CariGANs: Unpaired Photo-to-Caricature Translation

KAIDI CAO*[†], Tsinghua University
JING LIAO[‡], City University of Hong Kong, Microsoft Research
LU YUAN, Microsoft AI Perception and Mixed Reality

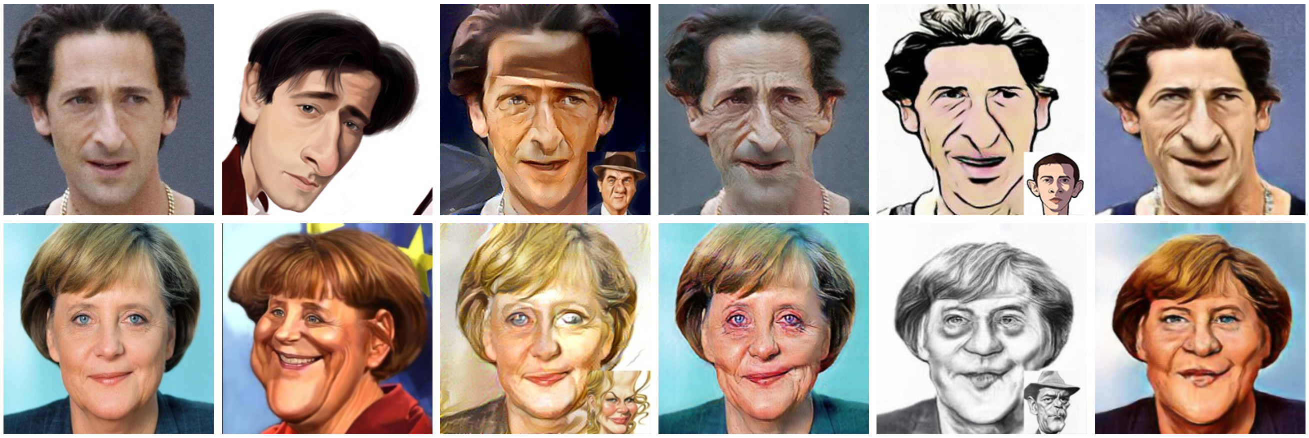

Fig. 1. Comparison of the caricature drawn manually (b) and generated automatically with neural style transfer [Gatys et al. 2015] (c), CycleGan [Zhu et al. 2017a] (d), and Our CariGANs with a given reference (e) or a random noise (f). Please note networks used in (d)(e)(f) are trained with the same dataset. And the reference used in the result is overlaid on its bottom-right corner. Photos: MS-Celeb-1M dataset, hand-drawn caricatures (from top to bottom): ©Lucy Feng/deviantart, ©Tonio/toonpool.

Facial caricature is an art form of drawing faces in an exaggerated way to convey humor or sarcasm. In this paper, we propose the first Generative Adversarial Network (GAN) for unpaired photo-to-caricature translation, which we call "CariGANs". It explicitly models geometric exaggeration and appearance stylization using two components: *CariGeoGAN*, which only models the geometry-to-geometry transformation from face photos to caricatures, and *CariStyGAN*, which transfers the style appearance from caricatures to face photos without any geometry deformation. In this way, a difficult cross-domain translation problem is decoupled into two easier tasks. The perceptual study shows that caricatures generated by our *CariGANs* are closer to the hand-drawn ones, and at the same time better persevere the identity, compared to state-of-the-art methods. Moreover, our *CariGANs* allow users to control the shape exaggeration degree and change the color/texture style by tuning the parameters or giving an example caricature.

CCS Concepts: • **Computing methodologies → Image manipulation**; *Computational photography*; *Neural networks*;

Additional Key Words and Phrases: Caricature; Image translation; GAN

1 INTRODUCTION

A caricature can be defined as an art form of drawing persons (usually faces) in a simplified or exaggerated way through sketching, pencil strokes, or other artistic drawings. As a way to convey humor or sarcasm, caricatures are commonly used in entertainment, and as gifts or souvenirs, often drawn by street vendors. Artists have the amazing ability to capture distinct facial features of the subject from others, and then exaggerate those features.

There have been a few attempts to interactively synthesize facial caricature [Akleman 1997; Akleman et al. 2000; Chen et al. 2002; Gooch et al. 2004], but it requires professional skills to produce expressive results. A few automatic systems are proposed, which rely on hand-crafted rules [Brennan 2007; Koshimizu et al. 1999; Liang et al. 2002; Mo et al. 2004], often derived from the drawing procedure of artists. However, these approaches are restricted to a particular artistic style, *e.g.*, sketch or a certain cartoon, and predefined templates of exaggeration.

In recent years, deep learning, as the representative technique of learning from examples (especially from big data), has been successfully used in image-to-image translation [Hinton and Salakhutdinov 2006; Huang et al. 2018; Isola et al. 2017; Kim et al. 2017; Liu et al. 2017; Yi et al. 2017; Zhu et al. 2017b]. As is commonly known, most photo and caricature examples are unfortunately *unpaired* in the world. So the translation may be infeasible to be trained in a supervised way like Autoencoder [Hinton and Salakhutdinov 2006], Pix2Pix [Isola et al. 2017], and other paired image translation networks. Building such a dataset with thousands of image pairs (*i.e.*, a face photo and its associate caricature drawn by artists) would be too expensive and tedious.

On the other hand, there are two keys to generating a caricature: shape exaggeration and appearance stylization, as shown in Fig. 1

[*] Project page: https://cari-gan.github.io/
[†] This work was done when Kaidi Cao was an intern at Microsoft Research Asia.
[‡] indicates corresponding author.

Authors' addresses: Kaidi Cao, Department of Electronic Engineering, Tsinghua University; Jing Liao, City University of Hong Kong, Microsoft Research; Lu Yuan, Microsoft AI Perception and Mixed Reality.

(a)(b). Neural style transfer methods [Gatys et al. 2015; Johnson et al. 2016; Liao et al. 2017], which transfer the artistic style from a given reference to a photo through deep neural networks, are good at stylizing appearances, but do not exaggerate the geometry, as shown in Fig. 1 (c). There are a few works [Huang et al. 2018; Liu et al. 2017; Zhu et al. 2017a,b] proposed for unsupervised cross-domain image translation, which in principle will learn both geometric deformation and appearance translation simultaneously. However, the large gap of shape and appearance between photos and caricatures imposes a big challenge to these networks, and thus they generate unpleasant results, as shown in Fig. 1 (d).

In order to generate a reasonable result approaching caricature artists' productions, one has to ask "what is the desired quality of caricature generation?". Shape exaggeration is not a distortion, which is complete denial of truth [Redman 1984]. The exaggerated shape should maintain the relative geometric location of facial components, and only emphasize the subject's features, distinct from others. The final appearance should be faithful to visual styles of caricatures, and keep the *identity* with the input face, as addressed in other face generators [Brennan 2007; Liang et al. 2002; Mo et al. 2004]. Moreover, the generation must be diverse and controllable. Given one input face photo, it allows for the generation of variant types of caricatures, and even controls the results either by example caricature, or by user interactions (*e.g.*, tweaking exaggerated shape). It can be useful and complementary to existing interactive caricature systems.

In this paper, we propose the first Generative Adversarial Network (GAN) for unpaired photo-to-caricature translation, which we call "CariGANs". It explicitly models geometric exaggeration and appearance stylization using two components: *CariGeoGAN*, which only models the geometry-to-geometry transformation from face photos to caricatures, and *CariStyGAN*, which transfers the style appearance from caricatures to face photos without any geometry deformation. Two GANs are separately trained for each task, which makes the learning more robust. To build the relation between unpaired image pairs, both *CariGeoGAN* and *CariStyGAN* use cycle-consistency network structures, which are widely used in cross-domain or unsupervised image translation [Huang et al. 2018; Zhu et al. 2017b]. Finally, the exaggerated shape (obtained from *CariGeoGAN*) serves to exaggerate the stylized face (obtained from *CariStyGAN*) via image warping.

In *CariGeoGAN*, we use the PCA representation of facial landmarks instead of landmarks themselves as the input and output of GAN. This representation implicitly enforces the constraint of face shape prior in the network. Besides, we consider a new characteristic loss in *CariGeoGAN* to encourage exaggerations of distinct facial features only, and avoid arbitrary distortions. Our *CariGeoGAN* outputs the landmark positions instead of the image, so the exaggeration degree can be tweaked before the image warping. It makes results controllable and diverse in geometry.

As to the stylization, our *CariStyGAN* is designed for pixel-to-pixel style transfer without any geometric deformation. To exclude the interference of geometry in training *CariStyGAN*, we create an intermediate caricature dataset by warping all original caricatures to the shapes of photos via the reverse geometry mapping derived from *CariGeoGAN*. In this way, the geometry-to-geometry translation achieved by *CariGeoGAN* is successfully decoupled from the appearance-to-appearance translation achieved by *CariStyGAN*. In addition, our *CariStyGAN* allows multi-modal image translation, which traverses the caricature style space by varying the input noise. It also supports example-guided image translation, in which the style of the translation outputs are controlled by a user-provided example caricature. To further keep identity in appearance stylization, we add perceptual loss [Johnson et al. 2016] into *CariStyGAN*. It constrains the stylized result to preserve the content information of the input.

With our *CariGAN*, the photos of faces in the wild can be automatically translated to caricatures with geometric exaggeration and appearance stylization, as shown in Fig. 1 (f). We have extensively compared our method with state-of-the-art approaches. The perceptual study results show caricatures generated by our *CariGANs* are closer to the hand-drawn caricatures, and at the same time better persevere the identity, compared to the state-of-the-art. We further extend the approach to new applications, including generating video caricatures, and converting a caricature to a real face photo.

In summary, our key contributions are:

(1) We present the first deep neural network for unpaired photo-to-caricature translation. It achieves both geometric exaggeration and appearance stylization by explicitly modeling the translation of geometry and appearance with two separate GANs.
(2) We present *CariGeoGAN* for geometry exaggeration, which is the first attempt to use cycle-consisteny GAN for cross-domain translation in geometry. To constrain the shape exaggeration, we adopt two major novel extensions, like PCA representation of landmarks, and a characteristic loss.
(3) We present *CariStyGAN* for appearance stylization, which allows multi-modal image translation, while preserving the identity in the generated caricature by adding a perceptual loss.
(4) Our *CariGANs* allows user to control the exaggeration degree in geometric and appearance style by simply tuning the parameters or giving an example caricature.

2 RELATED WORK

Recent literature suggests two main directions to tackle the photo-to-caricature transfer task: traditional graphics-based methods and recent deep learning-based methods.

Graphics-based methods. In computer graphics, translating photo to caricature or cartoon is interesting, and has been studied for a long while. These techniques can be categorized into three groups.

The category develops deformation systems which allow users to manipulate photos interactively [Akleman 1997; Akleman et al. 2000; Chen et al. 2002; Gooch et al. 2004]. These kind of methods usually require expert knowledge and detailed involvement of experienced artists.

The second category defines hand-craft rules to automatically exaggerate difference from the mean (EDFM). Brennan [Brennan 2007] is the first to present the EDFM idea. Some following works [Koshimizu et al. 1999; Le et al. 2011; Liao and Li 2004; Liu et al. 2006; Mo et al.

2004; Tseng and Lien 2007] improve rules of EDFM to represent the distinctiveness of the facial features better. Besides 2D exaggeration, there is also some work utilizing tensor-based 3D model to exaggerate facial features [Yang et al. 2012]. However there is a central question regarding the effectiveness of EDFM: whether these handcrafted rules faithfully reflect the drawing styles of caricaturists.

The third category of methods directly learn rules from paired photo-caricature images, drawn by caricaturists. For examples, Liang et al.[Liang et al. 2002] propose learning prototypes by analyzing the correlation between the image caricature pairs using partial least-squares (PLS). Shet et al. [Shet et al. 2005] train a Cascade Correlation Neural Network (CCNN) network to capture the drawing style in relation to facial components. In practice, however, it is difficult to obtain a large paired training set. Learning from one-shot or a few exemplars makes it ineffective to cover the variances of existing caricatures.

Neural style transfer. Recently, inspired by the power of CNN, the pioneering work of Gatys et al. [Gatys et al. 2015] presents a general solution to transfer the style of a given artwork to any image automatically. Many follow-up works have been proposed to improve quality [Liao et al. 2017; Szirányi and Zerubia 1997], speed [Chen et al. 2017b; Johnson et al. 2016], or video extension [Chen et al. 2017a]. Notwithstanding their success in transferring photos or videos into many artistic styles like pencil, watercolor, oil painting and etc., they fail to generate caricatures with geometry exaggerations since these methods transfer textures and colors of a specific style while preserving the image content.

Image-to-image translation networks. There are a series of work based on the GAN proposed for a general image-to-image translation. Isola et al. [Isola et al. 2017] develop the pix2pix network trained with the supervision of images pairs and achieve reasonable results on many translation tasks such as photo-to-label, photo-to-sketch and photo-to-map. BicycleGAN [Zhu et al. 2017b] extends it to multi-modal translation. Some networks including CycleGAN [Zhu et al. 2017a], DualGAN [Yi et al. 2017], DiscoGAN [Kim et al. 2017], UNIT [Liu et al. 2017], DTN [Taigman et al. 2016] etc. have been proposed for unpaired one-to-one translation, while MNUIT [Huang et al. 2018] was proposed for unpaired many-to-many translation. These networks often succeed on the unpaired translation tasks which are restricted to color or texture changes only, *e.g.*, horse to zebra, summer to winter. For photo-to-caricature translation, they fail to model both geometric and appearance changes. By contrast, we explicitly model the two translations by two separated GANs: one for geometry-to-geometry mapping and another for appearance-to-appearance translation. Both GANs respectively adopt the cycle-consistent network structures (*e.g.*, CycleGAN [Zhu et al. 2017a], MNUIT [Huang et al. 2018]) since each type of translation still builds on unpaired training images.

3 METHOD

For caricature generation, previous methods, based on learning from examples, rely on paired photo-to-caricature images. Artists are required to paint corresponding caricatures for each photo. So it is infeasible to build such a paired image dataset for supervised learning due to high cost in money and time. In fact, there are a great number of caricature images found in the Internet, *e.g.*, Pinterest.com. How to learn the photo-to-caricature translation from unpaired photos and caricatures is our goal. Meanwhile, the generated caricature should preserve the identity of the face photo.

Let X and Y be the face photo domain and the caricature domain respectively, where no pairing exists between the two domains. For the photo domain X, we randomly sample $10,000$ face images from the CelebA database [Liu et al. 2015] $\{x_i\}_{i=1,\dots,N}, x_i \in X$ which covers diverse gender, races, ages, expressions, poses and etc. To obtain the caricature domain Y, we collect $8,451$ hand-drawn portrait caricatures from the Internet with different drawing styles (*e.g.*, cartoon, pencil-drawing) and various exaggerated facial features, $\{y_i\}_{i=1,\dots,M}, y_i \in Y$. We want to learn a mapping $\Phi : X \rightarrow Y$ that can transfer an input $x \in X$ to a sample $y = \Phi(x)$, $y \in Y$. This is a typical problem of cross-domain image translation, since photo domain and caricature domain may be obviously different in both geometry shape and texture appearance. We cannot directly learn the mapping form X to Y by other existing image-to-image translation networks. Instead, we decouple Φ into two mappings Φ_{geo} and Φ_{app} for geometry and appearance respectively.

Fig. 2 illustrates our two-stage framework, where two mappings Φ_{geo} and Φ_{app} are respectively learnt by two GANs. In the first stage, we use *CariGeoGAN* to learn geometry-to-geometry translation from photo to caricature. Geometric information is represented with facial landmarks. Let L_X, L_Y be the domains of face landmarks (from X) and caricature landmarks (from Y) respectively. In inference, the face landmarks l_x of the face photo x can be automatically estimated from an existing face landmark detector module. Then, *CariGeoGAN* learns the mapping $\Phi_{geo} : L_X \rightarrow L_Y$ to exaggerate the shape, generating the caricature landmark $l_y \in L_Y$. In the second stage, we use *CariStyGAN* to learn the appearance-to-appearance translation from photo to caricature while preserving its geometry. Here, we need to synthesize an intermediate result $y' \in Y'$, which is assumed to be as close as caricature domain Y in appearance and as similar as photo domain X in shape. The appearance mapping is denoted as $\Phi_{app} : X \rightarrow Y'$. Finally, we get the final output caricature $y \in Y$ by warping the intermediate stylization result y' with the guidance of exaggerated landmarks l_y. The warping is done by a differentiable spline interpolation module [Cole et al. 2017].

In next sections, we will describe the two GANs in detail.

3.1 Geometry Exaggeration

In this section, we present *CariGeoGAN* which learns geometric exaggeration of the distinctive facial features.

Training data. Face shape can be represented by 2D face landmarks either for real face photos X, or for caricatures Y. We manually label 63 facial landmarks for each image in both X and Y. For the annotation, we show overlaid landmarks in Fig. 3. To centralize all facial shapes, all images in both X and Y are aligned to the mean face via three landmarks (center of two eyes and center of the mouth) using affine transformation. In addition, all images are cropped to the face region, and resized to 256×256 resolution for normalizing the scale. Fig. 3 shows several transformed images with overlaid landmarks. Note that the dataset of real faces is also used to finetune

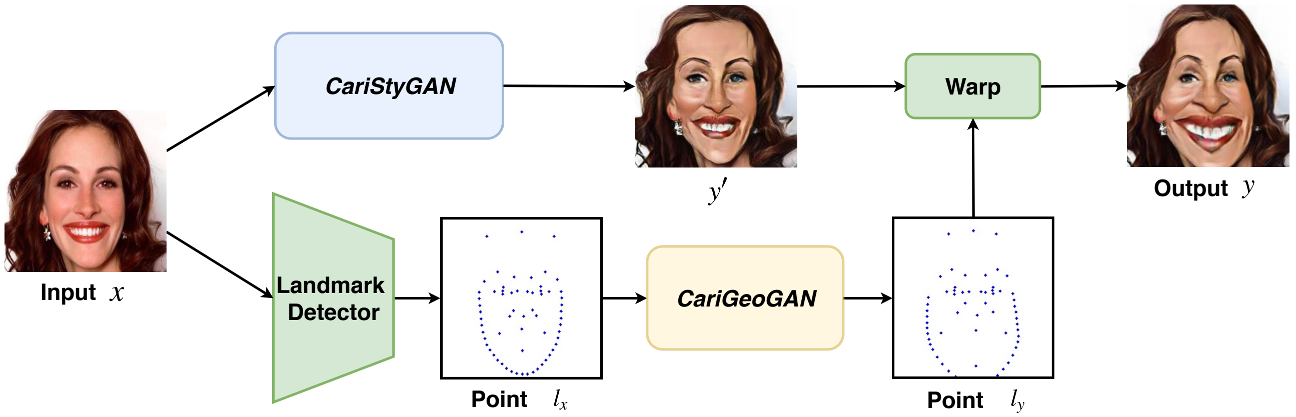

Fig. 2. Overall Pipeline of Proposed Method. Input image: CelebA dataset.

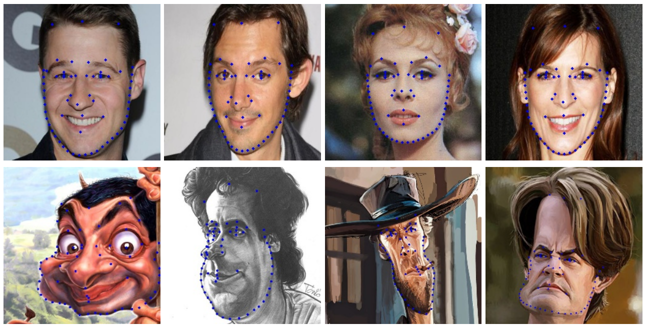

Fig. 3. Some samples from database of portrait photos (upper row) and caricatures (lower row). Photos: CelebA dataset, caricatures (from left to right): ©Tonio/toonpool, ©Tonio/toonpool, ©Alberto Russo/www.dessins.ch, ©Alberto Russo/www.dessins.ch.

a landmark detector ([Zhu et al. 2016]), which supports automatic landmark detection in the inference stage.

For our *CariGeoGAN*, to further reduce dimensions of its input and output, we further apply principal component analysis (PCA) on the landmarks of all samples in X and Y. We take the top 32 principal components to recover 99.03% of total variants. Then the 63 landmarks of each sample are represented by a vector of 32 PCA coefficients. This representation helps constrain the face structure during mapping learning. We will discuss its role in Section 4.1. Let L_X, L_Y be the PCA landmark domains of X and Y, respectively. Our *CariGeoGAN* learns the translation from L_X to L_Y instead.

CariGeoGAN. Since samples in L_X and L_Y are unpaired, the mapping function $\Phi_{geo} : L_X \rightarrow L_Y$ is highly under-constrained. CycleGAN [Zhu et al. 2017a] couples it with a reverse mapping $\Phi_{geo}^{-1} : L_Y \rightarrow L_X$. This idea has been successfully applied for unpaired image-to-image translation, *e.g.*, texture/color transfer. Our *CariGeoGAN* is inspired by the network architecture of CycleGAN. It contains two generators and two discriminators as shown in Fig. 4. The forward generator G_{L_Y} learns the mapping Φ_{geo} and synthesizes caricature shape $\widehat{l_y}$; while the backward generator G_{L_X} learns the reverse mapping Φ_{geo}^{-1} and synthesizes face shape $\widehat{l_x}$. The discriminator D_{L_X} (or D_{L_Y}) learn to distinguish real samples from L_X (or L_Y) and synthesized sample $\widehat{l_x}$ (or $\widehat{l_y}$).

The architecture of *CariGeoGAN* consists of two paths. One path models the mapping Φ_{geo}, shown in the top row of Fig. 4. Given a face shape $l_x \in L_x$, we can synthesize a caricature shape $\widehat{l_y} = G_{L_Y}(l_x)$. On one hand, $\widehat{l_y}$ is fed to the discriminator D_{L_Y}. On the other hand, $\widehat{l_y}$ can get back to approximate the input shape through the generator G_{L_X}. Similar operations are applied to the other path, which models the reverse mapping Φ_{geo}^{-1}, shown in the bottom row of Fig. 4. Note that G_{L_X} (or G_{L_Y}) shares weights in both pathes.

Our *CariGeoGAN* is different from CycleGAN since it takes PCA vector instead of image as input and output. To incorporate the PCA landmark representation with GAN, we replace all CONV-ReLu blocks with FC-ReLu blocks in both generators and discriminators.

Loss. We define three types of loss in the *CariGeoGAN*, which are shown in Fig. 4.

The first is the adversarial loss, which is widely used in GANs. Specifically, we adopt the adversarial loss of LSGAN [Mao et al. 2017] to encourage generating landmarks indistinguishable from the hand-drawn caricature landmarks sampled from domain L_Y:

$$\begin{aligned}\mathcal{L}_{\text{adv}}^{L_Y}(G_{L_Y}, D_{L_Y}) =& \mathbb{E}_{l_y \sim L_Y}[(D_{L_Y}(l_y) - 1)^2] \\ &+ \mathbb{E}_{l_x \sim L_X}[D_{L_Y}(G_{L_Y}(l_x)^2]. \qquad (1)\end{aligned}$$

Symmetrically, we also apply adversarial loss to encourage G_{L_Y} to generate portrait photo landmarks that cannot be distinguished by the adversary D_{L_X}. The loss $\mathcal{L}_{\text{adv}}^{L_X}(G_{L_X}, D_{L_X})$ is similarly defined as Eq. (1).

The second is the bidirectional cycle-consistency loss, which is also used in CycleGAN to constrain the cycle consistency between the forward mapping Φ_{geo} and the backward mapping Φ_{geo}^{-1}. The idea is that if we apply exaggeration to l_x with G_{L_Y}, we should get back to the input l_x exactly with G_{L_X}, *i.e.*, $G_{L_X}(G_{L_Y}(l_x)) \simeq l_x$. The consistency in the reverse direction $G_{L_Y}(G_{L_X}(l_y)) \simeq l_y$ is defined

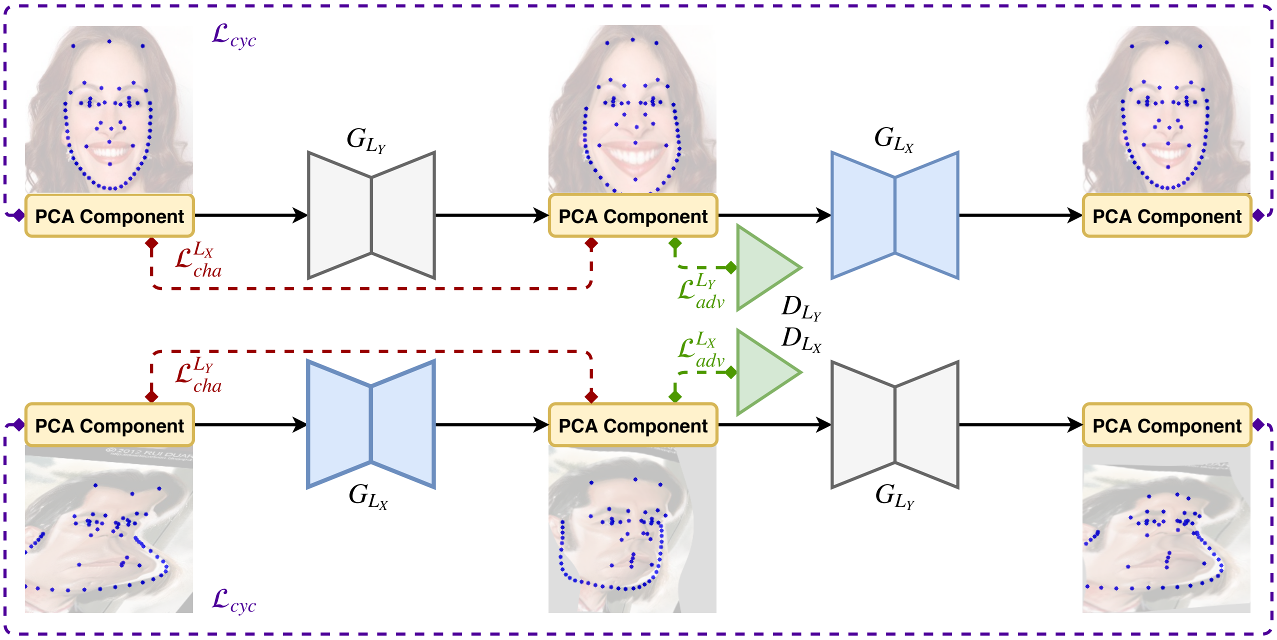

Fig. 4. Architecture of *CariGeoGAN*. It basically follows the network structure of CycleGAN with cycle Loss $\mathcal{L}_{cyc}$ and adversarial loss $\mathcal{L}_{gan}$. But our input and output are vectors instead of images, and we add a characteristic loss $\mathcal{L}_{cha}$ to exaggerate the subject's distinct features. Input images: CelebA dataset.

similarly. The loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G_{L_Y}, G_{L_X}) = &\mathbb{E}_{l_x\sim L_X}[||G_{L_X}(G_{L_Y}(l_x)) - l_x||_1] \\ &+\mathbb{E}_{l_y\sim L_Y}[||G_{L_Y}(G_{L_X}(l_y)) - l_y||_1]. \qquad (2)\end{aligned}$$

Cycle-consistency loss further helps constrain the mapping solution from the input to the output. However, it is still weak to guarantee that the predicted deformation can capture the distinct facial features and then exaggerate them. The third is a new characteristic loss, which penalizes the cosine differences between input landmark $l_x \in L_X$ and the predicted one $G_{L_Y}(l_x)$ after subtracting its corresponding means:

$$\mathcal{L}^{L_Y}_{\text{cha}}(G_{L_Y}) = \mathbb{E}_{l_x\sim L_X}[1-\cos(l_x - \overline{L_X}, G_{L_Y}(l_x) - \overline{L_Y})], \qquad (3)$$

where $\overline{L_X}$ (or $\overline{L_Y}$) denotes the averages of L_X (or L_Y). The characteristic loss in the reverse direction $\mathcal{L}^{L_X}_{\text{cha}}(G_{L_X})$ is defined similarly. The underlying idea is that the differences from a face to the mean face represent its most distinctive features and thus should be kept after exaggeration. For example, if a face has a larger nose compared to a normal face, this distinctiveness will be preserved or even exaggerated after converting to caricature.

Our objective function for optimizing *CariGeoGAN* is:

$$\mathcal{L}_{geo} = \mathcal{L}^{L_X}_{\text{adv}} + \mathcal{L}^{L_Y}_{\text{adv}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{cha}}(\mathcal{L}^{L_X}_{\text{cha}} + \mathcal{L}^{L_Y}_{\text{cha}}), \qquad (4)$$

where λ_{cyc} and λ_{cha} balance the multiple objectives.

Fig. 5 shows the roles of each loss, which is added to the objective function one by one. With adversarial only, the model will collapse and all face shapes in L_X map to a very similar caricature shape. With adding Cycle-consistency loss, the output varies with the input but the exaggeration direction is arbitrary. By adding characteristic loss, the exaggeration becomes meaningful. It captures the most distinct face features compared with the mean face, and then exaggerates the distinct facial features. Please note although the characteristic loss encourages the direction but itself is not enough to determine the exaggeration, since it cannot constrain the exaggeration amplitude and relationship between different facial features. For example, if we simply amplify differences from the mean by a factor of 2 or 3, it minimizes $\mathcal{L}_{\text{cha}}$ but leads to unsatisfactory results as shown in Fig. 6. In summary, our geometric exaggeration is

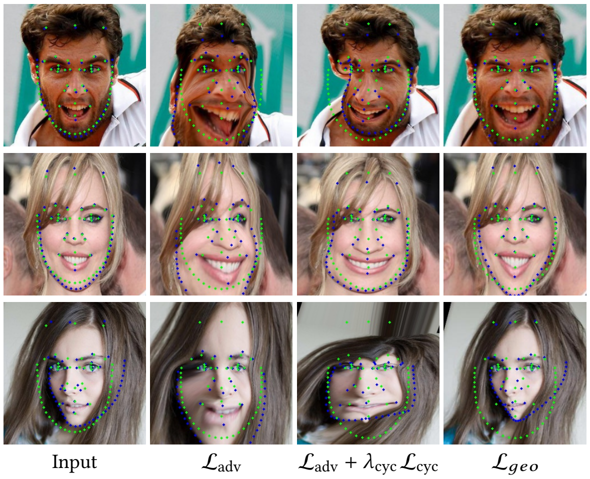
Input
$\mathcal{L}_{\text{adv}}$
$\mathcal{L}_{\text{adv}} + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}}$
$\mathcal{L}_{geo}$

Fig. 5. Comparing our *CariGeoGAN* with different losses. The green points represent the landmarks of the mean face, while the blue ones represent the landmarks of the input or exaggerated face. Input images: CelebA dataset.

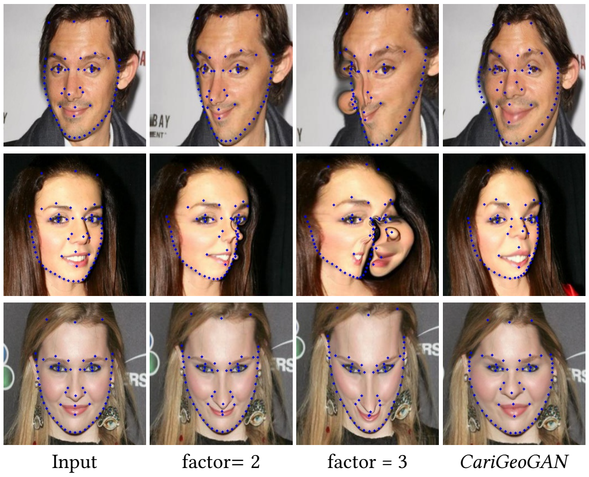

Fig. 6. Comparing our *CariGeoGAN* to simple exaggeration of the PCA coefficients from the mean by a factor of 2 or 3. Input images: CelebA dataset.

learned from data by *CariGeoGAN* to balance all the four types of losses. That is also the major difference compared to hand-crafted rules used in previous EDFM methods.

Training details. We use the same training strategy to CycleGAN [Zhu et al. 2017a]. For all the experiments, we set $\lambda_{cyc} = 10$ and $\lambda_{cha} = 1$ empirically and use the Adam solver [Kingma and Ba 2014] with a batch size of 1. All networks are trained from scratch with an initial learning rate of 0.0002.

3.2 Appearance Stylization

In this section, we present *CariStyGAN*, which learns to apply appearance styles of caricatures to portrait photos without changes in geometry.

Training data. In order to learn a pure appearance stylization without any geometric deformation, we need to synthesize an intermediate domain Y', which has the same geometry distribution to X and has the same appearance distribution as Y. We synthesize each image $\{y_i'\}_{i=1,\dots,M}, y_i' \in Y'$ by warping every caricature image $y_i \in Y$ with the landmarks translated by our *CariGeoGAN*: $G_{L_X}(y_i)$. Our *CariStyGAN* learns the translation from X to Y' instead.

CariStyGAN. Since the mapping from X to Y' is a typical image-to-image translation task without geometric deformation, some general image-to-image translation network, *e.g.*, CycleGAN [Zhu et al. 2017a], MNUIT [Huang et al. 2018], can be applied. As shown in the 2nd, 3rd and 4th columns of Fig. 7, the result obtained by CycleGAN is acceptable in preserving structure but lacks diversity; while MUNIT generates multiple results with various styles but these results fail to preserve face structure. We find the reason is that the feature-level cycle-consistency used in MUNIT is less constrained than the image-level cycle-consistency used in CycleGAN. This is verified by replacing the feature-level cycle-consistency in MUNIT with the image-level one, results of which are shown in 5th and 6th columns of Fig. 7.

Our *CariStyGan* combines merits of the two networks, *i.e.*, allowing diversity and preserving structure. We inherit the image-level cycle-consistency constraint from CycleGAN to keep the face structure, while we are inspired from MNUIT to explicitly disentangle image representation into a content code that is domain-invariant, and a style code that captures domain-specific properties. By recombining a content code with various style codes sampled from the style space of the target domain, we may get multiple translated results.

Different from a traditional auto-encoder structure, we design an auto-encoder consisting of two encoders and one decoder for images from each domain $I (I = X, Y')$. The content encoder E_I^c and the style encoder E_I^s, factorize the input image $z_I \in I$ into a content code c_I and a style code s_I respectively, *i.e.*, $(c_I, s_I) = (E_I^c(z_I), E_I^s(z_I))$. The decoder R_I reconstructs the input image from its content and style code, $z_I = R_I(c_I, s_I)$. The domain of style code S_I is assumed to be Gaussian distribution $\mathcal{N}(0, 1)$.

Fig. 8 shows our network architecture in the forward cycle. Image-to-image translation is performed by swapping the encoder-decoder pairs of the two domains. For example, given a portrait photo $x \in X$, we first extract its content code $c_x = E_X^c(x)$ and randomly sample a style code $s_{y'}$ from $S_{Y'}$. Then, we use the decoder $R_{Y'}$ instead of its original decoder R_X to produce the output image y', in caricature domain Y', denoted as $y' = R_{Y'}(c_x, s_{y'})$. y' is also constrained by the discriminator $D_{Y'}$.

By contrast to MUNIT [Huang et al. 2018], where the cycle-consistency is enforced in the two code domains, we enforce cycle-consistency in the image domain. It means that the recovered image $\widehat{x} = R_X(E_{Y'}^c(y'))$ should be close to the original input x.

With this architecture, the forward mapping $\Phi_{app} : X \longrightarrow Y'$ is achieved by $E_X + R_{Y'}$, while the back mapping $\Phi_{app}^{-1} : Y' \longrightarrow X$ is achieved by $E_{Y'} + R_X$. By sampling different style codes, the mappings become multi-modal.

Loss. The *CariStyGAN* comprises four types of loss, which are shown in Fig. 8.

The first is adversarial loss $\mathcal{L}_{\text{adv}}^{Y'}(E_X, R_{Y'}, D_{Y'})$, which makes the translated result $R_{Y'}(E_X^c(x), s_{y'})$ identical to the real sample in Y', where $D_{Y'}$ is a discriminator to distinguish the generated samples from the real ones in Y'. Another adversarial loss $\mathcal{L}_{\text{adv}}^{X}(E_{Y'}, R_X, D_X)$ is similarly defined for the reverse mapping $Y' \longrightarrow X$, where D_X is discriminator for X.

The second is reconstruction loss which penalizes the $L1$ differences between the input image and the result, reconstructed from its style code and content code, *i.e.*,

$$\mathcal{L}_{\text{rec}}^{I}(E_I^c, E_I^s, R_I) = \mathbb{E}_{z \sim I}[||R_I(E_I^c(z), E_I^s(z)) - z||_1] \tag{5}$$

The third is cycle-consistency loss, which enforces the image to get back after forward and backward mappings. Specifically, given $x \in X$, we get the result $R_{Y'}(E_X^c(x), s_{y'}), s_{y'} \in S_{Y'}$, which translates from X to Y'. The result is then fed into the encoder $E_{Y'}^c$ to get its content code. After combining the content code with a random code s_x sampled form S_X, we use the decoder R_X to get the final result.

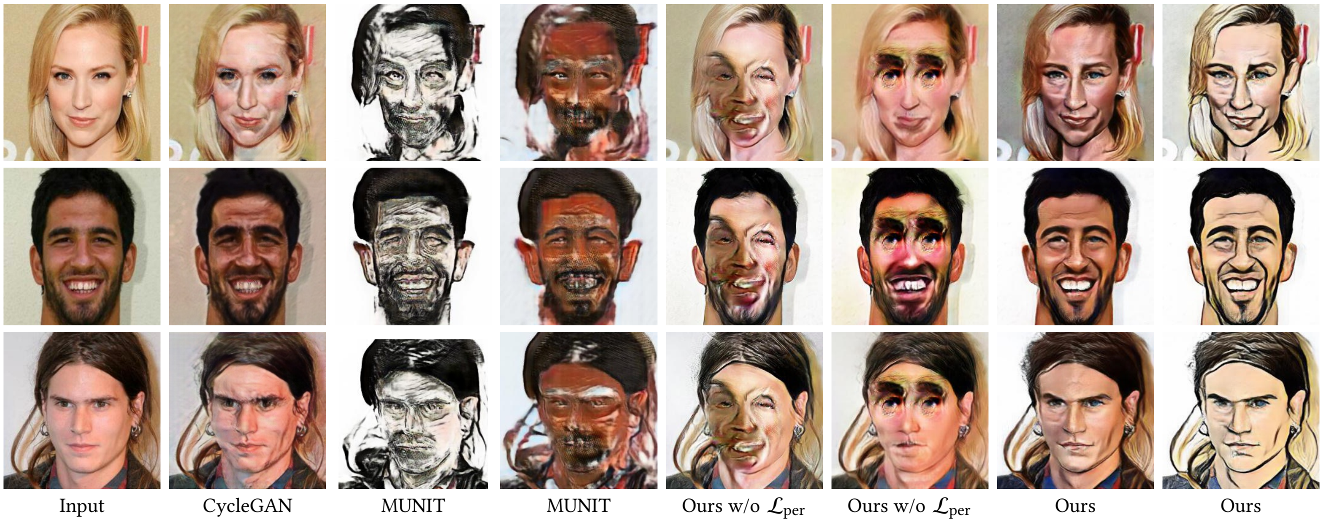
Input
CycleGAN
MUNIT
MUNIT
Ours w/o $\mathcal{L}_{\text{per}}$
Ours w/o $\mathcal{L}_{\text{per}}$
Ours
Ours

Fig. 7. Comparing our *CariStyGAN* with CycleGAN[Zhu et al. 2017b] and MUNIT[Huang et al. 2018]. All networks are trained with the same datasets to learn appearance style mapping $X \Rightarrow Y'$. CycleGAN generates a single result (2nd column). MUNIT is capable to generate diverse results but fails to preserve face structure (3rd and 4th second columns). Our *CariStyGAN* generates better diverse results by combining both CycleGAN and MUNIT (5th and 6th columns), and preserves identity by introducing a new perceptual loss (7th and 8th columns). Input images: CelebA dataset.

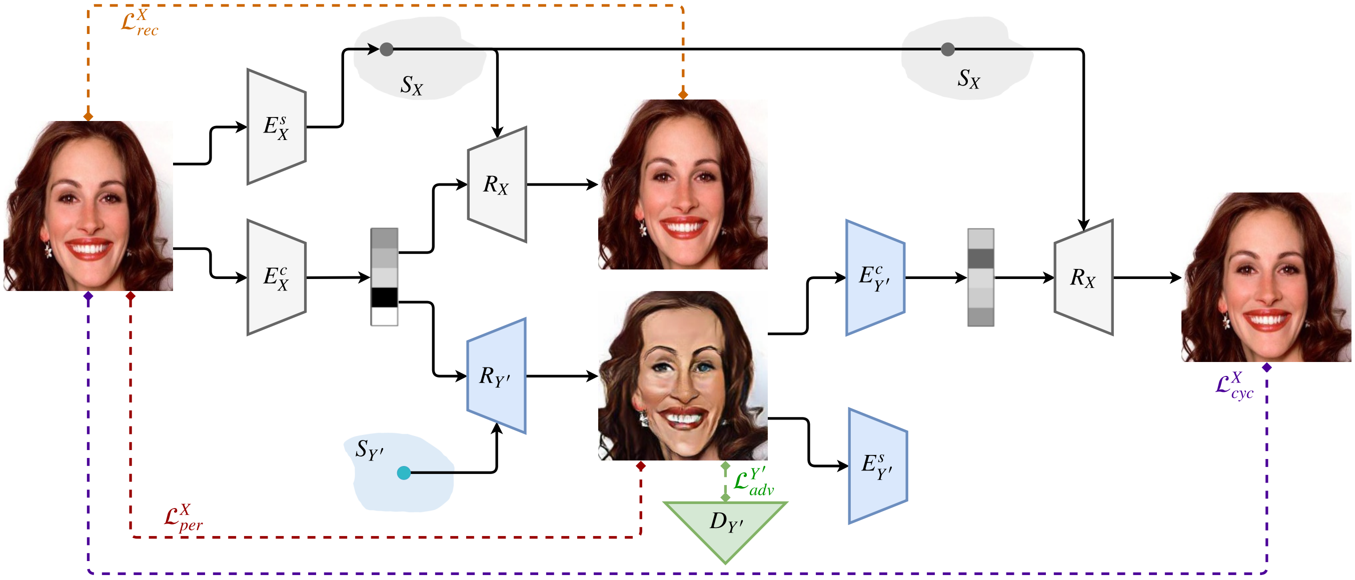

Fig. 8. Architecture of our *CariStyGan*. For simplicity, here we only show the network architecture for the translation $X \rightarrow Y'$. And the network architecture for the reverse translation $Y' \rightarrow X$ is symmetric. Input image: CelebA dataset.

It should be the same to the original input x:

$$\mathcal{L}^X_{\text{cyc}}(E^c_X, R_{Y'}, E^c_{Y'}, R_X) =$$
$$\mathbb{E}_{x \sim X', s_x \sim S_X, s_{y'} \sim S_{Y'}}[||R_X(E^c_{Y'}(R_{Y'}(E^c_X(x), s_{y'})), s_x) - x||_1], \quad (6)$$

The cycle loss for the reverse mapping $Y' \longrightarrow X$ is defined symmetrically and denoted as $\mathcal{L}^{Y'}_{\text{cyc}}(E^c_{Y'}, R_X, E^c_X, R_{Y'})$.

The aforementioned three losses are inherited form MUNIT and cycleGAN. With the three only, the decoupling of style code and content code is implicitly learned and vaguely known. In the photo-caricature task, we find that style code is not completely decoupled with the content code, which may cause the failure of preserving the identity after translation, as shown in the 5&6 $- th$ rows of Fig. 7. To address this issue, we add a new perceptual loss [Johnson et al. 2016], which can explicitly constrain the translated result to have

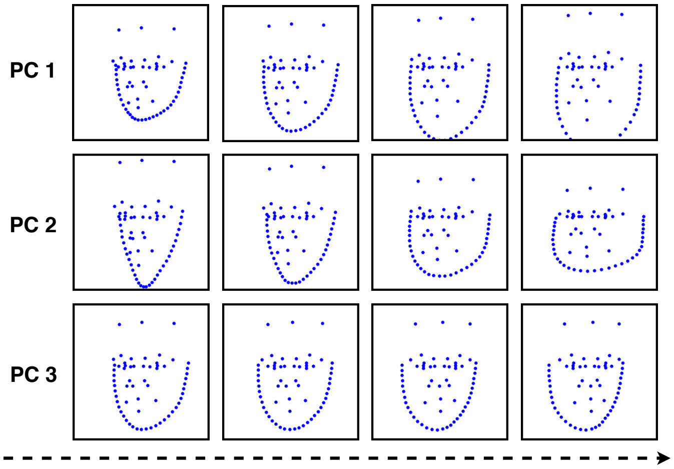

Fig. 9. Visualization of top three principal components of landmarks

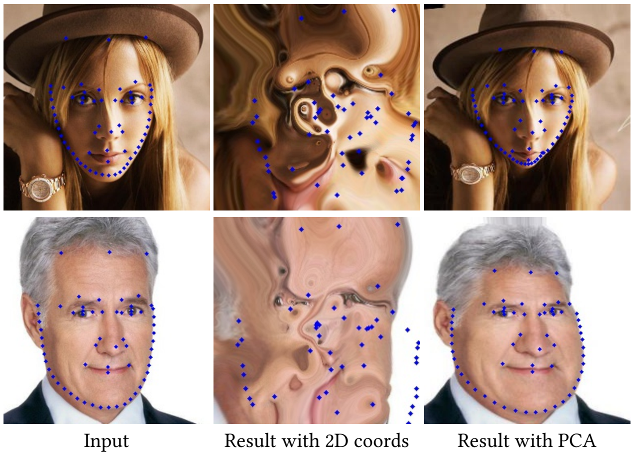

Fig. 10. Comparison between using PCA representation and using 2D coordinate representation in *CariGeoGAN*. Input images: CelebA dataset.

the same content information to the input:

$$\mathcal{L}^{X}_{\text{per}}(E^{c}_{X}, R_{Y'}) = \mathbb{E}_{x\sim X, s_{y'}\sim Y'}[||\text{VGG19}_{5\_3}(R_{Y'}(E^{c}_{X}(x), s_{y'})) - \text{VGG19}_{5\_3}(x)||_2], \tag{7}$$

where VGG19_{5_3} denotes to the *relu*5_3 feature map in VGG19 [Simonyan and Zisserman 2014], pre-trained on image recognition task. $\mathcal{L}^{Y'}_{\text{per}}(E^{c}_{Y'}, R_X)$ is defined symmetrically.

In summary, we jointly train the encoders, decoders and discriminators in our *CariStyGAN* by optimizing the final loss function:

$$\mathcal{L}_{app} = \mathcal{L}^{X}_{\text{adv}} + \mathcal{L}^{Y'}_{\text{adv}} + \lambda_{\text{rec}}(\mathcal{L}^{X}_{\text{rec}} + \mathcal{L}^{Y'}_{\text{rec}}) + \lambda_{\text{cyc}}(\mathcal{L}^{X}_{\text{cyc}} + \mathcal{L}^{Y'}_{\text{cyc}}) + \lambda_{\text{per}}(\mathcal{L}^{X}_{\text{per}} + \mathcal{L}^{Y'}_{\text{per}}). \tag{8}$$

λ_{rec}, λ_{cyc} and λ_{per} balance the multiple objectives.

Training details. We use the same structure as MUNIT in our encoders, decoders, and discriminators, and follow its training strategy. For all the experiments, we set $\lambda_{rec} = 1$, $\lambda_{per} = 0.2$ and $\lambda_{cyc} = 1$ empirically and use the Adam solver [Kingma and Ba 2014] with a batch size of 1. All networks are trained from scratch with an initial learning rate of 0.0001.

4 DISCUSSION

In this section, we analyze two key components in our *CariGANs*, *i.e.*, PCA representation in *CariGeoGAN* and intermediate domain in *CariStyGAN*.

4.1 Why PCA representation is essential to *CariGeoGAN*?

Although the dimensionality of landmarks with 2D coordinates (63 × 2) is low compared to the image representation, in our *CariGeoGAN* we still use the PCA to reduce dimensions of the landmark representation. That is because geometry translation is sometimes harder than image translation. First, landmarks are fed into fully-connected layers instead of convolutional layers, so they lose the locally spatial constraint during learning. Second, the result is more sensitive to small errors in landmarks than in image pixels, since these errors may cause serious geometric artifacts, like foldover or zigzag contours. If we use the raw 2D coordinates of landmarks to train *CariGeoGAN*, the face structure is hardly preserved as shown in Fig. 10. On the contrary, the PCA helps constrain the face structure in the output. It constructs an embedding space of face shapes, where each principle component represents a direction of variants, like pose, shape, size, as shown by the visualization of top three principle components in Fig. 9. Any sample in the embedding space will maintain the basic face structure. We compared two kinds of representations used in *CariGeoGAN* (PCA *vs.* 2D coordinate), and show visual results in Fig. 10.

4.2 Why is intermediate domain crucial to *CariStyGAN*?

The construction of the intermediate domain Y' is an important factor for the success of our *CariStyGAN*, since it bridges the geometric differences between photo domain X and caricature domain Y, and thus allows the GAN focusing on appearance translation only. To understand the role of Y' well, we train *CariStyGAN* to learn the mapping from X to Y directly. In this setting, some textures may mess up the face structure, as shown in Fig. 11. One possible reason is that the network attempts to learn two mixed mappings (geometry and appearance) together. The task is so difficult to be learned.

4.3 How many styles have been learned by *CariStyGAN*?

To answer this question, we randomly select 500 samples from the testing photo dataset and 500 samples from the of training hand-drawn caricatures dataset. For each photo sample we generate a caricature with our *CariGANs* and a random style code. Then we follow [Gatys et al. 2015] to represent the appearance style of each sample with the Gram Matrix of its VGG19 feature maps. The embedding of appearance styles on 2D is visualized via the T-SNE method. It is clearly shown in Fig. 12, there is little interaction between photos and hand-drawn caricatures, however, through translation our

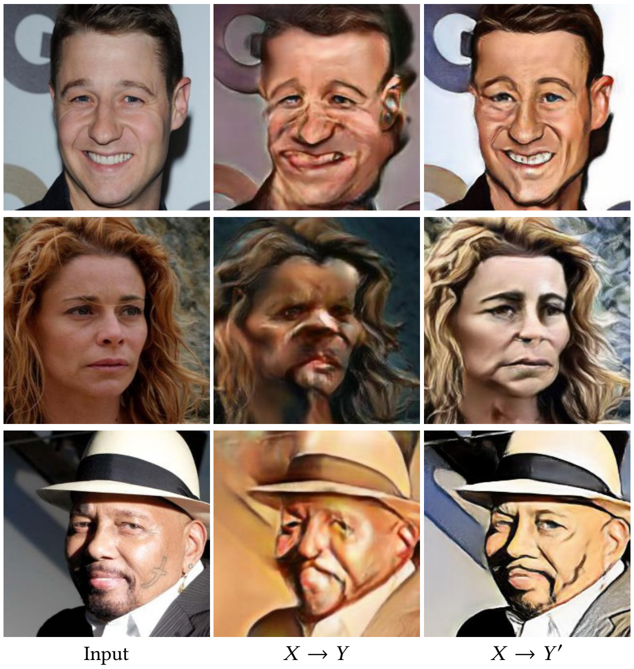

Fig. 11. The importance of intermediate domain to train *CariStyGAN*. Input images: CelebA dataset.

generated results almost share the same embedding space as caricatures. That means most styles in the training caricature dataset have been learned by our *CariStyGAN*.

5 COMPARISON AND RESULT

In this section, we first show the performance of our system, and demonstrate the result controllability in two aspects. Then we qualitatively compare our results to previous techniques, including both traditional graphics-based methods and recent deep-learning based methods. Finally, we provide the perceptual study results.

5.1 Performance

Our core algorithm is developed in PyTorch [Paszke et al. 2017]. All of our experiments are conducted on a PC with an Intel E5 2.6GHz CPU and an NVIDIA K40 GPU. The total runtime for a 256 × 256 image is approximately 0.14 sec., including 0.10 sec. for appearance stylization, 0.02 sec. for geometric exaggeration and 0.02 sec. for image warping.

5.2 Results with control

Our *CariGANs* support two aspects of control. First, our system allows users to tweak the geometric exaggeration extent with a parameter $\alpha \in [0.0, 2.0]$. Let l_x to be the original landmarks of the input photo x, and l_y to be the exaggerated landmarks predicted by *CariGeoGAN*. Results with different exaggeration degrees can be obtained by interpolation and extrapolation between them: $l_x + \alpha(l_y - l_x)$. Figure 13 shows such examples.

Except for geometry, our system allows user control on appearance style as well. On one hand, our *CariStyGAN* is a multi-modal image translation network, which can convert a given photo into different caricatures, which are obtained by combining photos with different style codes, sampled from a Gaussian distribution. On the other hand, in the *CariStyGAN*, a reference caricature can be encoded into style code using $E^s_{Y'}$. After combining with the code, we can get the result with a similar style as the reference. So the user can control the appearance style of the output by either tuning the value of style code or giving a reference. In Figure 14, we show diverse results with 4 random style codes and 2 style codes from references.

5.3 Comparison to graphics-based methods

We compare our *CariGANs* with four representative graphics-based methods for caricature generation, including Gooch et al. [2004], Chen et al. [2002], Liang et al. [2002] and Mo et al. [2004]. We test on the cases from their papers and show the visual comparison in Fig. 15. It can be seen that these compared methods focus less on the appearance stylization and only model some simple styles like sketch or cartoon, while ours can reproduce much richer styles by learning from thousands hand-drawn caricatures. As to the geometric exaggeration, Gooch et al. [2004] and Chen et al. [2002] require manual specification. Liang et al. [2002] needs to learn the deformation form a pair of examples which is not easy to get in practice. Mo et al. [2004] is automatic by exaggerating differences from the mean, but the hand-crafted rules are difficult to describe all geometric variations in the caricature domain. In contrast, our learning-based approach is more scalable.

5.4 Comparison to deep-learning-based methods

We visually compare our CariGAN with existing deep-learning based methods in Fig. 16. Here, all methods are based on author provided implementations with the default settings, except for [Selim et al. 2016] which has no code released and we implement ourselves. First we compare with style transfer techniques which migrate the style from a given reference. We consider two general style transfer methods (Gatys et al. [2015] and Liao et al. [2017]), and two methods tailed for faces ([Selim et al. 2016] and [Fišer et al. 2017]). All the references used in these methods are randomly selected from our hand-drawn caricature dataset. As we can see, they can transfer the style appearance from the caricature to the input photo, but cannot transfer geometric exaggerations.

Our *CariGANs* is compared with three general image-to-image translation networks, including two representative works (CycleGAN [Zhu et al. 2017b] and UNIT [Liu et al. 2017]) for single-modal unpaired image translation, and the sole network for multi-modal unpaired image translation (MUNIT [Huang et al. 2018]). We train these networks using the same dataset as ours. Since their networks should learn both two mappings of geometry and appearance jointly, this poses a challenge beyond their capabilities. UNIT and MUNIT fail to preserve the face structure. CycleGAN keeps the face structure but few artistic style and exaggeration learned. Thanks to the two GANs framework, our network better simulates hand-drawn caricatures in both geometry and appearance, while keeping the identity of the input.

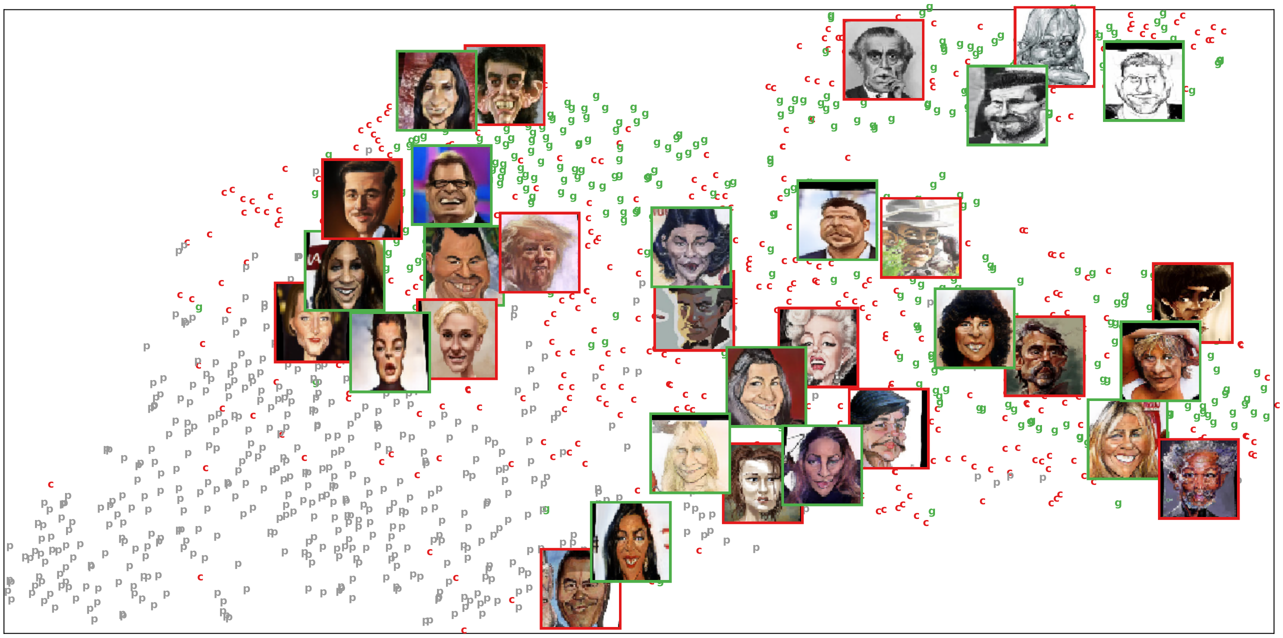

Fig. 12. T-SNE visualization of the style embedding. Gray points represent photos, red points represent hand-drawn caricatures, and green points represent generated results. The corresponding image of some point is shown with the same color border.

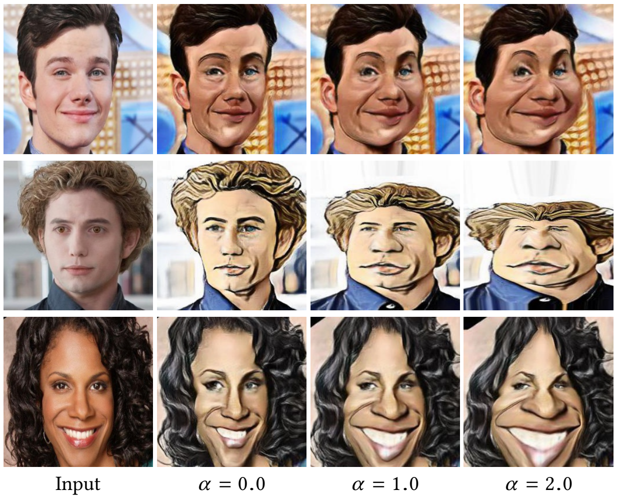

Fig. 13. Results with geometric control. Inputs are from CelebA dataset excluding the 10K images used in training.

5.5 Perceptual study

We conduct two perceptual studies to evaluate our *CariGAN* in terms of recognizability and style faithfulness. We compare caricatures drawn by artists with the results achieved by the following techniques: two neural style transfer methods (Gatys et al. [2015],Liao et al. [2017]), two image-to-image translation networks (CycleGAN [Zhu et al. 2017b], MUNIT [Huang et al. 2018]), and our *CariGAN*. The compared methods are trained and tested in the same way as described in Section 5.4.

We construct a new testing dataset with identity information for the two user studies. We randomly select 500 samples from our hand-drawn caricature dataset. For each caricature sample, we collect 20 different portrait photos of the same person from the Internet. All examples in the two studies are randomly sampled from this dataset, which are included in our supplemental material.

The first study assesses how well the identity is preserved in each technique. The study starts from showing 8 example caricature-photo pairs, to let the participant be familiar with the way of exaggeration and stylization in caricatures. Then 60 questions (10 for each method) follow. In each question, we present a caricature generated by our method or the method we compare it with, and ask the participant to select a portrait photo with the same identity as the caricature from 5 choices. Among the choices, one is the correct subject, while the other four items are photos of other subjects with similar attributes (*e.g.*, sexual, age, glasses) to the correct subject. The attributes are automatically predicted by Azure Cognitive Service. Participants are given unlimited time to answer. We collect 28 responses for each question, and calculate the recognition rate for each method, shown in Fig. 17 (a).

As we can see, the results of MUNIT, Neural Style, and Deep Analogy pose more difficulty in recognizing as the correct subject, since the visual artifacts in their results mess up the facial features. We show examples in Fig. 16. CycleGAN is good at preserving the identity because it is more conservative and produces photo-like results. Surprisingly, hand-drawn caricatures have the highest recognition rate, even better than the photo-like CycleGAN. We guess this is because professional artists are good at exaggerating the most distinct facial features which helps the recognition. Our

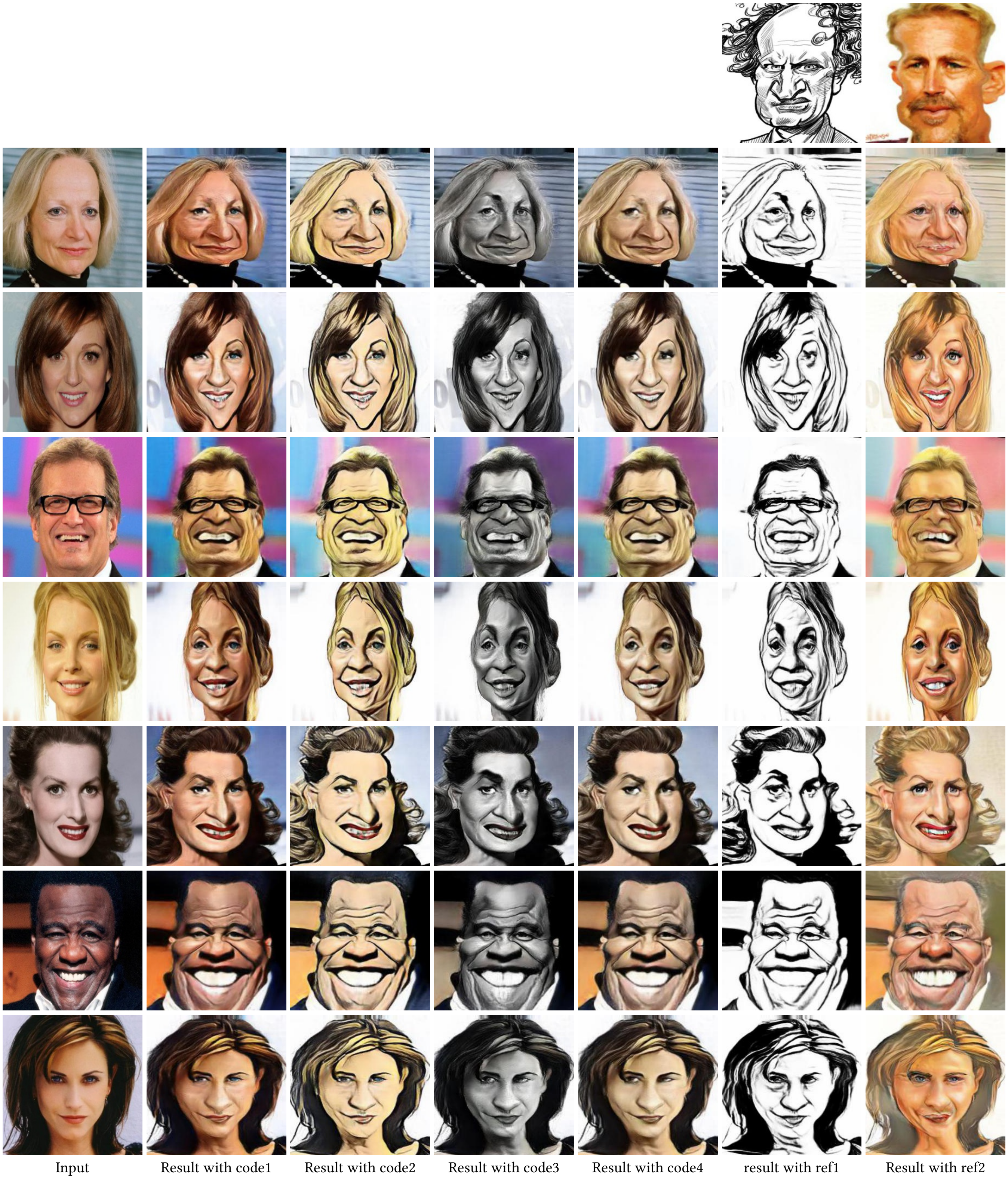

Fig. 14. Our system allows user control on appearance style. Results are generated with a random style code (first four) or a given reference (last two). Top row shows the two reference images. From left to right: ©Tom Richmond/tomrichmond.com, ©wooden-horse/deviantart. Inputs are from CelebA dataset excluding the 10K images used in training.

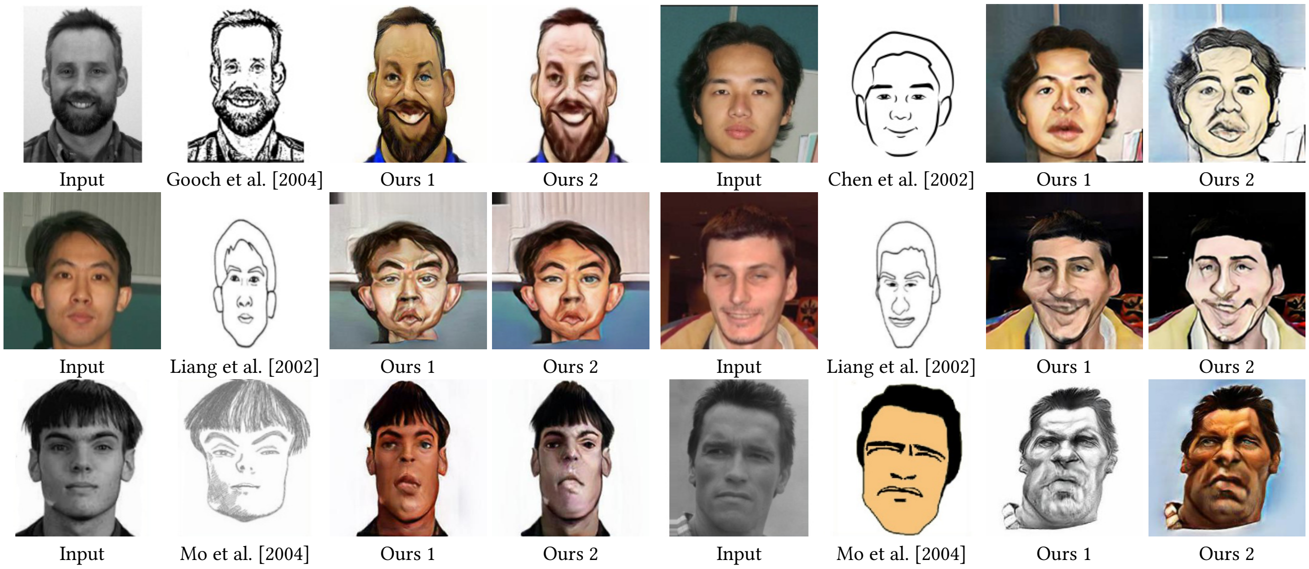

Fig. 15. Comparison with graphics-based caricature generation techniques, including two interaction-based methods (Gooch et al. [2004], Chen et al. [2002]), one paired-example-based method (Liang et al. [2002]) and one rule-based method (Mo et al. [2004]). Inputs are from their papers.

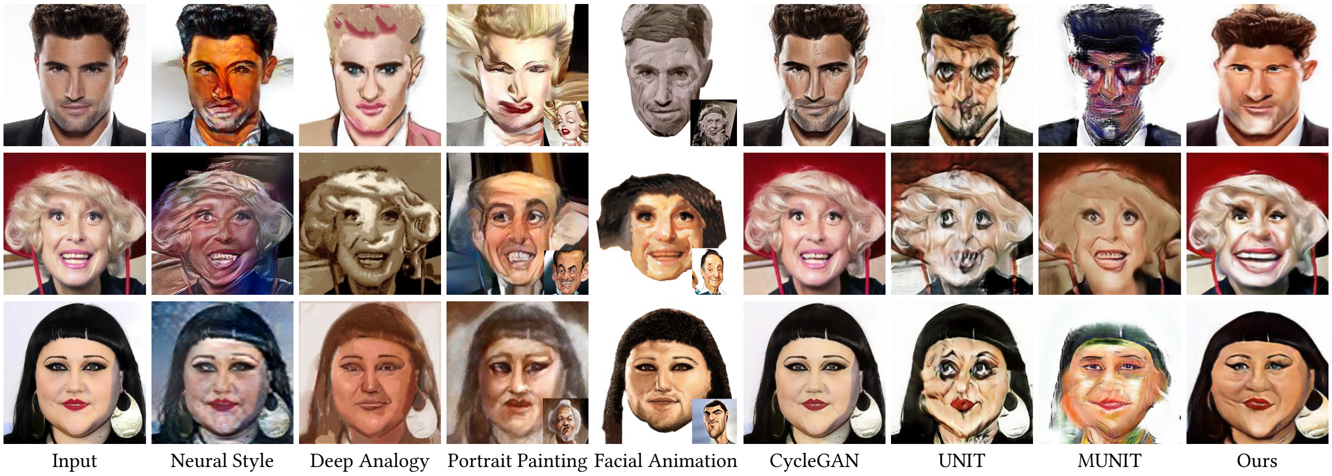

Fig. 16. Comparison with deep-learning-based methods, including two general image style transfer methods (neural style [Gatys et al. 2015] and Deep Analogy [Liao et al. 2017]), two face-specific style transfer methods (Portrait Painting [Selim et al. 2016] and Facial Animation [Fišer et al. 2017]), two single-modal image translation networks (CycleGAN[Zhu et al. 2017b], UNIT [Liu et al. 2017]) and one multi-modal image translation network (MUNIT[Huang et al. 2018]). Inputs are from CelebA dataset excluding the 10K images used in training. Input images: CelebA dataset.

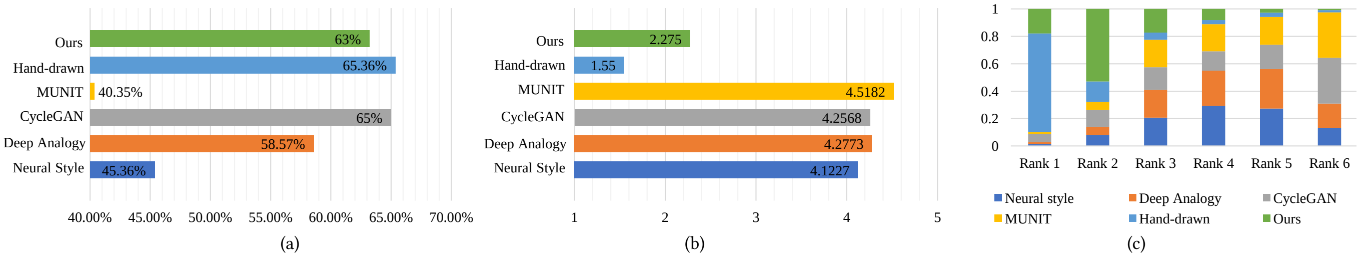

Fig. 17. User study results. (a) Percentages of correct face recognition in task 1. (b) Average rank of each method in task 2. (c) Percentage of each method that has been selected in each rank.

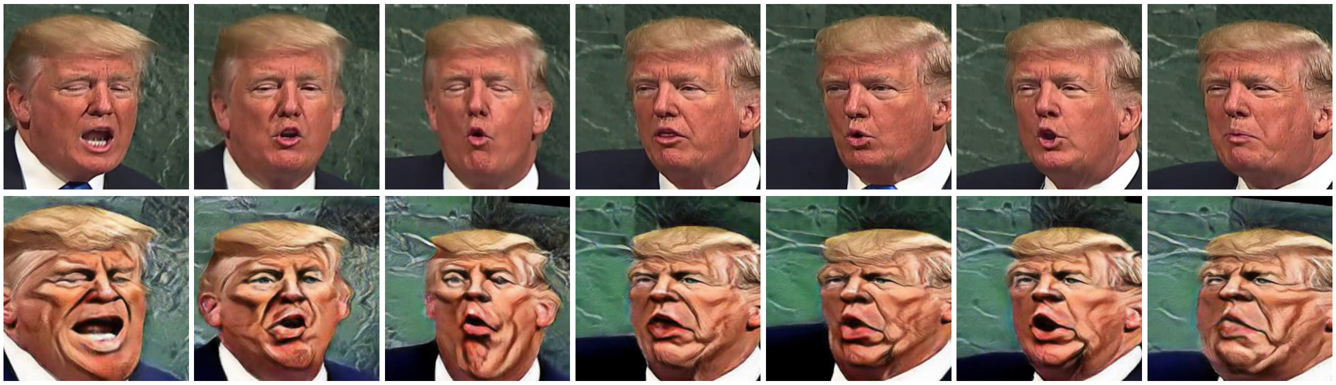

Fig. 18. A video caricature example. The upper row shows the input video frames and the bottom row shows the generated caricature results. Trump video courtesy of the White House (public domain).

recognition rate is also high, and very close to that of hand-drawn caricatures and photo-like results produced by CycleGAN.

The second study assesses how close the generated caricatures are to the hand-drawn ones in visual styles. The study begins with the showcase of 8 caricatures drawn by artists, which lets the participant know what the desired caricature styles is. Later, we present one hand-drawn caricature, and five results generated by ours and compared methods, to participants in a random order, at every question. These 6 caricatures depict the same person. We ask participants to rank them from “the most similar to given caricature samples" to “the least similar to caricature". We use 20 different questions and collect 22 responses for each question.

As shown in Fig. 17 (b), hand-drawn caricatures and ours rank as the top two. Our average rank is 2.275 compared to their rank 1.55. Other four methods have comparable average ranks but far behind ours. We further plot the percentages of each method that has been selected in each rank (Fig. 17 (c)). Note that ours is ranked better than the hand-drawn one 22.95% of the times, which means our results sometime can fool users into thinking it is the real hand-drawn caricature. Although it is still far from an ideal fooling rate (*i.e.*, 50%), our work has made a big step approaching caricatures drawn by artists, compared to other methods.

6 EXTENSIONS

We extend of CariGANs to two interesting applications.

6.1 Video caricature

We directly apply our CariGANs to the video frame by frame. Since our CariGANs exaggerate and stylize the face according to facial features. The results are overall stable in different frames, as shown in Fig. 18. Some small flickering can be resolved by adding temporal constraint in our networks, which is left for future work. The video demo can be found in our supplemental material.

6.2 Caricature-to-photo translation

Since both *CariGeoGAN* and *CariGeoStyGAN* are trained to learn the forward and backward mapping symmetrically, we can reverse the pipeline (Fig. 2) to convert an input caricature into its corresponding photo. Some examples are shown in Fig. 19. We believe it might be useful for face recognition in caricatures.

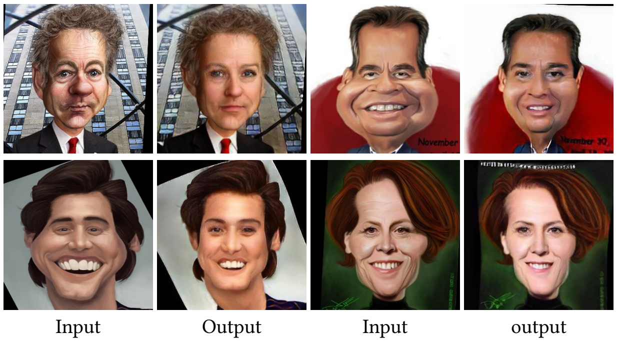

Fig. 19. Converting caricatures to photos. Inputs (from left to right, top to bottom): ©DonkeyHotey/Flickr, ©Rockey Sawyer/deviantart, ©Guillermo Ramírez/deviantart, ©Michael Dante/wittygraphy.

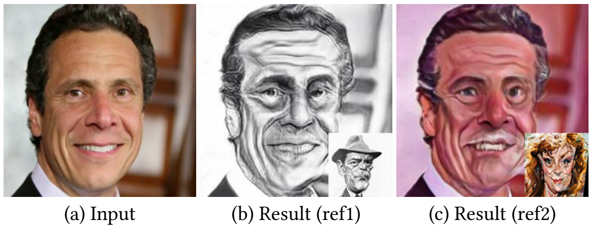

Fig. 20. Our result is faithful to the reference style which is common in the caricature dataset (b), but is less faithful to some uncommon style (c). Input image: CelebA dataset.

7 CONCLUSIONS

We have presented the first deep learning approach for unpaired photo-to-caricature translation. Our approach reproduces the art of caricature by learning both geometric exaggeration and appearance stylization respectively with two GANs. Our method advances the existing methods a bit in terms of visual quality and preserving identity. It better simulates the hand-drawn caricatures to some extent. Moreover, our approach supports flexible controls for user to change results in both shape exaggeration and appearance style.

Our approach still suffers from some limitations. First, Our geometric exaggeration is more obviously observed in the face shape than other facial features and some small geometric exaggerations on ears, hairs, wrinkles and etc., cannot be covered. That is because there are 33 out of total 63 landmarks lying on the face contour. Variants of these landmarks dominate the PCA representation. This limitation can be solved by adding more landmarks. Second, it is better to make our *CariGeoGAN* to be multi-modal as well as our *CariStyGAN*, but we fail to disentangle content and style in geometry since their definitions are still unclear. As to the appearance stylization, our results are faithful to the reference style which are common in the caricature dataset (*e.g.*, sketch, cartoon) but are less faithful to some uncommon styles (*e.g.*, oil painting), as shown in Figure 20. That is because the our *CariStyGAN* cannot learn the correct style decoupling with limited data. Finally, our *CariStyGAN* is trained and tested with low-res (256 × 256) images, we consider applying the progressive growing idea from [Karras et al. 2017] in our *CariStyGAN* to gradually add details for high-res images (*e.g.*, 1080p HD). These are interesting, and will explored in future work.

ACKNOWLEDGMENTS

We want to thank the anonymous referees for their valuable comments and helpful suggestions. We also want to thank the participants in our user study, the artists for allowing us to use their works, and authors of [Fišer et al. 2017] for helping us generate the comparison examples with their method.

REFERENCES

Ergun Akleman. 1997. Making caricatures with morphing. In *Proc. ACM SIGGRAPH*. ACM, 145.

Ergun Akleman, James Palmer, and Ryan Logan. 2000. Making extreme caricatures with a new interactive 2D deformation technique with simplicial complexes. In *Proc. Visual.* 165–170.

Susan E Brennan. 2007. Caricature generator: The dynamic exaggeration of faces by computer. *Leonardo* 40, 4 (2007), 392–400.

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017a. Coherent online video style transfer. In *Proc. ICCV*.

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. 2017b. Stylebank: An explicit representation for neural image style transfer. In *Proc. CVPR*.

Hong Chen, Nan-Ning Zheng, Lin Liang, Yan Li, Ying-Qing Xu, and Heung-Yeung Shum. 2002. PicToon: a personalized image-based cartoon system. In *Proc. ACM international conference on Multimedia*. ACM, 171–178.

Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T Freeman. 2017. Synthesizing normalized faces from facial identity features. In *Proc. CVPR*. 3386–3395.

Jakub Fišer, Ondřej Jamriška, David Simons, Eli Shechtman, Jingwan Lu, Paul Asente, Michal Lukáč, and Daniel Sýkora. 2017. Example-based synthesis of stylized facial animations. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 36, 4 (2017), 155.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

Bruce Gooch, Erik Reinhard, and Amy Gooch. 2004. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 23, 1 (2004), 27–44.

Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-image Translation. *arXiv preprint arXiv:1804.04732* (2018).

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *Proc. CVPR*.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*. Springer, 694–711.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192* (2017).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Hiroyasu Koshimizu, Masafumi Tominaga, Takayuki Fujiwara, and Kazuhito Murakami. 1999. On KANSEI facial image processing for computerized facial caricaturing system PICASSO. In *Proc. IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 6. IEEE, 294–299.

Nguyen Kim Hai Le, Yong Peng Why, and Golam Ashraf. 2011. Shape stylized face caricatures. In *Proc. International Conference on Multimedia Modeling*. Springer, 536–547.

Lin Liang, Hong Chen, Ying-Qing Xu, and Heung-Yeung Shum. 2002. Example-based caricature generation with exaggeration. In *Proc. Pacific Conference on Computer Graphics and Applications*. IEEE, 386–393.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088* (2017).

Pei-Ying Chiang Wen-Hung Liao and Tsai-Yen Li. 2004. Automatic caricature generation by analyzing facial features. In *Proc. ACCV*, Vol. 2.

Junfa Liu, Yiqiang Chen, and Wen Gao. 2006. Mapping learning in eigenspace for harmonious caricature generation. In *Proc. ACM international conference on Multimedia*. ACM, 683–686.

Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*. 700–708.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proc. ICCV*.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proc. ICCV*. IEEE, 2813–2821.

Zhenyao Mo, John P Lewis, and Ulrich Neumann. 2004. Improved automatic caricature by feature normalization and exaggeration. In *ACM SIGGRAPH Sketches*. ACM, 57.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *Proc. of NIPS*.

Lenn Redman. 1984. *How to draw caricatures*. Vol. 1. Contemporary Books Chicago, IL.

Ahmed Selim, Mohamed Elgharib, and Linda Doyle. 2016. Painting style transfer for head portraits using convolutional neural networks. *ACM Trans. Graph. (Proc. of SIGGRAPH)* 35, 4 (2016), 129.

Rupesh N Shet, Ka H Lai, Eran A Edirisinghe, and Paul WH Chung. 2005. Use of neural networks in automatic caricature generation: an approach based on drawing style capture. (2005).

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Tamás Szirányi and Josiane Zerubia. 1997. Markov random field image segmentation using cellular neural network. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 44, 1 (1997), 86–89.

Yaniv Taigman, Adam Polyak, and Lior Wolf. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200* (2016).

Chien-Chung Tseng and Jenn-Jier James Lien. 2007. Synthesis of exaggerative caricature with inter and intra correlations. In *Proc. ACCV*. Springer, 314–323.

Fei Yang, Lubomir Bourdev, Eli Shechtman, Jue Wang, and Dimitris Metaxas. 2012. Facial expression editing in video using a temporally-smooth factorization. In *Proc. CVPR*. IEEE, 861–868.

Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. *arXiv preprint* (2017).

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017a. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss. In *Proc. ICCV*.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. 2017b. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*. 465–476.

Shizhan Zhu, Cheng Li, Chen-Change Loy, and Xiaoou Tang. 2016. Unconstrained face alignment via cascaded compositional learning. In *Proc. CVPR*. 3409–3417.