Unpaired Cartoon Image Synthesis via Gated Cycle Mapping

Yifang Men[1], Yuan Yao[1], Miaomiao Cui[1], Zhouhui Lian[2], Xuansong Xie[1], Xian-Sheng Hua[1]
[1]DAMO Academy, Alibaba Group
[2]Wangxuan Institute of Computer Technology, Peking University, China

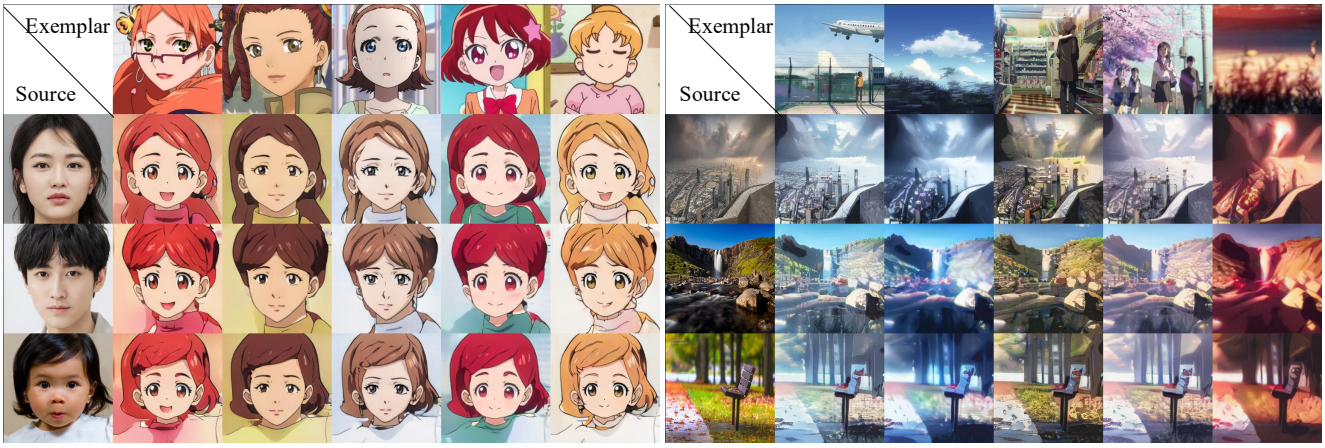


Figure 1. The proposed unpaired cartoon image synthesis method is able to convert diverse source photos (portrait in left, scene in right) into cartoon images with controllable cartoon styles as the corresponding exemplars. Source face credits: FFHQ [15] and Seeprettyface [2].

Abstract

In this paper, we present a general-purpose solution to cartoon image synthesis with unpaired training data. In contrast to previous works learning pre-defined cartoon styles for specified usage scenarios (portrait or scene), we aim to train a common cartoon translator which can not only simultaneously render exaggerated anime faces and realistic cartoon scenes, but also provide flexible user controls for desired cartoon styles. It is challenging due to the complexity of the task and the absence of paired data. The core idea of the proposed method is to introduce gated cycle mapping, that utilizes a novel gated mapping unit to produce the category-specific style code and embeds this code into cycle networks to control the translation process. For the concept of category, we classify images into different categories (e.g., 4 types: photo/cartoon portrait/scene) and learn finer-grained category translations rather than overall mappings between two domains (e.g., photo and cartoon). Furthermore, the proposed method can be easily extended to cartoon video generation with an auxiliary dataset and a new adaptive style loss. Experimental results demonstrate the superiority of the proposed method over the state of the art and validate its effectiveness in the brand-new task of general cartoon image synthesis.

1. Introduction

Cartoon is a popular art form that can be widely used in diverse scenes such as advertising, animation production, and the creation of virtual characters. Artists aim to build a vivid cartoon world in a simplified or exaggerated way based on real-world persons and scenarios. However, manually recreating the real world in cartoon styles is labor intensive and requires substantial professional skills.

Recently, inspired by the power of Generative Adversarial Networks (GANs) [10] in image-to-image translation tasks, a series of GAN-based methods have been proposed to achieve photo-to-cartoon (P2C) translation. These methods can be roughly categorized as scene cartoonization [6, 7, 31] and portrait cartoonization [26, 28, 33, 34], which are tailored to different use cases. For the former, the main idea is to introduce specialized losses or pre-extracted representations to sharpen edges and smooth surfaces, thus learning an abstract conversion between photo and cartoon images. However, they are incapable of generating vivid cartoon faces with exaggerated geometry transform, such as delicate big eyes and simplified mouths. Portrait cartoonization methods are proposed to produce manga [28, 33, 34] or caricature [5, 26] faces with large geometric changes. Yet, they heavily depend on facial characteristics (e.g., decomposed facial components or guided facial landmarks) and are not suitable for common scenes.

There also exist some unsupervised image-to-image translation (UIT) models [17, 23, 35] or StyleGAN-based methods [25, 27] that aim to handle the challenging selfie2anime task, while they either produce unsatisfactory results with missing contents or require training a model for each specific style. Overall, neither P2C nor UIT is capable of providing flexible user controls on cartoon styles, i.e., generating cartoon images in the style of an arbitrary input exemplar, and the portraits and scenes need to be processed via specifically designed models.

The goal of this paper is to design a general framework of cartoon image synthesis that is capable of rendering **diverse source photos** with **controllable cartoon styles**. As shown in Figure 1, with a single trained generator, exaggerated cartoon faces and realistic cartoon scenarios in desired styles (specified by input exemplars) can be simultaneously synthesized. The challenge of this task lies in three aspects. First, no paired training data is available and the model needs to be trained in an unsupervised way. Existing methods [7, 17, 31] typically utilize the cycle consistency to exploit unpaired data. But it is difficult to generate high-quality results due to the significant geometry changes along with texture style variation. Second, in contrast to pre-defined styles that can be straightforwardly learned by training on large-scale databases, we only have a style-mixed cartoon collection and aim to render images in an arbitrary style with the trained model. Third, due to different conversion requirements for portraits and scenes, multiple generators that are trained respectively for them are needed, making it a heavy architecture and thus limiting its practical usage.

To address the aforementioned challenges, we propose a simple yet effective cartoon image synthesis model with gated cycle mapping. In contrast to previous works [7, 17, 37] that forcedly learn bidirectional mappings between two domains using multiple generators ($G_{A\to B}$ and $G_{B\to A}$), we design a simplified cycle network with a single generator equipped with the gated style encoder E_{gs}. E_{gs} utilizes a novel gated mapping unit (GMU) consisting of domain and group specific layers to produce the category-specific style code, which can be directly injected into the generator to provide a target style guidance, making it easier to learn the texture style, meanwhile, enabling the network to transfer the corresponding style into a given image. For the concepts of group and category, considering the huge semantic discrepancy and different conversion requirements between portrait and scene images, we introduce a fine-grained category translation mechanism. All images in each domain (photo or cartoon) are further classified into two groups (portrait and scene), and only category translations within each group will be learned, aiming to ignore unreasonable mappings with mismatched structures. Cooperating with the gated cycle networks mentioned above, we can simply use a single generator for image translation in all directions, where only the decoder part is modulated by the corresponding style codes. The proposed strategy not only achieves a common cartoon translator with significantly lighter architecture, but also provide a flexible user control for desired cartoon styles. In summary, major contributions of this paper are threefold:

- We propose a brand-new task of synthesizing style-controllable cartoon images with a common translator for both portraits and scenes, and solve it by designing a novel gated cycle mapping network.
- We develop a gated mapping unit which utilizes the gating mechanism to learn category-specific style representations via domain and group specific layers.
- We extend the proposed method to video synthesis of cartoon portraits, leveraging an auxiliary dataset and a new adaptive style loss, which achieves stable results with the precise control of facial expressions.

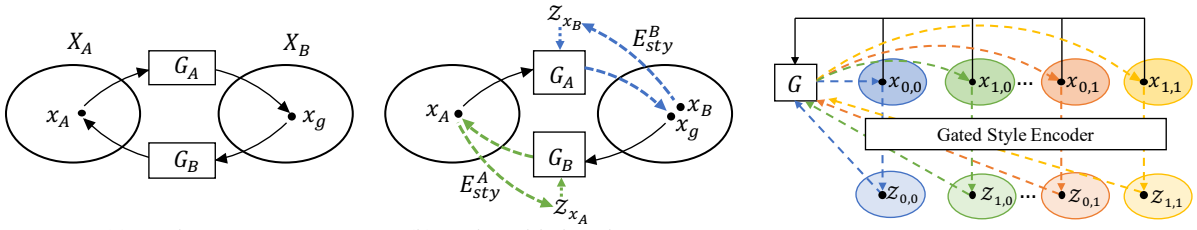
2. Related Work

2.1. GAN-based Image-to-Image Translation

Generative adversarial networks (GANs) [10] have been widely used for many computer vision tasks such as image translation [14, 21], image super-resolution [19, 32] and image inpainting [24]. Among these tasks, the image-to-image translation framework provides a general solution of translating images between two domains via supervised [14, 30] or unsupervised learning [21, 37]. Pix2pix [14] is the first work to propose a supervised image translation model with conditional GANs [22], and was later extended to generate high-resolution images [30]. Due to the difficulty of obtaining paired images, CycleGAN [37] exploits cycle consistency to learn the transform from unpaired data. UNIT [21] tackles the same problem by making a shared-latent space assumption. To produce diverse outputs from the source domain image, multimodal methods [13, 20] were proposed by combining the domain-invariant content with a random domain-specific style. Despite great progresses achieved, these techniques have limited scalability for cartoon image synthesis, due to misaligned structures with exaggerated geometry and simplified strokes. Recently, U-GAT-IT [17] introduces an attention module and a new normalization to alleviate this issue, but it still cannot produce satisfactory results with smooth lines and diverse styles. Our model overcomes these challenges and is able to synthesize high-quality cartoon images with controllable cartoon styles.

2.2. Cartoon Image Generation

Scene cartoonization. Chen *et al*. [7] first proposed a GAN-based model for cartoon stylization and introduced



(a) CycleGAN (b) Style-guided cycle structure (c) Gated cycle mapping structure

Figure 2. An illustration of our gated cycle mapping framework. The proposed model starts from CycleGAN (a). Instead of forcibly learning a mapping between two domains $\{X_A, X_B\}$, we introduce a style guidance z by directly injecting the style code z_x of a target sample x into the generator to achieve exaggerated geometric transform and adaptive style transfer (b). It is further developed to (c), a gated cycle mapping structure utilizing the gated style encoder to produce the category-specific style code $z_{i,j}$, thus handling diverse transform requirements in a single framework.

the semantic content loss and edge loss to preserve clear edges and smooth shading. It was further extended to AnimeGAN [6], a faster model with lightweight designs and improved loss functions. Wang *et al*. [31] utilized white-box representations extracted from images to guide the cartoonization process. Although high-quality results from the real-world scenes to cartoon animations are produced, these models only learn the texture abstraction and they are unable to synthesize exaggerated cartoon portraits.

Portrait cartoonization. Yi *et al*. [33] proposed APDrawingGAN using a hierarchical structure to transfer face photos to portrait drawings and extended it to an unsupervised version [34]. MangaGAN [28] employed a multi-GANs architecture to generate each facial component respectively and combined them together to synthesize final manga results. [5, 26] achieved photo-to-caricature translation by warping stylized portraits via estimated landmarks. Recently, StyleGAN-based cartoonization methods [25, 27] have gained large popularity by combining inversion algorithms [3, 29, 36] with transferred StyleGAN models [15, 16]. Despite high-quality results, they require training a model for each specific style and easily suffer from content missing. Moreover, all these methods tailored to portrait transfer lack the generality for common scenes and scalability for various styles. In this paper, we propose a universal framework, which can transfer arbitrary cartoon styles to diverse photos, including both portrait and scene.

3. Method Description

In this section, we first formulate the task of general cartoon image synthesis and give an overview of how to solve the problem via the proposed gated cycle mapping (Section 3.1). Then we present a detailed description for each part of the network architecture (Section 3.2) and the design of the training scheme (Section 3.3). Finally, we extend the proposed method to video generation of cartoon portraits via an auxiliary dataset and a new adaptive style loss (Section 3.4).

3.1. Problem Formulation and Analysis

Let X_A and X_B be the image sets in the photo and cartoon domains, respectively, and no pair data exists between these two domains. The proposed method aims at converting a source photo $x_A \in X_A$ to a target cartoon image $x_g \in X_B$ with controllable cartoon styles. Our model starts from CycleGAN [37], which leverages cycle consistency $G_B(G_A(X)) \sim X$ to achieve domain translations without paired training data, as shown in Figure 2 (a). However, we observe that when applying the above strategy in the task of "portrait photo to anime face" translation, due to the significant geometry changes along with texture style variation, it is difficult to generate high-quality results with correct structures preserved. Considering that it is hard to forcibly learn a mapping $G_A(x_A) \in X_B$ in an unsupervised way, but much simpler to learn such a translation by directly injecting the texture style of target domain images into source features, we introduce a style-guided cycle structure as shown in Figure 2 (b). During training, a target exemplar x_t is randomly fetched from the target domain X_t to provide a style guidance z_{x_t}, combined with delicately designed style losses, the network is encouraged to generate style-adaptive cartoon images with the analogous style as x_t. Such an intuitive strategy could kill two birds with one stone: 1) This makes it easier for the network to achieve exaggerated geometry transform. 2) It enables a flexible and continuous user control of cartoon styles.

Furthermore, different from other image translation tasks (e.g., cat2dog, female2male) where each domain includes a set of images belonging to the same species, our task defines "domain" as all kinds of images in the style of photo or cartoon, leading to a significant structure discrepancy among images in each domain. According to extensive observations of cartoon painting samples, we found that most cartoon characters are composed of exquisite big eyes and simplified noses and mouths, which reflects real-world persons in an exaggerated way with large geometric changes. However, cartoon scenes are produced from real

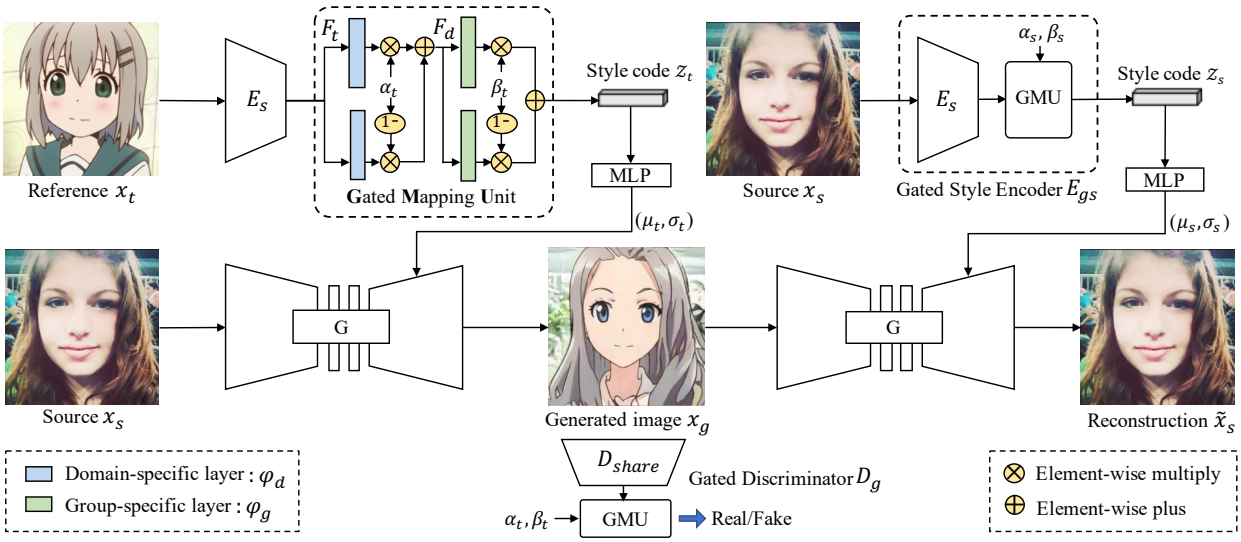


Figure 3. An overview of our network architecture, which consists of a generator G, a gated style encoder E_{gs} and a gated discriminator D_g. G is used to synthesize the cartoonized output x_g of a source image x_s following the analogous style of a reference image x_t. E_{gs} utilizes the gated mapping unit (GMU) consisting of domain and group specific layers φ_d, φ_g to produce the category-specific style code, which is injected into the decoder via MLP to guide the generation process. D_g also utilizes GMU to learn a category-specific binary classification. Since these is no paired data available, cycle mapping is adopted for image reconstruction. Source: ©selfie2anime [17].

photos with only clear boundaries and sparse color blocks, which reflects the real-world photography in a relatively realistic way. This property makes different conversion requirements from scene and portrait images. To resolve this problem, we first perform a fine-grained data partition by dividing images in each domain into two groups (portrait or scene) and thus classify all images as four categories with distinct styles or requirements, defined as $X_{i,j}, i, j \in \{0, 1\}$, where i, j denotes the domain label (photo or cartoon) and the group label (portrait or scene), respectively. In contrast to previous methods learning translations between photo and cartoon domains, our method learns only category translations within each group (e.g., photo portrait $\leftrightarrow$ cartoon portrait), thus avoiding unreasonable mappings with mismatched structures. In this way, the complicated general cartoonization task can be simplified to a special multi-domain translation problem [8, 9] with customized mappings. Instead of using multiple generators and encoders, an elegant gated cycle mapping structure is designed by embedding a gated style encoder E_{gs} into the cycle mapping networks, as shown in Figure 2 (c). The encoder E_{gs} equipped with a novel gated mapping unit can produce the category-specific style code $z_{i,j}$ for a style image $x_{i,j}$. With $z_{i,j}$ representing the style of a specific category, we can replace the original generators $\{G_A, G_B\}$ with a common generator G and utilize $z_{i,j}$ from the target category to control the translation direction, i.e., forcing G to learn how to transform an image into the specific category. The gated mapping unit is also integrated into the discriminator for multi-category discrimination. In this way, our method can generate exaggerated cartoon faces and realistic cartoon scenes simultaneously by using a significantly light architecture. In the following, we will give a detailed description for each part of the proposed model.

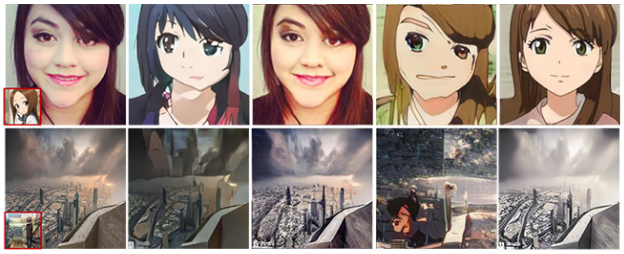
3.2. Network Architecture

3.2.1 Generator

Let x_s and x_t represent the samples from the source and target categories, respectively, and z_t denote the style code outputted by the gated style encoder E_{gs}. Our generator adopts the encoder-decoder architecture and the style code z_t is fed into the decoder via adaptive instance normalization (AdaIN) [12, 15]. Given a source image x_s and the style code z_t extracted from a reference style image x_t, a generated image can be obtained by $x_g = G(x_s, f(z_t))$, where $f(z_t)$ denotes the AdaIN parameters (scale μ and shift σ) dynamically generated by a multilayer perceptron (MLP). Since it is hard to collect paired data in this task, we utilize a cycle structure [37] to reconstruct the source image as $\widetilde{x}_s = G(x_g, f(z_s))$, where G is the shared generator and the style code z_s is extracted from x_s.

3.2.2 Gated style encoder

The gated style encoder E_{gs} aims to produce the category-specific style code z_t for a reference style image x_t in the category X_t. Considering both shared and unique style representations for images in different categories, we construct the gated style encoder E_{gs} by connecting a gated mapping unit (GMU) at the backend of a regular style encoder



(a) Input (b) w/o style guide (c) w/o φ_d (d) w/o φ_g (e) Full model

Figure 4. Effects of the gated style guidance. (a) Inputs of source and reference. (b) Results generated without style guidance. (c, d) Results generated without domain/group specific layers in GMU. (e) Full model results. Source face: ©selfie2anime [17].

E_s. Specifically, E_s is used to extract a common feature F_t from a reference image x_t, and GMU embeds F_t into a specific category space to obtain the customized style code z_t. The proposed GMU consists of domain-specific layers and group-specific layers connected by a gating mechanism. The common feature F_t firstly goes through domain-specific layers $\varphi_{d_i}(i = 0, 1)$ in different branches and then we obtain the feature F_d via the selection gate by:

$$F_d = \alpha_t \cdot \varphi_{d_0}(F_t) + (1 - \alpha_t) \cdot \varphi_{d_1}(F_t), \tag{1}$$

where $\alpha_t \in \{0, 1\}$ is the control factor acting as a switch to make the output features of the selected domain layer effective. For example, with x_t from the cartoon domain, α_t is set to 1 making F_t go through the specific layer φ_{d_0}. The same goes for the final style code z_t produced by group-specific layers as:

$$z_t = \beta_t \cdot \varphi_{g_0}(F_d) + (1 - \beta_t) \cdot \varphi_{g_1}(F_d). \tag{2}$$

As we can see, the values of α_x and β_x depend on the domain label and group label of the image x, respectively. The layers in GMU are constructed with fully-connected layers. The category-specific style code z_t is later injected into the generator to guide the translation process.

Figure 4 shows some synthesis results demonstrating effects of the gated style guidance. The style guidance makes it more accessible for our method to achieve large geometric changes, especially for cartoon portraits. For GMU, without domain-specific layers φ_d, photo and cartoon images are regarded as the same category, and thus compromised results are generated with the intermediate texture style of two domains. Group-specific layers φ_g eliminate the mutual interference brought by portraits and scenes with discrepant semantics and help producing exaggerated portraits and realistic scenes in cartoon styles simultaneously.

3.2.3 Gated discriminator

Given an image x, the discriminator is expected to discriminate whether x is a real image of the desired category or a fake image produced by G. Similar to E_{gs}, the proposed

(a) Source (b) Reference (c) w/o L_{sty} (d) w/o L_{ds} (e) Full model

Figure 5. Effects of the style reconstruction loss $\mathcal{L}_{sty}$ and the diverse style loss $\mathcal{L}_{ds}$. Source: ©selfie2anime [17].

GMU is integrated into the regular discriminator to help it learn a category-specific binary classification, denoted as the gated discriminator D_g. In the $X_s \rightarrow X_t$ process, the generated image x_g (or the reference image x_t) is fed into D_g as a fake (or real) sample. The control factors in GMU for x_g are equal to (α_t, β_t), since x_g and x_t belong to the same category. The same goes for the reverse process $X_t \rightarrow X_s$.

3.3. Training

Given a source image $x_s \in X_s$ and a reference image $x_t \in X_t$, we train our model with a loss function consisting of an adversarial term, an image reconstruction term, a style reconstruction term and a style diversity term:

$$\mathcal{L}_{total} = \mathcal{L}_{adv} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{sty}\mathcal{L}_{sty} + \lambda_{ds}\mathcal{L}_{ds}, \tag{3}$$

where $\lambda_{rec}, \lambda_{sty}$ and λ_{ds} denote the weights of corresponding losses, respectively.

Adversarial loss. We apply the adversarial loss $\mathcal{L}_{adv}$ [10] to both mapping directions. For the mapping direction: $X_s \rightarrow X_t$, given a source image x_s and a reference style image x_t, the generator G synthesizes the cartoonized result x_g with the similar style as x_t. The distance between the distribution of real samples X_t and the distribution of fake samples X_g generated by G is computed as:

$$\begin{aligned}\mathcal{L}_{adv} =& \mathbb{E}_{x_s,x_t}[log(1 - D_g(G(x_s, f(z_t))))] \\ &+ \mathbb{E}_{x_t}[log(D_g(x_t)],\end{aligned} \tag{4}$$

where the style code z_t is extracted using $E_{gs}(x_t)$.

Image reconstruction loss. Since there is no paired data available in this task, we employ the cycle consistency loss [37] to push the reconstructed image $\widetilde{x}_s$ produced by sequential translations $X_s \rightarrow X_t \rightarrow X_s$ be identical as x_s, making the source image be successfully translated back to its original category. It implicitly ensures that the generated image $\widetilde{x}_s$ properly preserves the semantic content of the source image x_s and can be formulated using the L1 distance as:

$$\mathcal{L}_{rec} = ||G(G(x_s, f(z_t)), f(z_s)) - x_s||_1. \tag{5}$$

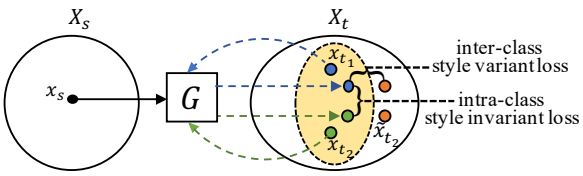


Figure 6. An illustration of the adaptive style loss. With samples x_{t_1} and x_{t_2} fetched from the same class (marked in yellow) as references, the intra-class style invariant loss encourages generated images produced by $G(x_s, x_{t_1})$ and $G(x_s, x_{t_2})$ to be equal. The inter-class style variant loss encourages generated results to be different with inter-class samples $(x_{t_1}, \widetilde{x}_{t_2})$.

Style reconstruction loss. To ensure the cartoon style of the generated image x_g be coherent with the reference target sample x_t, we apply a style reconstruction loss $\mathcal{L}_{sty}$ similar to [13, 38], which provides a style constraint for style representation in the latent space:

$$\mathcal{L}_{sty} = ||E_{gs}(G(x_s, f(z_t))) - z_t||_1. \tag{6}$$

The effect of $\mathcal{L}_{sty}$ is shown in Figure 5 (c)(e).

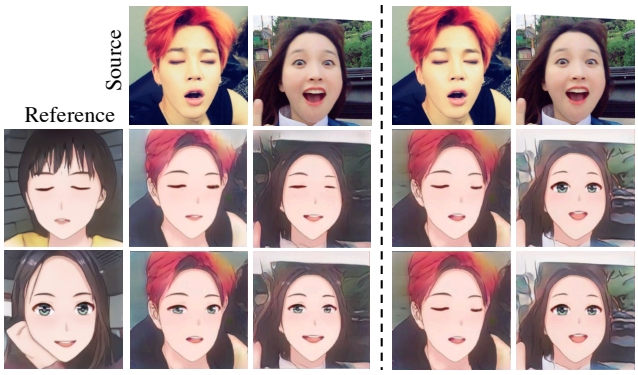
Diverse style loss. To further encourage the network synthesizing diverse outputs coherent with the different styles provided by reference images, we apply an intuitive constraint to the generator. Given two samples (x_{t_1}, x_{t_2}) from the target domain X_t that provide various style representations (z_{t_1}, z_{t_2}) and a source image x_s, the synthesized images x_{g_1}, x_{g_2} should have different appearances. We define $\mathcal{L}_{ds}$ as the L1 distance between x_{g_1} and x_{g_2}:

$$\mathcal{L}_{ds} = -||G(x_s, f(z_{t_1})) - G(x_s, f(z_{t_2}))||_1. \tag{7}$$

As shown in Figure 5 (d)(e), $\mathcal{L}_{ds}$ encourages the style code containing more cartoon details of references, not only the abstract texture and color palette, but also the hair color, eye size and face shape for various personified degrees. It is worthy of noticing that all local styles of the reference are automatically captured without any local guidance.

3.4. Further Extension

For cartoon portraits, the proposed method guarantees that the generated image x_g can properly preserve the high-level content structures (e.g., poses, viewpoints and human attributes) of the source image x_s and the texture style (e.g., abstract stroke, hair color and facial features) of the reference image x_t. However, it is still difficult to produce results with content structures completely preserved. Not only high-level attributes, but also local details such as facial expressions should be coherent with the source image. One of the basic reasons for this is that there is a bias in the training data, and images in dynamic expressions rarely appear. In this section, we tackle the above problem and extend the proposed method to video synthesis of cartoon portraits. Specially, an auxiliary dataset and a new adaptive style loss are introduced to produce stable results with high consistency in content details.



(a) Results with $\mathcal{L}_{ds}$ (b) Results with $\mathcal{L}_{as}$

Figure 7. Effects of replacing the original diverse style loss $\mathcal{L}_{ds}$ with the new adaptive style loss $\mathcal{L}_{as}$. Source: ©Google [1].

Data expansion. The cartoon and photo portraits we use are from the selfie2anime dataset [17], which includes diverse anime faces in mixed styles and each image is regarded as a class of cartoon style. Considering the data bias (e.g., most anime faces have bangs) and the absence of content variety for certain styles, we introduce a set of cartoon portraits containing diverse character characteristics and different facial expressions (e.g., open/closed eyes/mouth) in a similar cartoon style as a new class c_{new}, which is added to the original dataset, making it possible to synthesize dynamic expressions.

Adaptive style loss. The original diverse style loss $\mathcal{L}_{ds}$ assumes that each reference image $x_t \in X_t$ represents a cartoon style and encourages the network to generate diverse outputs with different reference style images. However, it may bring content inconsistency for the extended dataset with a new class c_{new}, that includes a series of cartoon portraits in a similar style. When x_{t_1} and x_{t_2} are both sampled from c_{new} in a similar style, the diverse style loss $\mathcal{L}_{ds}$ still forces the network to produce various images, making local contents change following the reference, and thus generating inconsistent facial expressions with the source image, as shown in Figure 7 (a). Thus, we replace $\mathcal{L}_{ds}$ with a new adaptive style loss $\mathcal{L}_{as}$ which is intuitively visualized in Figure 6. Given intra-class samples x_{t_1} and x_{t_2} as the reference style images, we encourage the synthesis results produced by $G(x_s, x_{t_1})$ and $G(x_s, x_{t_2})$ to be equal, which is defined as the intra-class style invariant loss. For inter-class samples x_{t_1} and $\widetilde{x}_{t_2}$, the generated results should be different with the inter-class style variant loss (equals to $\mathcal{L}_{ds}$). Specifically, the adaptive style loss $\mathcal{L}_{as}$ is computed by:

$$\mathcal{L}_{as} = \begin{cases} -\mathcal{L}_{ds}, & (x_{t_1}, x_{t_2}) \in c_{new} \\ \mathcal{L}_{ds}, & \text{others.} \end{cases} \tag{8}$$

This intuitive strategy ensures that only content-

irrelevant features are extracted from x_t. thus achieving a precise control of facial expressions (see Figure 7 (b)).

4. Experimental Results

In this section, we first describe implementation details and the dataset used for evaluation. Then, we verify the effectiveness of the proposed method for universal cartoon image synthesis and illustrate its superiority over other state-of-the-art methods. Finally, we show that our method can be extended for video synthesis of cartoon portraits.

Implementation details. Our method is implemented in PyTorch using a single NVIDIA Tesla-V100 GPU with 32GB memory. The architectures of our generator, gated style encoder and gated discriminator are described in the supplementary materials. α_x and β_x are the control factors in GMU and are set to denote the domain label (0 for photo, 1 for cartoon) and group label (0 for portrait, 1 for scene) of the image x, respectively. The weights for the loss terms are set to $\lambda_{rec} = 1, \lambda_{sty} = 1$. The initial value of $\mathcal{L}_{ds}/\mathcal{L}_{as}$ is set to 2 and linearly decayed to 0 over 100k iterations. We use the Adam optimizer [18] with the learning rate 1e-4 to train our model for around 100k iterations.

Datasets. We conduct experiments on a mixed photo2cartoon dataset, which consists of portrait and scene data covering diverse situations. For the portrait data, we use the selfie2anime dataset [17] to provide cartoon portraits and photo portraits, serving as two categories. Following the same data configuration in selfie2anime, 3400 selfie photos and 3400 anime faces with the resolution of 256×256 are used for training, and 100 selfie photos and 100 anime faces for testing. For the scene data, we construct a scene2cartoon dataset by collecting 5100 landscape photos and 5100 animation scenes from the dataset proposed by [31], serving as photo scene and cartoon scene, respectively. We randomly pick 5000 scene images for training and the remaining 100 scene images for testing.

4.1. General Cartoon Image Synthesis

4.1.1 Cartoon image synthesis in controllable styles

Our experiments verify the effectiveness of the proposed method in transferring desired cartoon styles to diverse source photos. As shown in Figure 8, given a source photo and an arbitrary cartoon exemplar in the test set, our method can generate high-quality results preserving the semantic structure of the source and the cartoon style of the exemplar. It achieves an adaptive geometry transfer with a common cartoon translator, which enables both exaggerated facial features for cartoon portraits and realistic structure textures for cartoon scenes. Besides the photo images in the test set, we also test our model with in-the-wild images and diverse scene cases (e.g., animals, foods, city views and other objects) to demonstrate the generation ability of the networks (see our supplemental materials).

Method	selfie2anime		scene2cartoon	
	FID ↓	KID ↓	FID ↓	KID ↓
CartoonGAN [7]	-	-	267.84	7.86±0.80
AnimeGAN [6]	-	-	255.85	**6.19±0.74**
CycleGAN [37]	91.35	2.50±0.27	265.26	6.59±0.74
MUNIT [13]	93.69	2.48±0.26	270.80	8.38±0.55
DRIT++ [20]	93.07	2.84±0.27	282.73	10.57±0.84
U-GAT-IT [17]	90.05	2.61±0.31	285.32	9.10±0.65
CouncilGAN [23]	89.51	2.36±0.23	-	-
Ours	**79.74**	**1.59±0.25**	**253.83**	6.40±0.73

Table 1. FID and KID×100±std.×100 scores for two tasks.

Style interpolation. Our model constructs a complex manifold that is constituted of various cartoon images in different contents and diverse styles. We can travel along this manifold by mixing and interpolating the style representations extracted from different references, thus synthesizing an animation from one cartoon style to another. The results of style interpolation are provided in the supplemental video.

4.1.2 Comparisons with state-of-the-art methods

In this section, we compare our proposed method with other existing approaches both qualitatively and quantitatively.

Qualitative comparison. In Figure 9, we first compare the selfie2anime results of our method with four state-of-the-art methods: CycleGAN [37], U-GAT-IT [17], CouncilGAN [23] and White-box [31]. Due to the inability of UIT models [13, 17, 20, 23, 37] for learning portrait and scene translation simultaneously, synthesis results of these methods are produced by using independent selfie2anime (or scene2cartoon) models trained with the corresponding data. It should be pointed out that we only train a single model with mixed data. Still, our method outperforms other approaches and synthesizes high-quality anime faces with clear edges and delicate features, such as exquisite big eyes and fluent structure lines. More content details are also better preserved. Due to the great semantic discrepancy among scene images, multimodal methods [13, 23] fail to synthesize reasonable results. Thus, we replace CouncilGAN with AnimeGAN [6], a P2C method tailored to scenes, for scene2cartoon comparison. As shown in the right of Figure 9, our method alleviates issues in multimodal models and produces more exquisite results than P2C methods.

Quantitative evaluation. We first evaluate the visual quality using Fréchet Inception Distance (FID) [11] and Kernel Inception Distance (KID) [4] between the feature representations of real and generated images. As CartoonGAN [7] and AnimeGAN [6] are P2C methods designed for cartoon abstraction within specific styles, they are incapable of synthesizing anime characters. We only calculate the metrics on the scene2cartoon task. For fair comparison, we evaluate the performance of CouncilGAN [23] for the selfie2anime task only when it is trained with face related tasks. We also include another multimodal method DRIT++ [20] for eval-

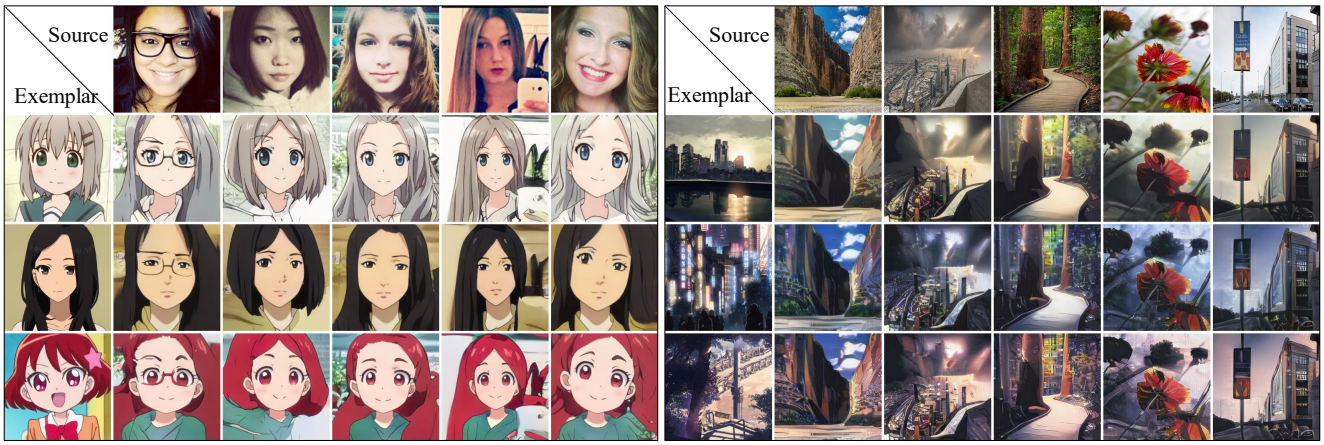


Figure 8. Results of synthesizing diverse cartoon images (portrait in left, scene in right) with controllable styles provided by the corresponding exemplars. With a single trained model, cartoon scenes can be simultaneously synthesized with portraits. Source: ©selfie2anime [17].

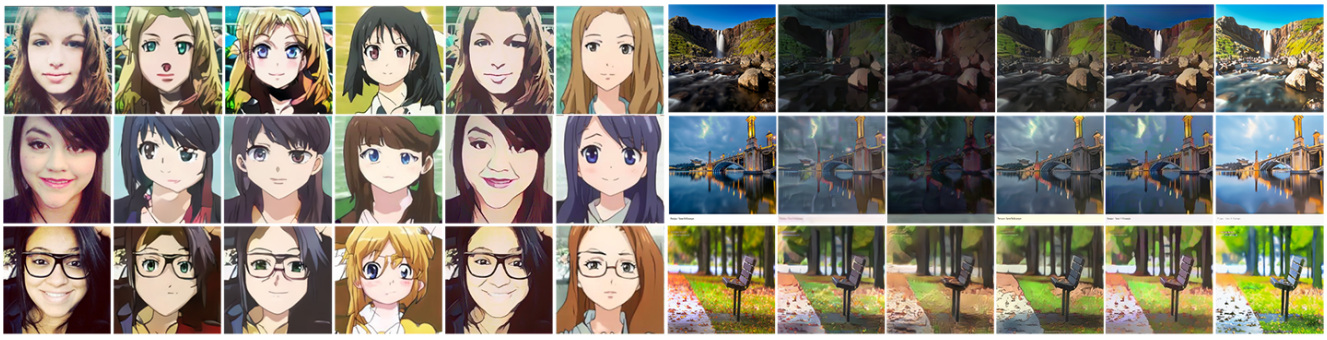


Figure 9. Qualitative comparison with state-of-the-art methods. Our results are generated with random styles. Source: ©selfie2anime [17].

Figure 10. Results of representative frames for cartoon video synthesis and the corresponding original video. Source: ©Google [1].

uation. We use the same test images from the source domain for all methods. As shown in Table 1, our method outperforms others to a large margin in the selfie2anime task, further verifying that our generated anime faces are more visually similar with real images in the target domain. For the scene2cartoon task, our method outperforms other UIT methods and is comparable to P2C methods.

4.2. Video Synthesis of Cartoon Portraits

With the method extension described in Section 3.4, our model is able to achieve visually-pleasing video synthesis of cartoon portraits, which can not only preserve the content details of the source image but also make a precise control of facial expressions. Given an image $x_t \in c_{new}$ as the reference and a source video consisting of a series of portrait frames, the proposed model can generate a cartoon video with continuous facial changes. Results of some representative frames are depicted in Figure 10 and more complete videos can be found in the supplemental video. More results and other discussions (e.g., limitations, negative impact, etc.) can be found in the supplemental materials.

5. Conclusion

In this paper, we presented an unpaired cartoon image synthesis method that enables not only adaptive geometry transfer for diverse photos, but also flexible user control of cartoon styles. We formulated the task of general cartoon image synthesis as a multimodal and multi-domain image translation problem and proposed gated cycle mapping to solve it, in which the gated style guidance is embedded into cycle networks to control the translation process. With a novel gated mapping unit, category-specific style codes adapted to various images with distinct textures or structures can be obtained. Experimental results not only demonstrated the effectiveness and superiority of our method by comparing with the state of the art, but also validated its extensibility for video synthesis of cartoon portraits.

References

[1] Google. [EB/OL]. https://google.com/. 6, 8

[2] Seeprettyface. [EB/OL]. https://seeprettyface.com/. 1

[3] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 3

[4] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 7

[5] Kaidi Cao, Jing Liao, and Lu Yuan. Carigans: Unpaired photo-to-caricature translation. *arXiv preprint arXiv:1811.00222*, 2018. 1, 3

[6] Jie Chen, Gang Liu, and Xin Chen. Animegan: A novel lightweight gan for photo animation. In *International Symposium on Intelligence Computation and Applications*, pages 242–256. Springer, 2019. 1, 3, 7

[7] Yang Chen, Yu-Kun Lai, and Yong-Jin Liu. Cartoongan: Generative adversarial networks for photo cartoonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9465–9474, 2018. 1, 2, 7

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 4

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 4

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2, 5

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems(NIPS)*, pages 6626–6637, 2017. 7

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 4

[13] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 2, 6, 7

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1, 3, 4

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 3

[17] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019. 2, 4, 5, 6, 7, 8

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[19] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[20] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018. 2, 7

[21] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 2

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[23] Ori Nizan and Ayellet Tal. Breaking the cycle-colleagues are all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7860–7869, 2020. 2, 7

[24] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[25] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020. 2, 3

[26] Yichun Shi, Debayan Deb, and Anil K Jain. Warpgan: Automatic caricature generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10762–10771, 2019. 1, 3

[27] Guoxian Song, Linjie Luo, Jing Liu, Wan-Chun Ma, Chunpong Lai, Chuanxia Zheng, and Tat-Jen Cham. Agilegan: stylizing portraits by inversion-consistent transfer learning. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 2, 3

[28] Hao Su, Jianwei Niu, Xuefeng Liu, Qingfeng Li, Jiahe Cui, and Ji Wan. Unpaired photo-to-manga translation based on the methodology of manga drawing. *arXiv preprint arXiv:2004.10634*, 2020. 1, 3

[29] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 3

[30] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018. 2

[31] Xinrui Wang and Jinze Yu. Learning to cartoonize using white-box cartoon representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8099, 2020. 1, 2, 3, 7

[32] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 2

[33] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2019. 1, 3

[34] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Unpaired portrait drawing generation via asymmetric cycle mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8217–8225, 2020. 1, 3

[35] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss. *arXiv preprint arXiv:2003.04858*, 2020. 2

[36] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 3

[37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2, 3, 4, 5, 7

[38] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in neural information processing systems*, pages 465–476, 2017. 6