Abstract—In this paper, we propose a novel framework to translate a portrait photo-face into an anime appearance. Different from existing translation methods which do not designate specific styles, we aim to synthesize anime-faces which are style-consistent with a given reference anime-face. However, unlike typical translation tasks, such anime-face translation is particularly challenging due to the large and complex variations of appearances among anime-faces. Existing methods often fail to transfer the styles of reference anime-faces to the generated anime-faces, or introduce noticeable artifacts/distortions in the local shapes of their generated anime-faces. We propose a novel GAN-based anime-face translator, called AniGAN, to synthesize high-quality anime-faces. Specifically, a new generator architecture is proposed to simultaneously transfer color/texture styles and transform local facial shapes into anime-like counterparts based on the style of a reference anime-face, while preserving the global structure of the source photo-face. New normalization functions are designed for the generator to further improve local shape transformation and color/texture style transfer. Besides, we propose a double-branch discriminator to learn domain-specific distributions through individual branches and learn cross-domain shared distributions via shared layers, helping generate visually pleasing anime-faces and effectively mitigate artifacts/distortions. Extensive experiments on benchmark datasets qualitatively and quantitatively demonstrate the superiority of our method over state-of-the-art methods.

Index Terms—GAN, Image translation, Non-photorealistic rendering, Style transfer

I. INTRODUCTION

Animations play an important role in our daily life and have been widely used in entertainment, social, and educational applications. Recently, anime, aka Japan-animation, has been popular in social media platforms. Many people would like to transfer their profile photos into anime images, whose styles are similar to that of the roles in their favorite animations such as Cardcaptor Sakura and Sailor Moon. However, commercial image editing software fails to do this transfer, while manually producing an anime image in specific styles needs professional skills.

In this paper, we aim to automatically translate a photo-face into an anime-face based on the styles of a reference anime-face. We refer to such a task as Style-Guided Face-to-Anime Translation (StyleFAT). Inspired by the advances of generative adversarial networks (GANs) [1], many GAN-based methods (e.g., [2]–[5]) have been proposed to automatically translate images between two domains. However, these methods [2, 6] focused on learning one-to-one mapping between the source and target images, which does not transfer the information of the reference image into a generated image. Consequently, the styles of their generated anime-faces [2, 6] are usually dissimilar from that of the reference ones. Recently, a few reference-guided methods [7], [8] were proposed for multimodal translation which generates diverse results by additionally taking reference images from the target domain as input. These methods, however, usually fail to fulfill the StyleFAT task and generate low-quality anime images.

Different from the image translation tasks of reference-guided methods, StyleFAT poses new challenges in two aspects. First, an anime-face usually has large eyes, a tiny nose, and a small mouth which are dissimilar from natural ones. The significant variations of shapes/appearances between anime-faces and photo-faces require translation methods to largely overdraw the local structures (e.g., eyes and mouth) of a photo-face, different from caricature translation [9] and makeup-face transfer [10]–[12] which preserve the identity of a photo-face. Since most reference-guided methods are designed to preserve the local structures/identity of a source image, these methods not only poorly transform the local shapes of facial parts into anime-like ones, but also fail to make these local shapes style-consistent with the reference anime-face. On the other hand, simultaneously transforming local shapes and transferring anime
styles is challenging and has not yet been well explored. Second, anime-faces involve various appearances and styles (e.g., various hair textures and drawing styles). Such large intra-domain variations pose challenges in devising a generator to translate a photo-face into a specific-style anime-face, as well as in training a discriminator to capture the distributions of anime-faces.

To address the above problems, we propose a novel GAN-based model called AniGAN for StyleFAT. First, since it is difficult to collect pairs of photo-faces and anime-faces, we train AniGAN with unpaired data in an unsupervised manner. Second, we propose a new generator architecture that preserves the global information (e.g., pose) of a source photo-face, while transforming local facial shapes into anime-like ones and transferring colors/textures based on the style of a reference anime-face. The proposed generator does not rely on face landmark detection or face parsing. Our insight is that the local shapes (e.g., large and round eyes) can be treated as a kind of styles like color/texture. In this way, transforming a face’s local shapes can be achieved via style transfer. To transform local facial shapes via style transfer, we explore where to inject the style information into the generator. In particular, the multi-layer feature maps extracted by the decoder represent multi-level semantics (i.e., from high-level structural information to low-level textural information). Our generator therefore injects the style information into the multi-level feature maps of the decoder. Through the guidance of the injected style information and feature maps with semantics of different levels, our generator adaptively learns to transfer color/texture styles and transform local facial shapes. Furthermore, two normalization functions are proposed for the generator to further improve local shape transformation and color/texture style transfer.

In addition to the generator, we propose a double-branch discriminator, that explicitly considers large appearance variations between photo-faces and anime-faces as well as variations among anime images. The double-branch discriminator not only learns domain-specific distributions by two branches of convolutional layers, but also learns the distributions of a common space across domains by shared shallow layers, so as to mitigate artifacts in generated faces. Meanwhile, a domain-aware feature matching loss is proposed to reduce artifacts of generated images by exploiting domain information in the branches.

Our major contributions are summarized as follows:

1) To the best of our knowledge, this is the first study on the style-guided face-to-anime translation task.
2) We propose a new generator to simultaneously transfer color/texture styles and transform the local facial shapes of a source photo-face into their anime-like counterparts based on the style of a reference anime-face, while preserving the global structure of the source photo-face.
3) We devise a novel discriminator to help synthesize high-quality anime-faces via learning domain-specific distributions, while effectively avoiding noticeable distortions in generated faces via learning cross-domain shared distributions between anime-faces and photo-faces.
4) Our new normalization functions improve the visual qualities of generated anime-faces in terms of transforming local shapes and transferring anime styles.

II. RELATED WORK

Generative Adversarial Networks. Generative Adversarial Networks (GANs) [1] have achieved impressive performance for various image generation and translation tasks [13]–[22]. The key to the success of GANs is the adversarial training between the generator and discriminator. In the training stage, networks are trained with an adversarial loss, which constrains the distribution of the generated images to be similar to that of the real images in the training data. To better control the generation process, variants of GANs, such as conditional GANs (cGANs) [13] and multi-stage GANs [14], [15], have been proposed. In our work, we also utilize an adversarial loss to constrain the image generation. Our model uses GANs to learn the transformation from a source domain to a significantly different target domain, given unpaired training data.

Image-to-Image Translation. With the popularization of GANs, GAN-based image-to-image translation techniques have been widely explored in recent years [2]–[6], [23]–[26]. For example, trained with paired data, Pix2Pix [4] uses a cGAN framework with an L1 loss to learn a mapping function from input to output images. Wang et al. proposed an improved version of Pix2Pix [5] with a feature matching loss for high-resolution image-to-image translation.

For unpaired data, recent efforts [2], [3], [6], [23], [24] have greatly improved the quality of generated images. CycleGAN [2] proposes a cycle-consistency loss to get rid of the dependency on paired data. UNIT [6] maps source-domain and target-domain images into a shared-latent space to learn the joint distribution between the source and target domains in an unsupervised manner. MUNIT [23] extends UNIT to multi-modal contexts by incorporating AdaIN [27] into a content and style decomposition structure. To focus on the most discriminative semantic parts of an image during translation, several works [3], [28] involve attention mechanisms. ContrastGAN [28] uses the object mask annotations from each dataset as extra input data. UGATIT [3] applies a new attention module and proposes an adaptive layer-instance normalization (AdaLIN) to flexibly control the amount of change in shapes and textures. However, the style controllability of the above methods is limited due to the fact that the instance-level style features are not explicitly encoded. To overcome this, FUNIT [24] utilizes a few-shot image translation architecture for controlling the categories of output images, but its stability is still limited.

Neural Style Transfer. StyleFAT can also be regarded as a kind of the neural style transfer (NST) [29]–[31]. In the field of NST, many approaches have been developed to generate paintings with different styles. For example, CartoonGAN [32] devises several losses suitable for general photo cartoonization. ChipGAN [33] enforces voids, brush strokes, and ink wash tone constraints to a GAN loss for Chinese ink wash painting style transfer. APDrawingGAN [34] utilizes a hierarchical GAN to produce high-quality artistic portrait
Fig. 2: Architecture of the proposed generator, which consists of a content encoder, a style encoder and a decoder to translate the source image to the output image reflecting the style of the reference image. The grey trapezoids indicate typical convolution blocks. More detailed notations are described in the contexts of Sec. III.

Fig. 3: Architecture of the proposed double-branch discriminator, where \( D_U \) denotes the shared layers between the two branches for domain \( X \) and \( Y \). The discriminator distinguishes real images from fake ones in the two individual branches.

drawings. CariGANs [9] and WarpGAN [35] design special modules for geometric transformation to generate caricatures. Yaniv et al. [36] proposed a method for geometry-aware style transfer for portraits utilizing facial landmarks. However, the above methods either are designed for a specific art field which is completely different from animation, or rely on additional annotations (such as facial landmarks).

III. PROPOSED APPROACH

Problem formulation. Let \( X \) and \( Y \) denote the source and target domains, respectively. Given a source image \( x \in X \) and a reference image \( y \in Y \), our proposed AniGAN learns multimodal mapping functions \( G : (x, y) \mapsto \hat{x} \) that transfer \( x \) into domain \( Y \).

To generate high-quality anime-faces for the StyleFAT task, the goal is to generate an anime-face \( \hat{x} \) that well preserves the global information (e.g., face pose) from \( x \) as well as reflects the styles (e.g., colors and textures) of reference anime-face \( y \), while transforming the shapes of facial parts such as eyes and hair into anime-like ones.

To achieve the above goals, a question is posed, **how to simultaneously transform local shape while transferring color/texture information?** Existing methods focus on transferring styles while preserving both local and global structures/shapes from the source image, which, however, cannot well address the above problem. Differently, we here explore **where** to inject style information into the generator, and a novel generator architecture is thereby proposed, as shown in Fig. 2. Different from existing methods which inject style information in the bottleneck of the generator, we introduce two new modules in the decoder of the generator for learning to interpret and translate style information. Furthermore, we also propose two normalization functions to control the style of generated images while transforming local shapes, inspired by recent work [3], [27].

In addition, anime-faces contain significant intra-variations, which poses large challenge for generating high-quality images without artifacts. To further improve the stability of the generated results, a novel double-branch discriminator is devised to better utilize the distribution information of different domains.

Figs. 2 and 3 illustrate the architectures of our proposed generator and discriminator, respectively. The generator takes a source image and a reference image as inputs and then learns to synthesize an output image. The double-branch discriminator consists of two branches, where one branch discriminates real/fake images for domain \( X \), and the other for \( Y \).

A. Generator

The generator of AniGAN consists of a content encoder \( E_c \), a style encoder \( E_s \) and a decoder \( F \), as shown in Fig. 2.

**Encoder.** The encoder includes a content encoder \( E_c \) and a style encoder \( E_s \). Given a source image \( x \), the content encoder \( E_c \) is used to encode the content of \( x \):

\[
\alpha = E_c(x). \tag{1}
\]

where \( \alpha \) is the content code encoded by the content encoder \( E_c \).

Existing works [7], [8], [23], [24] on style transfer have shown that a CNN-based encoder can capture the styles of an image, including color, textures, and drawing styles, etc. Following these works, we introduce a style encoder to extract the style codes from reference image \( y \). The function can be formulated as follows:

\[
(\gamma_s, \beta_s) = E_s(y) \tag{2}
\]

where \( \gamma_s \) and \( \beta_s \) are the style codes extracted from reference image \( y \).
The decoder $F$ constructs an image from the content and style codes. Our decoder aims to simultaneously transform the local shapes of facial parts and transfer reference anime styles into generated anime-faces, while preserving the global structure of the source photo-face. Recent image translation methods [8], [23], [27] have achieved success in transferring color/texture of the target domain into the generated images for various tasks such as street-scene translation. However, since these methods do not consider simultaneously transforming local shapes and transferring styles, their decoders cannot well handle the StyleFAT task. Directly employing the decoders in [8], [23], [27] is either insufficient to elaborately transfer an anime style, or introduces visually annoying artifacts in local facial parts.

One possible solution of transforming local facial features is to first employ face parsing or facial landmark detection to detect prominent facial features and then transform local facial features via face warping like [35]. However, since the local structures of anime-faces and photo-faces are significantly dissimilar to each other, face warping often leads to severe artifacts in the generated anime-faces. For example, it is difficult to well warp the mouth of a photo-face into a tiny one of a reference anime-face.

Instead of employing facial landmark detection or designing a complex network, our insight is that local structures can be treated as a kind of styles like color/texture and can be altered through style transfer. Therefore, we propose a simple yet effective architecture for the decoder by exploring two questions: (1) where to inject the style information? (2) how to inject the style information? For the former question, we propose an Adaptive Stack convolutional (ASC) block and Fine-grained Style Transfer (FST) block, while designing two normalization functions for the decoder (see Fig. 2) for the latter question.

1) ASC block: We first explore where to inject the style information. Existing methods such as MUNIT [23] and EGSC-IT [8] inject the style information into the residual blocks in the bottleneck of their decoder to transfer styles, where normalization layers are equipped with the residual blocks to receive and process injected style information. Such architecture remarkably transfers color/texture for various translation tasks. Inspired by existing methods [23] and EGSC-IT [8], ASC also injects style information into the bottleneck of the decoder via the normalization layers.

However, unlike existing methods, ASC does not employ residual blocks in the bottleneck of the decoder. Instead, ASC discards residual connections from the bottleneck and uses typical convolutions to decode content codes and style codes, since we observe that residual blocks may ignore some anime-style information (e.g. the color of right eye) and degrades the translation performance of the decoder for the StyleFAT task. More specifically, although multiple reference images with different anime styles are given, the decoder with residual blocks synthesizes similar styles for specific regions, especially on eyes. For example, the decoder with residual blocks improperly renders the right eyes with the same color in the generated images (see Fig. 7). We argue that the decoder would ignore some injected style information due to the residual operation. Therefore, compared with existing methods using residual blocks with normalization layers, ASC is a simpler architecture that only stacks convolutional layers and normalization layers\(^1\) without residual connections, as shown in Fig. 2. More implementation details of ASC and FST blocks are available in the supplemental materials.

2) FST block: ASC is able to transfer color/texture coarsely, however, is still insufficient to subtly transfer the style information and transform local shapes for the StyleFAT task. We argue the reason is that ASC only injects style information into the bottleneck of the decoder. In particular, a feature map of the bottleneck represents high-level semantic information, which is too high-level to capture low-level and fine information such as local structures of facial parts. As a result, using ASC solely would coarsely transfer color/texture and fail to properly transform the local shapes of facial parts into anime-like one.

To address the above issues, we propose the FST block to explicitly exploit feature maps with multiple-level information, such that our decoder subtly transfers styles and transforms local shapes of facial parts. Different from existing methods, FST injects style information into multiple deeper layers of the decoder rather than only the bottleneck. In particular, it is known that shallow- and deep-layer feature maps with low and high resolutions in the decoder represent different levels of semantic information such as high-level structures, middle-level information (e.g., local shapes of facial parts) and low-resolution textures. Motivated by the fact, we argue that FST can learn to adaptively fuse style information with different-level semantic information and transform local shape of facial parts, by injecting style information into the decoded feature maps representing multiple-level information.

Therefore, as shown in Fig. 2, FST consists of a stack of upsampling, convolutional, and normalization layers, similar to existing methods. However, different from the prior works that only inject style codes into the bottleneck of the decoder, style codes are injected after the upsampling layer in FST. Since feature maps decoded by convolutional layers coming after the upsampling layers are of different resolutions and contain multiple-level semantic information, FST enables the decoder to adaptively synthesize color/texture and transform local shapes, according to both the style codes and different levels of feature maps.

3) Normalization: We then explore how to inject the style information. Recent image translation methods [8], [23], [27] normalize the feature statistics of an image via Adaptive Instance Normalization (AdaIN) to adjust the color/texture styles of the image. However, we aim to devise normalization functions which not only transfer color/texture styles from the reference, but also transform local shapes of the source image based on the reference. Recently, it was shown in [3] that layer normalization (LN) [37] can transform the structure/shape of an image. Besides, AdaLIN was proposed in [3] to control the degree of changes in textures and shapes by adaptively

\(^1\)Note that ASC and FST only focus on the architecture (i.e., where to inject the style information). Style information is processed by the normalization layers e.g., [27]. In this paper, ASC and FST use our proposed normalization described below.
combining AdaIN and LN. However, AdaLIN is insufficient to simultaneously transfer the color/textures information of a local region and its shape information from a reference image to a generated image. That is, since AdaLIN combines IN and LN in a per-channel manner, AdaLIN ignores the correlations among channels. For example, the shape styles of eyes and their color/textures styles may respectively dominate in different channels. In such case, the features learned by AdaLIN often ignore shape styles or color/textures styles. In other words, the combination space of AdaLIN tends to be smaller than that of all-channel combinations of IN and LN.

To address the above issue, we propose two novel normalization functions called point-wise layer instance normalization (PoLIN) and adaptive point-wise layer instance normalization (AdaPoLIN) for the generator. Our PoLIN and AdaPoLIN learn to combine all channels of IN and LN, different from AdaLIN [3].

To achieve all-channel combination of instance normalization (IN) [38] and LN, PoLIN learns to combine IN and LN via a $1 \times 1$ convolutional layer as defined below:

$$\text{PoLIN}(z) = \text{Conv} \left( \begin{bmatrix} \frac{z - \mu_I(z)}{\sigma_I(z)} & \frac{z - \mu_L(z)}{\sigma_L(z)} \end{bmatrix} \right),$$

where $\text{Conv}(\cdot)$ denotes the $1 \times 1$ convolution operation, $[\cdot, \cdot]$ denotes the channel-wise concatenation, $z$ is the the feature map of a network layer, $\mu_I$, $\mu_L$, and $\sigma_I$, $\sigma_L$ denote the channel-wise and layer-wise means and standard deviations, respectively.

AdaPoLIN adaptively combines IN and LN, while employing the style codes from the reference anime-faces to retain style information:

$$\text{AdaPoLIN}(z, \gamma_s, \beta_s) = \gamma_s \cdot \text{Conv} \left( \begin{bmatrix} \frac{z - \mu_I(z)}{\sigma_I(z)} & \frac{z - \mu_L(z)}{\sigma_L(z)} \end{bmatrix} \right) + \beta_s,$$

where $\gamma_s$ and $\beta_s$ are style codes, and the bias in $\text{Conv}(\cdot)$ is fixed to 0.

Thanks to their all-channel combination of IN and LN, the proposed PoLIN and AdaPoLIN lead to a larger combination space than AdaLIN, thereby making them beneficial to handle color/textures style transfer and local shape transformation for StyleFAT.

### B. Discriminator

It is challenging to design a discriminator which effectively distinguishes real anime-faces from fake ones for StyleFAT. In particular, both the appearances and shapes vary largely among anime-faces, leading to significant intra-variations in the distribution of anime-faces. Thus, it is difficult for a typical discriminator (e.g. [2]) to well learn the distribution of anime-faces. As a result, the generated anime-faces may contain severely distorted facial parts and noticeable artifacts.

To address the above issues, we propose a double-branch discriminator. In particular, we assume that anime-faces and photo-faces partially share common distributions and such cross-domain shared distributions constitute meaningful face information, since these two domains are both about human faces. In other words, by learning and utilizing the cross-domain shared distributions, the discriminator can help reduce distortions and artifacts in translated anime-faces. Therefore, as shown in Fig. 3, the proposed double-branch discriminator consists of shared shallow layers followed by two domain-specific output branches: one branch for distinguishing real/fake anime-faces and the other for distinguishing real/fake photo-faces. With the Siamese-like shallow layers shared by the photo-face and anime-face branches, the additional photo-face branch aims to assist the anime-face branch to learn domain-shared distributions. As a result, the anime-face branch learns to effectively discriminate those generated anime-faces with distorted facial parts or noticeable face artifacts. On the other hand, each branch contains additional domain-specific deep layers with an extended receptive field to individually learn the distributions of anime-faces and photo-faces.

We formulate the two-branch discriminator in terms of domain $X$ and domain $Y$ for generality. Let $D_X$ and $D_Y$ denote the discriminator branches corresponding to domain $X$ and $Y$, respectively, and $D_U$ denote the shallow layers shared by $D_X$ and $D_Y$. An input image $h$ is discriminated either by $D_X$ or by $D_Y$ according to the domain that $h$ belongs to. The discriminator function is formulated as follows:

$$D(h) = \begin{cases} D_X(D_U(h)) & \text{if } h \in X, \\ D_Y(D_U(h)) & \text{if } h \in Y. \end{cases}$$

Our discriminator helps significantly improve the quality of generated images and the training stability of the generator, since it not only individually learns domain-specific distributions using separable branches, but also learns domain-shared distributions across domains using shared shallow layers. In addition, our discriminator is scalable and can be easily expended to multiple branches for tasks across multiple domains.

### C. Loss Functions

StyleFAT is different from typical image translation tasks which preserve the identity or the whole structure of the input photo-face. If we directly employ loss functions in existing methods [8], [9] that preserves both local and global structures or an identity in the source image, the quality of a generated anime-face would be negatively affected. Instead, besides adversarial loss like in [8], [10], the objective of our model also additionally involves reconstruction loss, feature matching loss, and domain-aware feature matching loss.

**Adversarial loss.** Given a photo-face $x \in X$ and a reference anime-face $y \in Y$, the generator aims to synthesize from $x$ an output image $G(x, y)$ with a style transferred from $y$. To this end, we adopt an adversarial loss similar to [13], [24] as follows:

$$L_{\text{adv}} = \mathbb{E}_x[\log D_X(D_U(x))] + \mathbb{E}_{x,y} [\log (1 - D_X(D_U(G(x, y)))]$$

$$+ \mathbb{E}_y [\log D_Y(D_U(y))] + \mathbb{E}_{y,x} [\log (1 - D_Y(D_U(G(x, y)))).$$

Two separate discriminators without shared shallow layers can also individually learn the distributions of anime-faces and photo-faces. However, we observe that such a design not only consumes more computational cost but also performs worse than our double-branch discriminator.
Feature matching loss. To encourage the model to produce natural statistics at various scales, the feature matching loss [5], [39] is utilized as a supervision for training the generator. Formally, let $D^k_U(h)$ denote the feature map extracted from the $k$-th down-sampled version of the shared layers $D_U$ of $D_X$ or $D_Y$ for input $h$, and $D^h_V(h)$ denote the global average pooling result of $D^h_U(h)$, the feature matching loss $L_{fm}$ is formulated as below:

$$L_{fm} = \mathbb{E}_h[\sum_{k \in K_1} ||D^k_U(h) - D^k_V(G(h, h))||_1],$$  \hspace{1cm} (7)

where $K_1$ denotes the set of selected layers in $D_U$ used for feature extraction. In our work we set $K_1 = \{1, 2\}$ according to the architecture of our discriminator.

Domain-aware feature matching loss. We further utilize the domain-specific information to optimize the generator. In particular, we extract features by the domain-specific discriminator $D_X$ or $D_Y$, respectively. Similar to $D^k_U(\cdot)$, let $D^k_X(\cdot)$ denote the $k$-th down-sampling feature map extracted from the branch $D_X$ in domain $X$, and $D^h_X(\cdot)$ denote the average pooling result of $D^h_X(\cdot)$ (similar notations are used for domain $Y$). Then we define a domain-aware feature matching loss $L_{dfm}$ as follows:

$$L_{dfm} = \begin{cases} \mathbb{E}_h[\sum_{k \in K_2} ||\bar{D}^k_X(D_U(h)) - \bar{D}^k_Y(D_U(G(h, h)))||_1], & \text{if } h \in X \\ \mathbb{E}_h[\sum_{k \in K_2} ||\bar{D}^k_Y(D_U(h)) - \bar{D}^k_Y(D_U(G(h, h)))||_1], & \text{if } h \in Y \end{cases},$$  \hspace{1cm} (8)

where $K_2$ represents the set of selected layers in $D_X$ and $D_Y$ used for feature extraction. In our work we set $K_2 = \{3\}$ according to the architecture of our discriminator. With $L_{dfm}$, the artifacts of generated images can be largely mitigated thanks to the additional domain-specific features.

Reconstruction loss. We aim to preserve the global semantic structure of a source photo-face. Without a well-designed loss on the discrepancy between the generated anime-face and source photo-face, the global structure information of the source photo-face may be ignored or distorted. However, as discussed previously, we cannot directly employ an identity loss to preserve the identity like [10] or a perceptual loss to preserve the structure of the image like [8].

Different from existing methods [8], [10], we impose a reconstruction loss to preserve the important information of photo-face. Specifically, given a source photo-face $x \in X$, we also use $x$ as the reference to generate a face $G(x, x) = F(E_c(x), E_s(x))$, where $E_c(x)$ and $E_s(x)$ represent the content and style information extracted from $x$, respectively. The reconstruction loss then constrains $G(x, x)$ to reconstruct $x$ as faithfully as possible:

$$L_{rec}(x) = ||G(x, x) - x||_1 = ||F(E_c(x), E_s(x)) - x||_1.$$  \hspace{1cm} (9)

where $||G(x, x) - x||_1$ measures the dissimilarity between $x$ and $G(x, x)$. We argue that the reconstruction loss encourages $E_c(x)$ to capture essential content information from a source photo-face and enforce $F(E_c(x), E_s(x))$ to preserve the captured content information, since $G(x, x)$ reconstructs only "x", instead of another photo-face.

Note that the reconstruction loss also encourages transferring the style feature $E_s(x)$ extracted from $x$ into the generated image $G(x, x)$. To further enforce learning the anime-styles and transferring them into generated anime-faces, we apply the reconstruction loss to anime-face $y$. Therefore, the total reconstruction loss is defined as follows:

$$L_{rec} = \mathbb{E}_x[||G(x, x) - x||_1] + \mathbb{E}_y[||G(y, y) - y||_1].$$  \hspace{1cm} (10)

With the total reconstruction loss, our method is encouraged to effectively preserve the global information of the source photo-face and transfer anime styles of the reference anime-face.

Full Objective. Consequently, we combine all the above loss functions as our full objective as follows:

$$L_G = L_{adv} + \lambda_{rec} \cdot L_{rec} + \lambda_{fm} \cdot (L_{fm} + L_{dfm}),$$
$$L_D = -L_{adv}.$$  \hspace{1cm} (11)

where $\lambda_{rec}, \lambda_{fm}$ are hyper-parameters to balance the losses.

IV. Experimental Results

We first conduct qualitative and quantitative experiments to evaluate the performance of our approach. We then conduct user studies to evaluate the subjective visual qualities of generated images. Finally, we evaluate the effectiveness of proposed modules by ablation studies.

Baselines We compare our method with CycleGAN [2], UGATIT$^5$ [3], and MUNIT [23], the state-of-the-arts in reference-free image translation. In addition, we also compare our method with reference-guided image translation schemes including FUNIT [24], EGSC-IT [8], and DRIT++ [7] which are most relevant to our method. Note that MUNIT also allows to additionally take a reference image as the input at the testing stage and transfer its styles. We refer to this baseline as RG-MUNIT. However, instead of using a reference image as part of a training image pair, RG-MUNIT takes a random latent code from the target domain as style information for the training (see more details in its github code$^4$). Due to such inadequate reference information during training, while performing StyleFAT with reference faces, RG-MUNIT tends to fail for face images with significant inter-variations between domains or data with large intra-variations within a domain. That is, RG-MUNIT performs much poorer than the original MUNIT that takes a random latent code from the target domain at the test stage. We hence only show the visual results with the original MUNIT in the qualitative comparisons, while evaluating quantitative results for both the original MUNIT and RG-MUNIT. We train all the baselines based on the open-source implementations provided by the original papers$^5$.

$^4$We used the full version implementation of UGATIT provided by its authors

$^5$https://github.com/NVlabs/MUNIT

$^5$As discussed in Section II, since neural-style transfer methods focus on specific art-style translation, we do not compare these methods for fair comparison.
for all experiments, \( \lambda \) only contains a small number of male characters. Face pose) from the source is preserved. Features are anime-like and the global information (e.g., the reference without introducing noticeable artifacts, while facial lar/consistent anime-styles (e.g., color and texture) with the good translation result for StyleFAT task should share simi-

C. Qualitative comparison

We follow the setup in UGATIT [3] and use the selfie2anime dataset to evaluate our proposed AniGAN. For the dataset, only female character images are selected and monochrome anime images are removed manually. Both selfie and anime images are separated into a training set with 3,400 images and a test set with 100 images.

Face2anime. We build an additional dataset called face2anime, which is larger and contains more diverse anime styles (e.g., face poses, drawing styles, colors, hairstyles, eye shapes, strokes, facial contours) than selfie2anime, as illustrated in Fig. 4. The face2anime dataset contains 17,796 images in total, where the number of both anime-faces and natural photo-faces is 8,898. The anime-faces are collected from the Danbooru2019 [40] dataset, which contains many anime characters with various anime styles. We employ a pretrained cartoon face detector [41] to select images containing anime-faces\(^4\). For natural-faces, we randomly select 8,898 female faces from the CelebA-HQ [14], [42] dataset. All images are aligned with facial landmarks and are cropped to size 128 \( \times \) 128. We separate images from each domain into a training set with 8,000 images and a test set with 898 images.

B. Implementation Details

We train and evaluate our approach using the face2anime and selfie2anime datasets. We use the network architecture mentioned in Section III as our backbone. We set \( \lambda_{\text{im}} = 1 \) for all experiments, \( \lambda_{\text{rec}} = 1.2 \) for the face2anime dataset, and \( \lambda_{\text{rec}} = 2 \) for the selfie2anime dataset.

For fast training, the batch size is set to 4 and the model is trained for \( 10^5 \) iterations. The training time is less than 14h on a single Tesla V100 GPU with our implementation in PyTorch [43]. We use RMSProp optimizer with a learning rate of \( 10^{-4} \).

C. Qualitative comparison

Given a source photo-face and a reference anime-face, a good translation result for StyleFAT task should share similar/consistent anime-styles (e.g., color and texture) with the reference without introducing noticeable artifacts, while facial features are anime-like and the global information (e.g., the face pose) from the source is preserved.

\(^4\)Images containing male characters are discarded, since Danbooru2019 only contains a small number of male characters.

Fig. 4: Example photo-faces and anime-faces of our face2anime dataset. From top to bottom: photo-faces and anime-faces.

Fig. 5 illustrates qualitative comparison results on face2anime dataset, where the photo-faces involve various identities, expressions, illuminations, and poses, whereas the reference anime-faces present various anime-styles. Compared with cat2dog and horse2zebra dataset, face2anime dataset contains larger variations of shape/appearance among anime-faces, whose data distribution is much more complex and is challenging for image translation methods.

The results show that CycleGAN introduces visible artifacts in their generated anime-faces (see the forehead in the fourth row). MUNIT also leads to visible artifacts in some generated anime-faces, as shown in 3rd and seventh row in Fig. 5. UGATIT better performs than CycleGAN and MUNIT. However, the anime styles of generated anime-faces by CycleGAN, UGATIT and MUNIT are dissimilar to that of the references. FUNIT is designed for few-shot reference-guided translation, and hence is not suitable for the StyleFAT task. Consequently, the style of generated anime-faces by FUNIT is much less consistent with the references compared with our method.

Although EGSC-IT usually well preserves the poses of photo-faces, it also attempts to preserve the local structures of photo-faces, which often conflicts with the transfer of anime styles, since the local shapes of facial parts like eyes and mouth in an anime-face are dissimilar to the counterparts of the corresponding photo-face. Consequently, EGSC-IT often leads to severe artifacts in the local structures of generated anime-faces (see eyes and hair in the first to third rows). DRIT++ also introduces artifacts into generated faces, when transferring the styles of the reference anime-faces. For example, DRIT++ generates two mouths in the generated face in the third row and distort eyes in the fifth row.

Outperforming the above state-of-art methods, our method generates the highest-quality anime-faces. First, compared with reference-guided methods FUNIT, EGSC-IT, and DRIT++, the styles of generated anime-faces with our method are the most consistent with that of reference faces, thanks to our well-designed generator. Moreover, our method well preserves the poses of photo-faces although our method does not use perceptual loss like EGSC-IT. Our method also well converts local structures like eyes and mouth into anime-like ones without introducing clearly visible artifacts.

Fig. 6 compares the results of various methods on the selfie2anime dataset. The results show that FUNIT introduces artifacts into some generated faces, as shown in the fourth, sixth and seventh rows in Fig. 6. Besides, EGSC-IT tends...
| Reference | Source | CycleGAN | UGATIT | MUNIT | FUNIT | DRIT++ | EGSC-IT | Ours |
|-----------|--------|----------|--------|-------|-------|--------|--------|------|

Fig. 5: Comparison of various image translation methods on the face2anime dataset. From left to right: source photo-face, reference anime-face, the results by CycleGAN [2], MUNIT [23], UGATIT [3], FUNIT [24], DRIT++ [7], EGSC-IT [8] and our AniGAN.

to generate anime-faces with similar styles, although the reference anime-faces are of different styles (see the reference and generated images in the fourth to eighth rows in Fig. 6). Similarly, DRIT++ tends to synthesize eyes with similar styles in the generated faces. For example, the synthesized eyes by DRIT++ are orange ellipse in the first, second, fourth, fifth, sixth, and eighth rows in Fig. 6. In contrast, our method generates anime-faces reflecting the various styles of the reference images. In other words, our method achieves the most consistent styles with those of reference anime-faces over the other methods. In addition, our method also generates high-quality faces which preserve the poses of source photo-faces, despite a photo-face is partially occluded by other objects (e.g., the hand and the phone in the fifth and sixth rows of Fig. 6).

D. Quantitative Comparisons

In addition to qualitative evaluation, we quantitatively evaluate the performance of our method in two aspects. One is the visual quality of generated images, and the other is the translation diversity.

Visual quality. We evaluate the visual quality of our results with Frechet Inception Distance (FID) metric [44] which has been popularly used to evaluate the quality of synthetic images in image translation works e.g., [24], [34], [45]. The FID score evaluates the distribution discrepancy between the real faces and synthetic anime-faces. A lower FID score indicates
that the distribution of generated images is more similar to that of real anime-faces. That is, those generated images with lower FID scores are more plausible as real anime-faces. Following the steps in [44], we compute a feature vector by a pretrained network [46] for each real/generated anime-face, and then calculate FID scores for individual compared methods, as shown in Table I. The FID scores in Table I demonstrate that our our AniGAN achieves the best scores on both the face2anime and selfie2anime datasets, meaning that the anime-faces generated by our approach have the closest distribution with real anime-faces, thereby making them look similar visually.

**Translation diversity.** For the same photo-face, we evaluate whether our method can generate anime-faces with diverse styles, given multiple reference anime-faces with different styles. We adopt the learned perceptual image patch similarity (LPIPS) metric, a widely adopted metric for assessing translation methods on multimodal mapping [7], [45] in the perceptual domain, for evaluating the translation diversity. Following [45], given each testing photo-face, we randomly sample 10 anime-faces as its reference images and then generate 10 outputs. For these 10 outputs, we evaluate the LPIPS scores between every two outputs. Table II shows the average of pairwise LPIPS over all testing photo-faces. A higher LPIPS score indicates that the translation method generates images with larger diversity.

Since CycleGAN, UGATIT focus on one-to-one mapping and cannot generate multiple outputs given a source image, we do not include them for comparison of translation diversity. We also do not compare with MUNIT and FUNIT, since MUNIT does not take a reference as the input and FUNIT focuses on few-shot learning instead of translation diversity. Instead, we compare with DRIT++ and EGSC-IT, which are state-of-the-arts in reference-guided methods. DRIT++ uses a
TABLE III: Comparison of preference percentages of subjects to the results generated by four methods in user study: higher is better.

| Method   | User study 1 | User study 2 |
|----------|--------------|--------------|
|          | Face2anime   | Selfie2anime | Face2anime   | Selfie2anime |
| FUNIT    | 17.50%       | 40.17%       | 19.17%       | 33.17%       |
| DRIT++   | 68.17%       | 47.83%       | 59.83%       | 60.50%       |
| EGSC-IT  | 27.33%       | 30.67%       | 32.17%       | 22.33%       |
| Ours     | 87.00%       | 81.33%       | 88.83%       | 84.00%       |

regularization loss which explicitly encourages the generator to generate diverse results. Although our method does not impose such loss, LPIPS scores in Table II shows that our method outperforms DRIT++ and EGSC-IT on translation diversity, thanks to the generator and discriminator of our method.

E. User Study

We conduct two subjective user studies to further evaluate our method. Following [5], [23], we adopt pairwise A/B test.

User study 1. The first user study evaluates the ability of our method in transferring the style of a reference anime-face and preserving the structure of a source photo-face. 20 subjects are invited to participate in our experiments, whose ages range from 22 to 35. For each subject, we show a source photo-face, a reference anime-face, and two anime-faces generated by two different translation methods. The generated anime-faces are presented in a random order, such that subjects are unable to infer which anime-faces are generated by which translation methods. We then ask each subject the following question:

"Q: Which generated anime-faces has better visual quality by considering the source photo-face and the anime styles of the reference anime-face?"

We compare our method with three reference-guided methods FUNIT, EGSC-IT and DRIT++. Since translation methods (e.g., CycleGAN and UGATIT) do not transfer the information of a reference image, we do not compare with these methods for fair comparison. The subjective user study is conducted for face2anime and selfie2anime dataset, respectively, where 10 pairs of photo-faces and anime-faces in each dataset are fed into these four translation methods to generate anime-faces.

We receive 1,200 answers from 20 subjects in total for each dataset, where each method is compared 600 times. As shown in Table III, most subjects are in favor of our method for the results on both face2anime and selfie2anime datasets, demonstrating that the anime-faces translated by our method are usually the most visually appealing to the subjects.

User study 2. The second user study evaluates the effectiveness of our method on generating plausible anime-faces, where 20 subjects are invited. By presenting a source photo-face and two translated anime-faces generated by two different translation methods, we ask each subject a question:

"Q2: which anime-face better represents the original photo-face in an anime style?"

We conduct this user study on anime-faces generated from 10 pairs of source photo-faces and reference anime-faces. Table III shows the user study results on face2anime and selfie2anime, respectively. Our method outperforms the other methods, indicating that our method generates the most plausible anime-faces.

F. Ablation Study

We conduct ablation experiments to validate the effectiveness of individual components in our method: (1) ASC block, (2) FST block, (3) PLIN and AdaPLIN, and (4) double-branch discriminator.

ASC block. We seek to validate whether the ASC block effectively retains the style information of the reference image and helps the generator transfer the style characteristics. Note that the key differentiating factor in our ASC block is the removal of residual blocks from the bottleneck of our generator, different from start-of-the-art image translation methods (e.g., MUNIT, UGATIT, and FUNIT). We hence implement a baseline called "w/o ASC" which adds residual blocks in the bottleneck of our decoder. As shown in Fig. 7, "w/o ASC" improperly renders "brown" right eyes in all the generated images.

Fig. 7: Visual comparison of the contributions of ASC blocks, where “w/o ASC” improperly renders “brown” right eyes in all the generated images.

Fig. 8: Visual comparison of the contributions of FST blocks

Reference  Source  w/o FST  Ours
TABLE IV: Quantitative comparison for ablation study using FID score. Lower is better.

| Method          | ASC | FST | DB  | PoLIN | AdaPoLIN | IN | LIN | AdaIN | AdaLIN | Face2anime | Selfie2anime |
|-----------------|-----|-----|-----|-------|-----------|----|-----|-------|--------|-------------|--------------|
| w/o ASC         | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 4.05        | 96.20        |
| w/o FST         | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 4.13        | 99.91        |
| w/o DB          | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 4.56        | 92.78        |
| w/o PoLIN w IN  | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 4.73        | 89.62        |
| w/o PoLIN w LIN | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 3.90        | 90.66        |
| w/o AdaPoLIN w AdaIN | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 4.16 | 90.31 |
| w/o AdaPoLIN w AdaLIN | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 3.92 | 91.93 |
| AniGAN (Ours)   | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 38.45       | 86.98        |

TABLE V: Quantitative comparison for ablation study using LPIPS score. Higher is better.

| Method          | ASC | FST | DB  | PoLIN | AdaPoLIN | IN | LIN | AdaIN | AdaLIN | Face2anime | Selfie2anime |
|-----------------|-----|-----|-----|-------|-----------|----|-----|-------|--------|-------------|--------------|
| w/o ASC         | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 0.357       | 0.321        |
| w/o FST         | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 0.391       | 0.340        |
| w/o DB          | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 0.395       | 0.342        |
| w/o PoLIN w IN  | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 0.409       | 0.362        |
| w/o PoLIN w LIN | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 0.402       | 0.367        |
| w/o AdaPoLIN w AdaIN | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 0.400 | 0.356 |
| w/o AdaPoLIN w AdaLIN | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | 0.397 | 0.336 |
| AniGAN (Ours)   | ✓   | ✓   | ✓   | ✓     | ✓         | ✓  | ✓   | ✓     | ✓      | 0.414       | 0.372        |

Fig. 9: Visual comparison of the contributions of the double-branch discriminator

Reference | Source | w/o DB | Ours

eyes in all the generated images. In contrast, our method well and consistently renders the style of the left and right eyes, despite no face landmarks or face parsing are used. Clearly, our method outperforms “w/o ASC” in transferring the styles of reference anime-faces, thanks to the ASC block.

**FST block.** Different from existing methods (e.g., MUNIT, FUNIT, EGSC-IT) which use typical upsampling blocks in the decoder, FST is additionally equipped with our normalization functions. To validate the effectiveness of FST, we build a baseline (called “w/o FST”) which replaces the FST blocks with typical upsampling blocks (like MUNIT, FUNIT, and EGSC-IT) without our normalization functions. As shown in Fig. 8, “w/o FST” performs poorly in converting the shape of local facial features and transferring styles. For example, “w/o FST” poorly converts the face shapes and eyes and introduces artifacts in the generated faces. In contrast, with FST, our method better converts the local shapes into anime-like ones than “w/o FST”. Moreover, our method also better transfers the styles of the reference anime-faces to the generated faces than “w/o FST” does. Similarly, the FID of “w/o FST” increases in Table IV, indicating that the anime-faces generated by “w/o FST” are less plausible than those generated by our method. In addition, the LPIPS scores of “w/o FST” and our method in Tab. V shows that FST is helpful for generating diverse anime-faces.

**PoLIN and AdaPoLIN.** We build four baselines to evaluate the effectiveness of PoLIN and AdaPoLIN. The first and second baselines are built for evaluating PoLIN, and the third and fourth baseline are for AdaPoLIN. The first baseline, named “w/o PoLIN w/ IN”, is constructed by replacing PoLIN with IN in [38]. We build “w/o PoLIN w/ IN”, since EGSC-IT, DRIT and FUNIT employ IN in the upsampling convolutional layers of their decoder, different from our method with PoLIN. The second baseline, named “w/o PoLIN w/ LIN”, is constructed by replacing PoLIN with layer-Instance Normalization (LIN). The third baseline, called “w/o AdaPoLIN w/ AdaIN”, replaces AdaPoLIN with AdaIN in [27], which was employed by many translation methods e.g., [8], [23], [24], [47]. The fourth baseline, called “w/o AdaPoLIN w AdaLIN”, replaces AdaPoLIN with AdaLIN which is used in UGATIT.

The FID scores in Table IV show that our method outperforms the four baselines. Without PoLIN, the performance of transforming local shapes into anime-like ones is degraded, as shown in the results generated by “w/o PoLIN w/ IN” and “w/o PoLIN w/ LIN” in Fig. 10. For example, “w/o PoLIN w/ IN” introduces artifacts in the hair at the right boundary of generated faces in the first row of Figs. 10(c) and (d). Similarly, without AdaPoLIN, both “w/o AdaPoLIN w AdaIN” and “w/o AdaPoLIN w AdaLIN” perform worse than our method. For example, compared with our method, the styles (e.g., color) of eyes synthesized by these two baselines are less consistent with those of the reference faces in the second row in Figs. 10(e) and (f). Table IV shows that “w/o AdaPoLIN w AdaIN” and “w/o AdaPoLIN w AdaLIN” degrade the performance in terms of translation diversity.

It is worth noting that all the above baselines, that replace our normalization functions with other normalization functions
TABLE VI: FID scores of our method using different value of $\lambda_{\text{rec}}$ on face2anime: lower is better.

| $\lambda_{\text{rec}}$ | 0.4 | 0.8 | 1.2 | 1.6 | 2  |
|------------------------|-----|-----|-----|-----|----|
| FID                    | 63.56 | 40.83 | 38.42 | 39.17 | 39.21 |

TABLE VII: FID scores of our method using different value of $\lambda_{\text{fm}}$ on face2anime: lower is better.

| $\lambda_{\text{fm}}$ | 0.5 | 1   | 1.5 | 2   | 2.5 |
|------------------------|-----|-----|-----|-----|-----|
| FID                    | 40.01 | 38.35 | 38.68 | 38.63 | 39.15 |

employed by DRIT, EGSC-IT, UGATIT, etc., still achieve better FID and LPIPS scores than state-of-the-art methods, as shown in Table I, II, IV, and V. This indicates that the architectures of our generator and discriminator are more advantageous for StyleFAT task.

**Double-branch discriminator.** We implement a baseline “w/o DB” that removes the photo-face branch (i.e., the branch that discriminates real/fake photo-faces) from our discriminator. Table IV shows that it yields poorer FID than our method. Table V also shows LPIPS scores of “w/o DB” is worse than that of our method. More specifically, as shown in Fig. 9, “w/o DB” distorts local facial shapes, leading to low-quality generated faces, especially for challenging source photo-faces. This is because “w/o DB” would mainly focus on generating plausible anime images, rather than on generating plausible anime human faces. In contrast, our method generates high-quality anime-faces, thanks to the additional photo-face branch in the discriminator. With the photo-face branch in our discriminator, the photo-face and anime-face branches share the first few shallow layers, which helps the anime-face branch better learn real facial features from photo-faces so as to well discriminate low-quality anime-faces with distorted facial features.

**G. Parameter Tuning**

There are two parameters $\lambda_{\text{rec}}$ and $\lambda_{\text{fm}}$ in the objective function formulated in Eq. (11). We set various values of parameters $\lambda_{\text{rec}}$ and $\lambda_{\text{fm}}$ to analyze their influence on the performance of our method.

$\lambda_{\text{rec}}$ is to control the strength of the reconstruction loss in (11) during training, where the reconstruction loss encourages our method to preserve the global structure of the source photo-face and transfer the styles of the reference anime-face.

$\lambda_{\text{fm}}$ is used to control the strength of domain-aware feature matching loss and feature matching loss in Eq. (11). Like $\lambda_{\text{rec}}$, the value of $\lambda_{\text{fm}}$ cannot be too small. Nevertheless, the overall performance of our method is not sensitive to the value of $\lambda_{\text{fm}}$, when the value is around 1.5. In particular, we uniformly set $\lambda_{\text{fm}} = 0.5$, 1, 1.5, 2, 2.5 to test the impact of $\lambda_{\text{fm}}$ on the performance of our method. As shown in Table VII, the overall performance of our method does not vary much with different values of $\lambda_{\text{fm}}$. We set $\lambda_{\text{fm}} = 1$ for both face2anime and selfie2anime datasets in our experiments.

**H. Discussions and Limitations**

Our method can be extended to other translation tasks especially for those involving large inter-domain variances of shapes between domains. In this paper, we test our method on an animal-face translation task which transfers a dog-face into a cat-face given a reference cat-face, where the training...
In this paper, we proposed a novel GAN-based method, called AniGAN, for style-guided face-to-anime translation. On top of AniGAN, we also proposed a new generator architecture and two normalization functions, that effectively transfer styles from the reference anime-face, preserve global information extracted from the source photo-face and convert local facial shapes into anime-like ones. We have also proposed a double-branch discriminator to assist the generator to produce high-quality anime-faces. Extensive experiments demonstrate that our method achieves superior performance compared with state-of-the-art methods.

**References**

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proc. IEEE/CVF Conf. Comput. Vis.*, 2014.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proc. IEEE/CVF Conf. Comput. Vis.*, 2017.

[3] J. Kim, M. Kim, H. Kang, and K. H. Lee, “U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation,” in *Proc. Int. Conf. Learn. Rep.*, 2020.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017.

[5] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018.

[6] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[7] H. Lee, H.-Y. Tseng, Q. Mao, J.-B. Huang, Y.-D. Lu, M. Singh, and M.-H. Yang, “Drit++: Diverse image-to-image translation via disentangled representations,” *Int. J. Comput. Vis.*, pp. 1–16, 2020.

[8] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. Van Gool, “Exemplar guided unsupervised image-to-image translation with semantic consistency,” in *ICLR*, May 2019.

[9] K. Cao, J. Liao, and L. Yuan, “Carigans: Unpaired photo-to-caricature translation,” *ACM Trans. Graphics*, 2018.

[10] T. Li, R. Qian, C. Dong, S. Liu, Q. Yan, W. Zhu, and L. Lin, “Beautygan: Instance-level facial makeup transfer with deep generative adversarial network,” *Proc. ACM Int. Conf. Multimedia*, 2018.

[11] H.-J. Chen, K.-M. Hui, S.-Y. Wang, L.-W. Tsao, H.-H. Shuai, and W.-H. Cheng, “Beautyflow: On-demand makeup transfer framework with reversible generative network,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10,042–10,050.

[12] W. Jiang, S. Liu, C. Gao, J. Cao, R. He, J. Feng, and S. Yan, “Pegan: Pose and expression robust spatial-aware gan for customizable makeup transfer,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, June 2020.

[13] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1406.0577*, 2014.

[14] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” *Proc. Int. Conf. Learn. Rep.*, 2018.

[15] T. Karras, T. Aila, and T. A. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4401–4410.

[16] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” *arXiv preprint arXiv:1805.08318*, 2018.

[17] A. Brock, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis,” in *Proc. Int. Conf. Learn. Rep.*, 2019.

[18] C. Yang, T. Kim, R. Wang, H. Peng, and C.-C. J. Kuo, “Show, attend, and translate: Unsupervised image translation with self-regularization and attention,” *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4845–4856, 2019.

[19] Y. Song, C. Yang, Z. Lin, X. Liu, Q. Huang, H. Li, and C.-C. Jay Kuo, “Contextual-based image inpainting: Infer, match, and translate,” in *Proc. European Conf. Comput. Vis.*, 2018, pp. 3–19.

[20] L. Chen, L. Wu, Z. Hu, and M. Wang, “Quality-aware unpaired image-to-image translation,” *IEEE Trans. on Multimedia*, vol. 21, no. 10, pp. 2664–2674, 2019.

[21] Z. Li, C. Deng, E. Yang, and D. Tao, “Staged sketch-to-image synthesis via semi-supervised generative adversarial networks,” *IEEE Trans. on Multimedia*, 2020.

[22] Y. Liu, W. Chen, L. Liu, and M. S. Lew, “Swappgan: A multistage generative approach for person-to-person fashion style transfer,” *IEEE Trans. on Multimedia*, vol. 21, no. 9, pp. 2209–2222, 2019.
[23] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” in Proc. European Conf. Comput. Vis., 2018.

[24] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, “Few-shot unsupervised image-to-image translation,” in Proc. IEEE/CVF Conf. Comput. Vis., 2019, pp. 10551–10560.

[25] 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 Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018.

[26] C. Yang and S.-N. Lim, “One-shot domain adaptation for face generation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 5921–5930.

[27] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1501–1510.

[28] Y. Chen, Y.-K. Lai, and Y.-J. Liu, “Cartoongan: Generative adversarial networks for photo cartoonization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 9465–9474.

[29] B. He, F. Gao, D. Ma, B. Shi, and L.-Y. Duan, “Chipgan: A generative adversarial network for chinese ink wash painting style transfer,” in Proc. ACM Int. Conf. Multimedia, 2018, pp. 1172–1180.

[30] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, “Adpavinggan: Generating artistic portrait drawings from face photos with hierarchical gans,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 10743–10752.

[31] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. European Conf. Comput. Vis., 2016, pp. 694–711.

[32] J. Yaniv, Y. Newman, and A. Shamir, “The face of art: landmark detection and geometric style in portraits,” ACM Trans. Graphics, vol. 38, no. 4, p. 60, 2019.

[33] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6626–6637.

[34] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826.