compared methods are all downloaded from the authors' websites. The elapsed time of testing, model size, and inception score [27] are employed as our metrics to quantitatively reveal the performance difference between our EDIT with the other competitors. In addition, to measure how well the content and style are preserved, by following [9], the content error and style error are also adopted. Take the mapping: $\mathbb{X} \to \mathbb{Y}$ as an example, the content error $\mathcal{E}_{cont}$ is defined as the L2 distance between feature maps of the input image I_x and the generated one $I_x^{y\in\mathbb{Y}}$, *i.e.* $||\phi_l(I_x) - \phi_l(I_x^{y\in\mathbb{Y}})||_2$, where $\phi_l(\cdot)$ means the feature maps of the l-th layer in the VGG-16 model. The style error is the average L2 distance between the Gram matrices of the generated image $I_x^{y\in\mathbb{Y}}$ and the exemplar I_y. Assume $\text{Gram}_l(\cdot)$, H_l, and W_l are

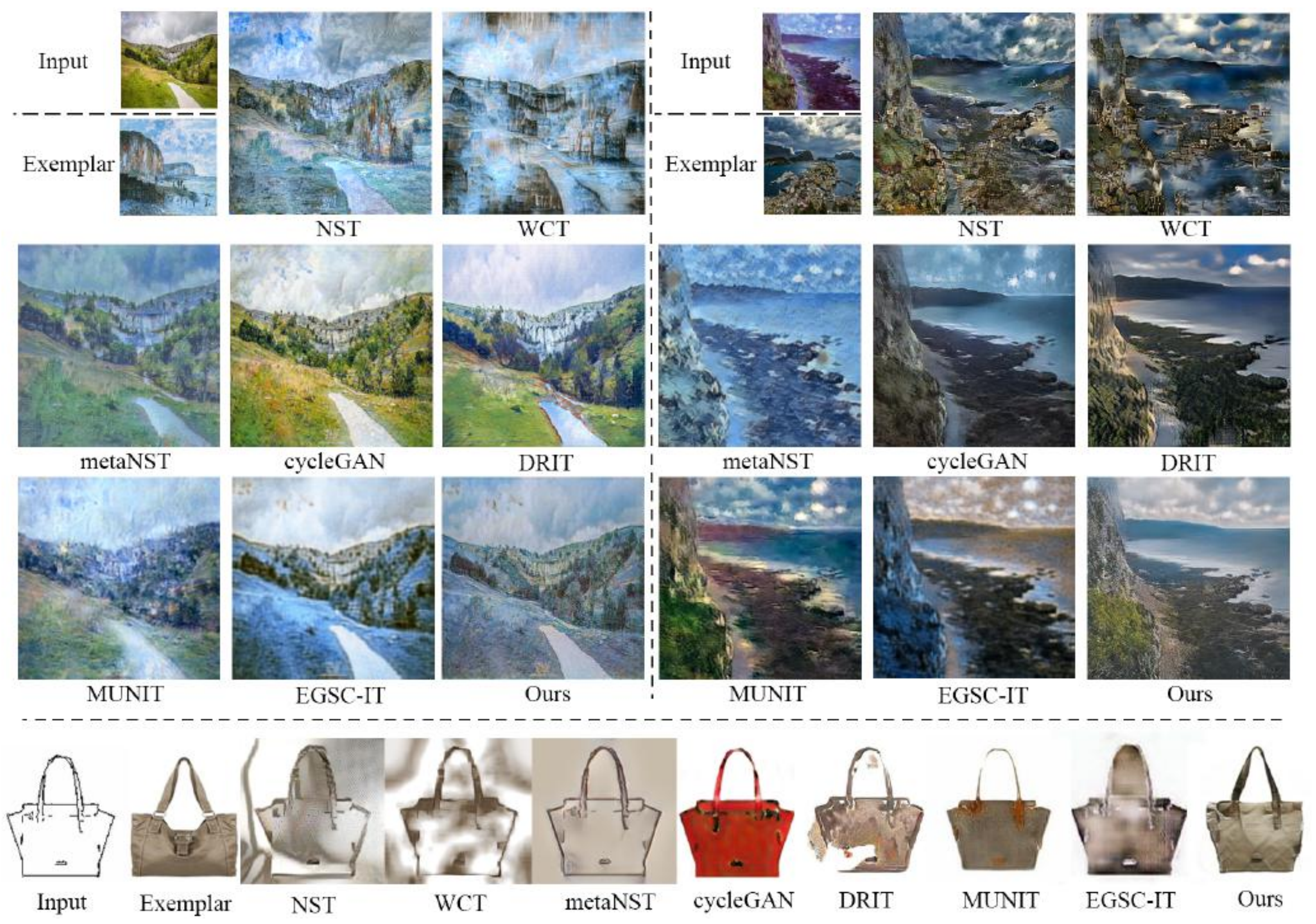

Figure 4: Visual comparison among the competitors on photo to painting, painting to photo, and edge to handbag.

Methods	Time (sec.)	Paras. (MB|1 pair)	($n = 4$ pairs)	Content Error ↓	Style Error ↓	IncepScore ↑
art2real [34]	1.2×10^{-2}	45.5	–	–	–	–
cycleGAN [37]	3.5×10^{-3}	45+45	(45+45)$\times n$	1.70±0.60	–	6.02
MUNIT [16]	3.9×10^{-2}	114.7	114.7$\times n$	2.43±1.28	0.19±0.15	4.58
DRIT [20]	1.2×10^{-2}	780	780$\times n$	2.83±1.17	0.14±0.09	5.06
NST [9]	4.3×10^{2}	576	576$\times n$	3.43±1.04	1.28±0.56	5.85
metaNST [28]	5.3×10^{-3}	64K+10[+870]	64K+10[+870]	2.97±0.93	0.13±0.09	4.74
WCT [21]	1.7×10^{0}	283.6	283.6	4.92±0.15	0.13±0.07	3.09
EGSC-IT [24]	8.2×10^{-1}	135	135$\times n$	2.71±1.29	0.26±0.19	5.50
EDIT w/o Adv	4.3×10^{-3}	8+3.6[+476]	8+3.6[+476]	3.39±0.98	0.26±0.10	4.59
EDIT	4.3×10^{-3}	8+3.6[+476]	8+3.6[+476]	2.33±0.98	0.09±0.07	5.90

Table 1: Quantitative comparison with the state-of-the-art methods. The two columns of model size (parameters) are for one domain pair and n domain pairs, respectively. For the content and style errors, lower values indicate better performance. While for the inception score, the higher the better.

the Gram matrix, height and width of the each feature map in the l-th layer, the style error can be expressed as $\frac{1}{N_L}\sum_{l=1}^{N_L}\frac{1}{4M_l^2H_l^2W_l^2}||\mathrm{Gram}_l(I_y) - \mathrm{Gram}_l(I_x^{y\in\mathbb{Y}})||_2^2$.

Comparisons. To quantitatively measure the performance of different competitors, we conduct the experiments on the translation from photo to painiting (Monet). The training and testing data are from [37]. The competitors are well-trained on the training data, and tested on the 750 testing data of the photo set with 10 exemplars from the Monet

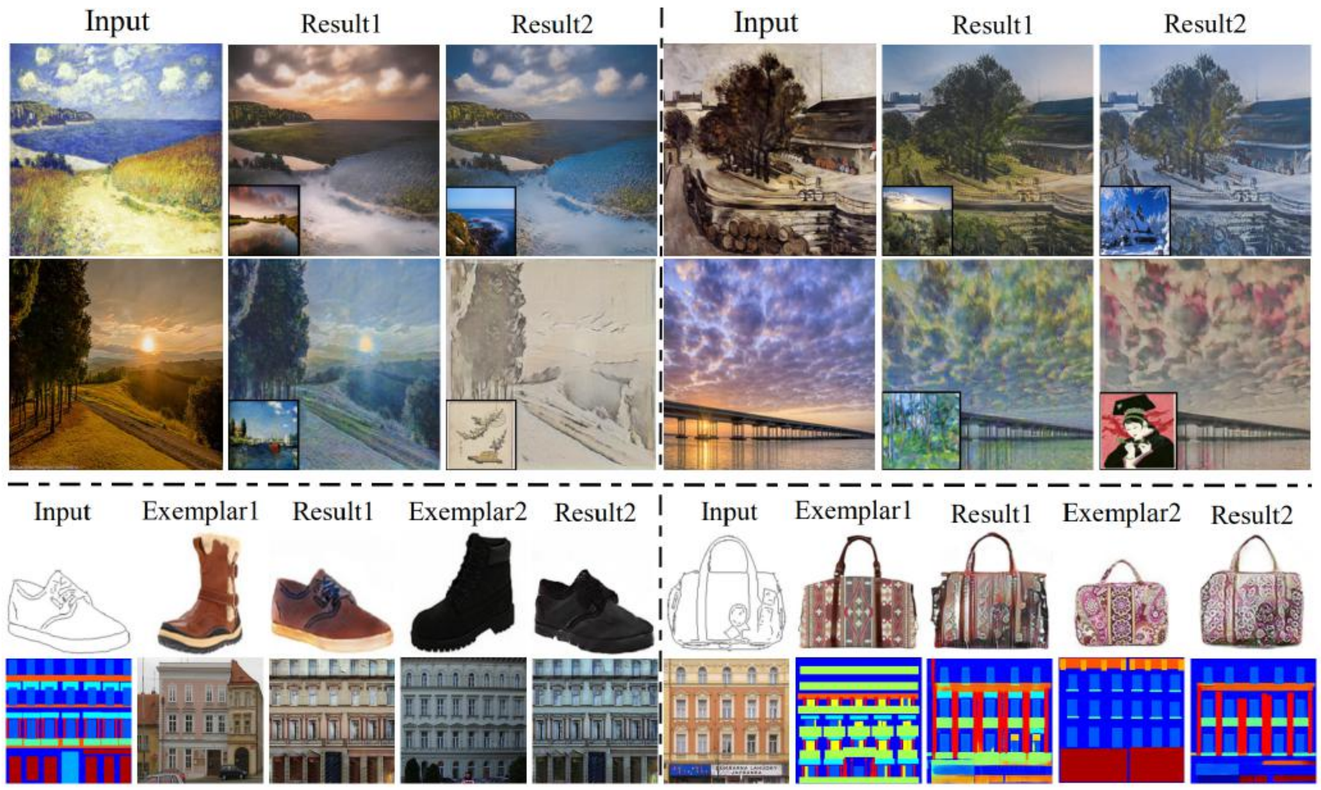

Figure 5: More visual results by our proposed EDIT.

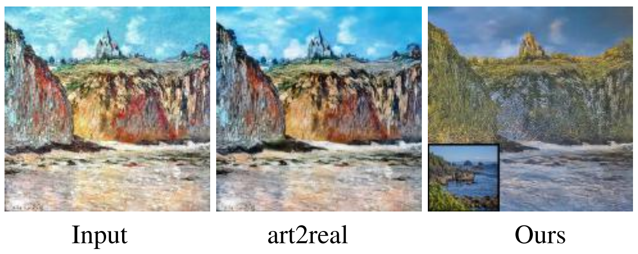

Figure 6: Visual comparison between art2real and EDIT

set. This is to say, each compared model generates 7,500 images, on which the content error, the style error, and the inception score are computed.

From Table 1, we can observe that in terms of content error, our EDIT slightly falls behind cycleGAN, while outperforms the others. Analogous analysis serves the inception score term. We notice that the I2I translation is a trade-off between the content and the style consistency. Notice that cycleGAN pays more attention on the content loss while only guaranteeing the domain style. As for the style loss, EDIT takes the first place among all the compared methods with a large margin. The cycleGAN is unable to take exemplars as reference, thus we do not provide its style error. In terms of model size, we provide two sets of comparison: one for one pair domain translation and the other for $n = 4$ pairs. Most of the methods including NST, MUNIT, cycleGAN, DRIT, and EGSC-IT require to train multiple models to handle multiple pairs of domain. While EDIT, metaNST and WCT can deal with multiple domain pairs in one model, thanks to either the dynamic parameter generators according to exemplars or the feature transfer manner. Different from metaNST and WCT, EDIT further takes into account the domain knowledge and the cycle consistency. The dynamic parameter generators for both EDIT and metaNST have several fully connected layers, making their models relatively large. A large part of the parameters in metaNST ($10Mb$ vs. $64K$ for shared part) are from the parameter generator, leading to a $870Mb$ storage. Ours demands the parameter network to produce about $3.6Mb$ (vs. $8Mb$ for shared part) parameters dynamically, significantly decreasing the storage $(476Mb)$ compared with metaNST. The above verifies, as previously stated in our principle, that the feature extraction can be done using a uniform extractor and part of feature reassembling can also be in common for different images from different domains. In this work, we consider 4 domain pairs, but it is possible to embrace more in the current models of metaNST and EDIT.

In terms of speed, NST takes much longer time, *i.e.* about $420s$ to process a case with size of 256×256, than

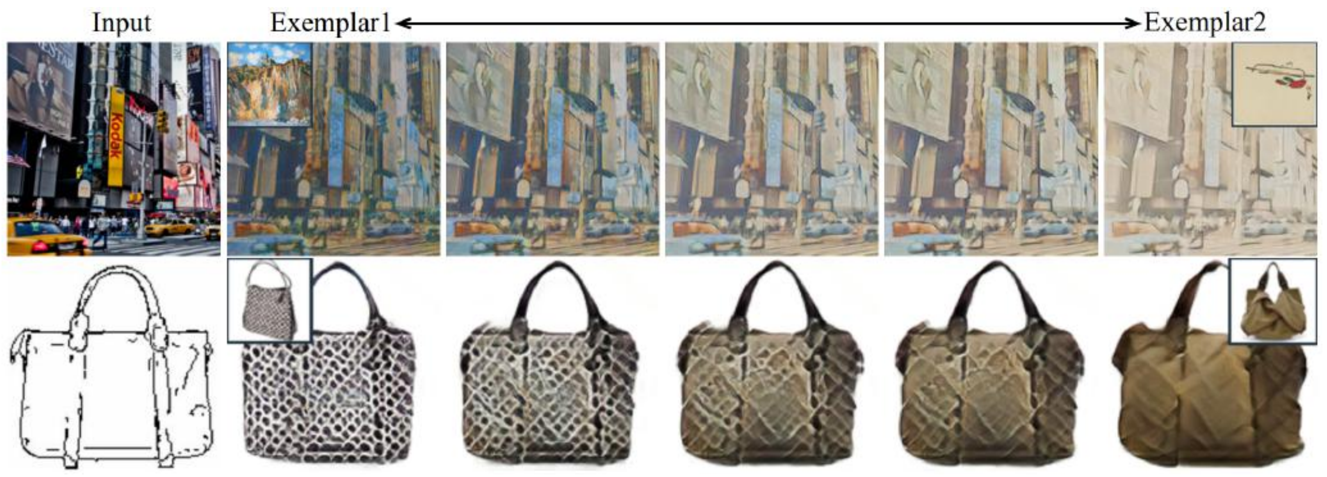

Figure 7: Interpolation results. The left-most column contains two inputs. The second and right-most columns are the results with respect to two different exemplars. The three columns in the middle are the interpolated results by EDIT.

the others, due to its processing way. The fastest method goes to cycleGAN ($3.5ms$), as it does not need to consider exemplars. Among the methods that consider exemplars, our EDIT is the most efficient one ($4.3ms$), slightly slower than cycleGAN. In addition, WCT is relatively slower ($1.7s$) because of the requirement of SVD operation that has to be executed in CPU. In Table 1, we also report the numbers corresponding to EDIT with the discriminator disabled, which reveal the importance of the adversarial mechanism for the target task. Figure 4 depicts three visual comparisons to qualitatively show the difference among the competitors. From the pictures, we can see that our EDIT can very well preserve the content of input and transfer the style of exemplar, making the final results visually striking. It is worth noting that art2real is specifically designed for translation from arts/paintings to realistic photos without using any exemplar, which if reasonably modified, needs multiple models for different domain pairs in nature. Figure 6 additionally gives a comparison between art2real and EDIT. The result by art2real indeed has some features of painting removed, however the unnatural-looking of which is still obvious. While, by taking an exemplar into consideration, EDIT produces a more realistic result. We provide other visual results by EDIT on painting$\leftrightarrow$ painting, edge $\rightarrow$ shoe, edge $\rightarrow$ handbag, and semantic map $\leftrightarrow$ facade in Figure 5 and at our website*. Comprehensively, the proposed EDIT is arguably the best candidate.

Style interpolation. One may want to take two or more exemplars/domains as style reference, and produce results simultaneously containing those styles in a controllable fashion. Considering that the dynamic parameters correspond to the exemplars, they can be viewed as their representations in the implicit manifold. Suppose the manifold is continuous and smooth, we can linearly combine the generated parameters to achieve the style interpolation. Figure 7 displays two cases of style interpolation. The second and the sixth columns offer the translated results by fully using different exemplars. The pictures shown in the middle columns are results by linearly interpolating the parameters of the second and the last columns. As can be seen, via controlling the parameter combination, the visual results vary smoothly between two styles with the content well-preserved.

*https://forawardstar.github.io/EDIT-Project-Page/

4. Concluding Remarks

In this paper, we have proposed a network, called EDIT, to translate images from different domains with consideration of specific exemplars in a unified model. The generator of EDIT is built upon a part of blocks configured by shared parameters to uniformly extract features for images from multiple domains, and another part by dynamic parameters exported by an exemplar-domain aware parameter network to catch specific style information. The concepts of cycle consistency and adversarial mechanism make the translation preserve the content and satisfy the distribution of target domain. We have conducted the extensive experiments to evaluate the performance of EDIT both quantitatively and qualitatively, which reveal the efficacy of our design, and its superiority over the state-of-the-art alternatives. Code is available at `https://github.com/ForawardStar/EDIT`.

References

[1] Y. Alharbi, N. Smith, and P. Wonka. Latent filter scaling for multimodal unsupervised image-to-image translation. In

CVPR, pages 1458–1466, 2019.

[2] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell. Multi-content gan for few-shot font style transfer. In *CVPR*, pages 7564–7573, 2018.

[3] K. Cao, J. Liao, and L. Yuan. Carigans: Unpaired photo-to-caricature translation. *arXiv preprint arXiv:1811.00222*, 2018.

[4] H. Chang, J. Lu, F. Yu, and A. Finkelstein. Pairedcyclegan: Asymmetric style transfer for applying and removing makeup. In *CVPR*, pages 40–48, 2018.

[5] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stylebank: An explicit representation for neural image style transfer. In *ICCV*, pages 1897–1906, 2017.

[6] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stereoscopic neural style transfer. In *CVPR*, pages 6654–6663, 2018.

[7] Y. Chen, Y.-K. Lai, and Y.-J. Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *ICCV*, pages 9465–9474, 2018.

[8] Y. Choi, M. Choi, M. Kim, J. W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, pages 8789–8797, 2018.

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[10] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio. Image-to-image translation for cross-domain disentanglement. In *NeurIPS*, pages 1287–1298, 2018.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[12] S. Gu, C. Chen, L. Jing, and Y. Lu. Arbitrary style transfer with deep feature reshuffle. In *CVPR*, pages 8222–8231, 2018.

[13] D. Ha, A.-M. Dai, and Q.-V. Le. Hypernetworks. In *ICLR*, 2017.

[14] H. Huang, W. Hao, W. Luo, M. Lin, W. Jiang, X. Zhu, Z. Li, and L. Wei. Real-time neural style transfer for videos. In *ICCV*, pages 783–791, 2017.

[15] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017.

[16] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 172–189, 2018.

[17] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[18] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[19] I. Korshunova, W. Shi, J. Dambre, and L. Theis. Fast face-swap using convolutional neural networks. In *ICCV*, pages 3677–3685, 2017.

[20] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, pages 35–51, 2018.

[21] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *NeurIPS*, pages 385–395, 2017.

[22] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[23] J. Lin, Y. Xia, T. Qin, Z. Chen, and T.-Y. Liu. Conditional image-to-image translation. In *CVPR*, pages 5524–5532, 2018.

[24] L. Ma, J. Xu, S. Georgoulis, T. Tuytelaars, and L. V. Gool. Exemplar guided unsupervised image-to-image translation with semantic consistency. In *ICLR*, 2019.

[25] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*, pages 3693–3703, 2018.

[26] E. Risser, P. Wilmot, and C. Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NeurIPS*, pages 2234–2242, 2016.

[28] F. Shen, S. Yan, and G. Zeng. Neural style transfer via meta networks. In *CVPR*, pages 8061–8069, 2018.

[29] Z. Shen, M. Huang, J. Shi, X. Xue, and T. S. Huang. Towards instance-level image-to-image translation. In *CVPR*, pages 3683–3692, 2019.

[30] L. Sheng, Z. Lin, J. Shao, and X.Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*, pages 8242–8250, 2018.

[31] Y. C. Shih, S. Paris, C. Barnes, W. T. Freeman, and F. Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics*, 33(4):1–14, 2014.

[32] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, pages 2107–2116, 2017.

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

[34] M. Tomei, M. Cornia, L. Baraldi, and R. Cucchiara. Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation. In *CVPR*, pages 5849–5859, 2019.

[35] Z. Wang, X. Tang, W. Luo, and S. Gao. Face aging with identity-preserved conditional generative adversarial networks. In *ICCV*, pages 7939–7947, 2018.

[36] P.-W. Wu, Y.-J. Lin, C.-H. Chang, E. Y. Chang, and S.-W. Liao. Relgan: Multi-domain image-to-image translation via relative attributes. *arXiv preprint arXiv:1908.07269*, 2019.

[37] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017.