Abstract—Caricature generation aims to translate real photos into caricatures with artistic styles and shape exaggerations while maintaining the identity of the subject. Different from the generic image-to-image translation, drawing a caricature automatically is a more challenging task due to the existence of various spacial deformations. Previous caricature generation methods are obsessed with predicting definite image warping from a given photo while ignoring the intrinsic representation and distribution for exaggerations in caricatures. This limits their ability on diverse exaggeration generation. In this paper, we generalize the caricature generation problem from instance-level warping prediction to distribution-level deformation modeling. Based on this assumption, we present the first exploration for unpaired CARICature generation with Multiple ExtrAgerations (CariMe). Technically, we propose a Multi-exaggeration Warper network to learn the distribution-level mapping from photo to facial exaggerations. This makes it possible to generate diverse and reasonable exaggerations from randomly sampled warp codes given one input photo. To better represent the facial exaggeration and produce fine-grained warping, a deformation-field-based warping method is also proposed, which helps us to capture more detailed exaggerations than other point-based warping methods. Experiments and two perceptual studies prove the superiority of our method comparing with other state-of-the-art methods, showing the improvement of our work on caricature generation.

Index Terms—Caricature Generation, Image-to-image Translation, Image Warping, Style Transfer

I. INTRODUCTION

A caricature is a pictorial representation of a person by exaggerating his/her most distinctive features in order to create an easily identifiable visual likeness [1]. Caricatures are often used in political satire as a ridiculous person or just for entertainment. Different from other non-photorealistic drawings like portraits [2], sketches [3] or cartoons [4], caricatures are committed to reminding the viewer of the identity of the person through the use of pictorial hyperbole.

Traditionally, artists draw caricatures by measuring the differences between the unique person and the average human face. While drawing caricatures, the artists are less bound by the constraints of reality, so that the facial features may be exaggerated beyond the possible. For example, an artist observes that someone’s nose is bigger than average, so in caricature, the nose becomes likewise larger. With suitable exaggerations, the ‘likeness’ of the caricatures may be even stronger than the real photos. However, the creation of a caricature usually takes great effort from a skillful artist, which is not convenient and friendly for others. Therefore, it will be interesting and meaningful to automatically generate caricatures from a given real photo.

Automatically synthesizing a caricature from a real photo is not trivial in the field of computer vision. Despite drawing the photo with a caricature texture style, we should also take spatial exaggerations into consideration [5], [6]. In fact, the exaggeration issue in faces is still an open problem in related researches ranging from detection [7], recognition [8] to generation [5]. To address the exaggeration issue, some methods apply extra information such as user interaction [9] or 3D information [10], [11] to guide the exaggerations. Some methods implicitly achieve the exaggerations using deep neural networks in an image-to-image translation manner [12], [13], [14]. There are also methods that learn point-based warping [5], [6] to translate real photos into caricatures.

However, there are still two problems remain unsolved. First, most of these methods produce a definite exaggeration from an input photo in a one-to-one mapping manner. We summarize them as instance-level mapping based methods. In the actual situation, however, the exaggerations of caricatures may differ among different professional artists [7]. Therefore, the instance-level mapping which only searches a local suboptimal solution should be reconsidered. Second, global affines or point-level sparse transformations mainly focus on changing overall facial contours, which makes the deformations more like global extrusions with less fine-grained local detail. This will make the deformations generated lack of reasonable local exaggerations. A latest method [15] proposes to use the deformation field on caricature generation, but it requires paired warped photos created by professional artists, which is still laborious.

In this paper, we present a novel multi-exaggeration caricature generation framework to tackle the problems above. Different from previous caricature generation methods, we assume the exaggeration patterns can be described as a learnable high dimensional distribution which can be represented by a low-dimensional warp code. Based on this assumption, we model the deformations in caricatures by learning a distribution-level mapping from the latent space to the exaggeration space. Given one input photo, random warp codes are sampled from a normal distribution to produce multiple exaggerations for this photo. Furthermore, to overcome the problem that affine transformation or point-based warping cannot produce meaningful fine-grained warping, our method is based on learning deformation field which can produce fine-grained image warping for caricature generation from
unpaired data. A deformation field is a position map specifying the sampling locations for each pixel in an image. Image transformations can be effectively conducted by any sampling method using a deformation field.

In addition, to achieve photo-specific deformation, a content code is also introduced to extract content information from input photos. During test, a Warper network takes a random warp code along with the content code as inputs to produce a meaningful exaggeration. By changing the warp codes, we can obtain multiple deformations while maintaining the identity of the given photo.

The main contributions of this paper can be summarized as follows:

- We present a novel CariMeature generation framework with Multi Exaggerations, CariMe, which can generate caricatures with both multiple exaggerations and multiple styles. This is the first work on multiple-exaggeration caricature generation.
- We propose a novel warping method based on deformation fields for unpaired caricature generation, which can effectively learn the spatial transforms distribution from real photos to caricatures. An auxiliary content code is also introduced to help our method to produce meaningful and photo-specific exaggerations.
- Experiments along with two perceptual studies show that CariMe can not only generate diverse exaggerations but also produce high-quality caricatures with better identity preservation and runtime performance, which shows the superiority of our method.

II. RELATED WORK

In this section, we introduce three kind of related work in this paper: geometric warping generation, neural style transfer and automatic caricature generation.

A. Geometric Warping Generation

There has been a rich amount of research on generating geometric warping in visual and media tasks [16], [17]. These works can be divided into three categories: parameter-based, point-based, and field-based. Parameter methods [18], [19] apply a global transformation on features via a set of parameters. Point-based methods [20], [21] learns to predict critical control points(e.g., facial landmarks) to guide the geometric transformation between two image domains. However, due to the limitation of the expressiveness of few parameters or sparse control points, these methods pay their most attention to global deformations and ignore detailed transforms in local areas. Field-based methods directly generate the deformation field which is used to perform sampling to produce image transformation. DeepWarp [22] performs coarse-to-fine warping of individual pixels to gaze redirection in a small region like eyes. Deformable VAE [23] learns displacement for the coordinates of each pixel and performs geometric warping, such as stretching and rotation. These methods mainly focus on generating deformation to avoid visual distortion and preserve disparity consistency [24], rather than generating diversified deformation especially. In contrast to these methods, we aim to learn the rules of exaggerations to generate caricatures. Our model learns global deformations that can represent exaggerations with fine-grained detail in caricature, which is naturally more suitable for caricature generation task.

B. Neural Style Transfer

Style transfer is one kind of image synthesis problem which aims to render a content image with different styles [25]. Gatys et al. [2] make the first exploration in neural style transfer by extracting hierarchical features from a VGG network [26] and rendering styles through an optimization process. WCT [27] and Adaptive instance normalization (AdaIN) [28] represents image style as second-order statistics of deep features and replace the optimization process with direct feature transformation. Adaptive layer-instance normalization (AdaLIN) [29] makes improvements over AdaIN by combining instance normalization and layer normalization together in the style stripping process. Recently, with the great ability of generative adversarial networks (GANs) [30] to fit a data distribution, quite a lot of GAN-based methods are proposed [31], [32]. UNIT [33] and MUNIT [34] assume a shared latent space across images. Upon this assumption, they can generate multiple styles from a common latent space. CycleGAN [35] achieves unpaired image translation with a cycle consistency loss. StarGAN [36] and StarGAN-v2 [37] learn mappings among multiple image domains with one single generator. However, directly learning the photo-to-caricature mapping in an image-to-image translation manner makes it hard for the model to capture the geometric transformation. In our method, we introduce a multi-exaggeration Warper module. This module can generate diverse and reasonable deformation fields that can be applied to warping photos into caricatures.

C. Automatic Caricature Generation

There are several automatic caricature generation works proposed before. CariGANs [5] first disentangles the generation into two sub-models. The first is a CycleGAN-based model (CariGeoGAN) which is trained on dense landmarks in the PCA subspace to perform a one-to-one geometry deformation from photo landmarks to caricature landmarks. The second is an MUNIT-based model (CariStyGAN), which transfers the texture to non-photorealistic caricature style.
A. Overall Framework

Let \( x^p \) be an image of real photo and \( x^c \) be an image of real caricature. We assume the image can be disentangled into two latent representations \((C, S)\), where \( C \) is the content representation capturing the spatial structure information and \( S \) maintains the style information which is often reflected by texture and color[38]. The goal of CariMe to translate \( x^p \) into a caricature with multiple geometric exaggerations along with various texture styles. To achieve multiple exaggerations, we propose a Multi-exaggeration Warper module which takes a photo and a latent warp code as inputs to produce an exaggerated photo. For various texture, we propose a Styler module which employs a latent style code to generate a caricature-like image while maintaining the spatial structure of this photo. Given an input photo \( x^p \), the geometric exaggeration is controlled by a warp code \( z_w \), while the texture rendering controlled by a style code \( z_s \). The whole framework of CariMe is shown in Fig. 1. Important notations used in this paper are illustrated in Table II.

TABLE II: Illustration for mathematical notations

| Name   | Meaning          | Name   | Meaning          |
|--------|------------------|--------|------------------|
| \( x^p \) | real photo      | \( E_p \) | photo encoder    |
| \( x^c \) | real caricature | \( G_p \) | photo decoder    |
| \( L_p \) | photo landmark  | \( E_w \) | warp encoder     |
| \( L_c \) | caricature landmark | \( G_w \) | warp decoder     |
| \( L_m \) | mean landmark   | \( E_c \) | content encoder  |
| \( z_s \) | random style code | \( E_s \) | style encoder    |
| \( z_w \) | random warp code | \( G_s \) | style decoder    |
| \( z_p \) | photo content code | \( D \) | Discriminator    |

WarpGAN [6] gather the warping and styling period in one single model. This method directly predicts a set of control points rather than facial landmarks. The control points warp a photo into a caricature without the supervision of landmarks. Besides, WarpGAN applies an identity-preservation adversarial loss to maintain the identity, this may bring one limitation that the discriminator needs to be pre-trained on an auxiliary dataset to obtain the ability to identify.

AutoToon [15] uses deformation field to implement the exaggerations, which is very close to our work. But our framework is different from AutoToon in two aspects. First, AutoToon needs artist-warped photos to supervise the learning of warping field, which involves substantial skills and hard to obtain. Secondly, same with CariGANs and WarpGAN, AutoToon considers the generation problem as a one-to-one mapping, thus it also cannot generate diverse exaggerations.

In this paper, we follow the assumption [5], [15] that disentangles caricature generation into two steps: texture style transfer and shape exaggeration. However, we also assume the exaggerations should be represented a distribution-level mapping from a latent code distribution to a deformation distribution. This endows our model the ability to generate diverse exaggerations by sampling warp codes, and to produce fine-grained image warping with the identity preserved.

III. THE PROPOSED METHOD

A. Overall Framework

Let \( x^p \) be an image of real photo and \( x^c \) be an image of real caricature. We assume the image can be disentangled into two latent representations \((C, S)\), where \( C \) is the content representation capturing the spatial structure information and \( S \) maintains the style information which is often reflected by texture and color[38]. The goal of CariMe to translate \( x^p \) into a caricature with multiple geometric exaggerations along with various texture styles. To achieve multiple exaggerations, we propose a Multi-exaggeration Warper module which takes a photo and a latent warp code as inputs to produce an exaggerated photo. For various texture, we propose a Styler module which employs a latent style code to generate a caricature-like image while maintaining the spatial structure of this photo. Given an input photo \( x^p \), the geometric exaggeration is controlled by a warp code \( z_w \), while the texture rendering controlled by a style code \( z_s \). The whole framework of CariMe is shown in Fig. 1. Important notations used in this paper are illustrated in Table II.

B. Multi-exaggeration Warper

Given an input real photo, the purpose of Multi-exaggeration Warper is to generate diverse but meaningful deformations that can be used to perform the exaggerations on it. We start by using a latent warp code that follows the normal distribution to represent an exaggeration mode. Then with an extra content code encoded from the input photo, we obtain a deformation field that is conditioned on both the exaggeration and the input photo. The reason for combining both the warp code and content code is that, when drawing caricatures, we should consider not only the exaggeration but also the content of the photo (e.g., pose, expression). In this case, we are trying to learn a mapping from a joint distribution of the geometric exaggeration and the image to the distribution of the deformation field. We define this as a distribution-level mapping problem. The advantage of learning a distribution-level many-to-many mapping is that the distribution of the common exaggeration patterns can be better represented, making it possible to generate various deformations.

Fig. 2 shows the detailed architecture of the Warper module. Let \( L_p \) and \( L_c \) denote the facial landmarks of the real photo and caricature, respectively. We first calculate the mean landmarks \( L_m \) of all the caricatures to represent a face with no exaggeration. After that, a deformation field \( F_{m \rightarrow c} \in \mathbb{R}^{W \times H \times 2} \) can be obtained for each caricature, which we respect to represent the specific exaggeration pattern in this caricature.
In the training stage, a warp encoder $E_w$ encodes $F_{m→c}$ into a normalized low-dimensional vector $z_w$, which is called a warp code. Meanwhile, we introduce an auxiliary photo-specific content code $z_p$ to capture content-aware information (i.e., pose, expression, identity) for the input photo $x^p$. This content code is obtained by synchronously training a photo encoder $E_p$ along with a photo decoder $G_p$.

In order to preserve the continuity the latent code space, the content code $z_p$ and warp code $z_w$ are first normalized to normal distribution, and then concatenated together to reconstruct the photo-specific deformation fields $F_{p→c}$ by a warp decoder $G_w$. In this way, a distribution-level bijection mapping is established between the exaggeration space and the latent space. In the test stage, a random $z_w \sim \mathcal{N}(0, 1)$ is sampled with the same dimension with $z_p$ to obtain diverse exaggerations. The deformation field is downsampled to $\mathbb{R}^{(W/2) \times (H/2) \times 2}$ before being fed into the warp encoder, and is upsampled to $\mathbb{R}^{W \times H \times 2}$ after the warp decoder. This operation makes the deformation field more smooth and noiseless.

There are three loss functions used in the Warper, i.e., a warp reconstruction loss $L_{\text{rec}}^{\text{warp}}$, a photo reconstruction loss $L_{\text{rec}}^{p}$, and a total variation loss $L_{\text{tv}}$.

**Warp Reconstruction Loss.** Given a photo $x^p$ and a caricature $x^c$, $F_{p→c}$ denotes the deformation flow from $x^p$ to $x^c$. We have $z_w = E_w(F_{p→c})$ and $z_p = E_p(x^p)$. Since $z_w$ encodes the exaggeration pattern and $z_p$ contains the content information, the warp decoder $G_w$ should be able to reconstruct $F_{p→c}$.

$$L_{\text{rec}}^{\text{warp}} = \mathbb{E}_{x^p \sim X^p, x^c \sim X^c} \left[ \| G_w(z_w, z_p) - F_{p→c} \|_1 \right],$$  

where $\| \cdot \|_1$ denote the $\ell_1$ norm of a matrix.

**Photo Reconstruction Loss.** To encourage $z_p$ to maintain the content and spatial information in $x^p$, the input photo $x^p$ should be able to be reconstructed from $z_p$ so that the content code $z_p$ is awareness on the content of the photo. Since it is difficult to directly reconstruct images from low dimensional vectors, we use the feature maps produced by the last convolutional layer in the photo encoder to reconstruct the input photo.

$$L_{\text{rec}}^{p} = \mathbb{E}_{x^p \sim X^p} \left[ \| G_p(E_{p}^{-1}(x^p)) - x^p \|_1 \right],$$  

where $E_{p}^{-1}$ denotes the activation of last layer in $E_p$.

**Total Variation Loss.** To encourage the warp decoder $G_w$ to produce smooth exaggerations, we apply a total variation regularization term to constrain the variation between adjacent pixels in the image. Given an input photo $x^p$ and a random warp code $z_w \sim \mathcal{N}(0, 1)$, a warped photo $I$ can be obtained by adjusting the deformation field $G_w(z_w, E_p(x^p))$ for this input photo using bi-linear sampling. The regularization term is formulated as follows:

$$L_{\text{tv}}(I) = \sum_{i=1}^{H-1} \sum_{j=1}^{W-1} \sum_{k=1}^{3} (I_{i+1,j,k} - I_{i,j,k})^2 + \sum_{i=1}^{H} \sum_{j=1}^{W-1} \sum_{k=1}^{3} (I_{i,j+1,k} - I_{i,j,k})^2.$$

The full objective of the Warper is as below:

$$L_W = \lambda_{\text{rec}}^{\text{warp}} L_{\text{rec}}^{\text{warp}} + L_{\text{rec}}^{p} + \lambda_{\text{tv}} L_{\text{tv}},$$

where $\lambda_{\text{rec}}^{\text{warp}}$ and $\lambda_{\text{tv}}$ are hyper-parameters.
Fig. 3: The training phase of the Styler. The Styler should be able to reconstruct both the input photo and caricature, and meanwhile it tries to generate a fake caricature that the discriminator can not figure out from the real caricatures.

C. Styler

The Styler aims to translate an input image to a caricature. Inspired by recent work on image-to-image translation [39], [40], we disentangle the image into a content representation \( c \) and a style code \( s \). Our Styler contains three modules: a content encoder \( E_c \), a style encoder \( E_s \) and a style decoder \( G_s \). Given an input image \( x \in \mathbb{R}^{W \times H \times C} \), \( E_c \) encodes \( x \) into a spatial feature map \( E_c(x) \) which retains the spatial structure information, while \( E_s \) encodes \( x \) into a style vector \( E_s(x) \) which represents the image style. For photo-to-caricature translation, a latent style code \( z_s \) is sampled from a normal distribution, then \( E_s(x^p) \) is processed by \( G_s \) to get a caricature-like image with the style controlled by \( z_s \). The overview of the Styler is shown in Figure 3.

We extend AdaLIN [29] to implement our style decoder. Let \( a \) denote the feature map produced by the content encoder \( E_c(x) \), both instance normalization and layer normalization are performed to get two normalized features \( a_{IN} \) and \( a_{LN} \). We combine them together with a learnable weight \( p \) and use several fully connected layers to predict the de-normalizing parameters \( (\gamma, \beta) \) from the style code \( z_s \).

\[
AdaLIN(a) = \gamma \cdot (p \cdot a_{IN} + (1 - p) \cdot a_{LN}) + \beta. \tag{5}
\]

After several residual blocks with AdaLIN [29], the style decoder \( G_s \) produces a caricature-like image from the processed feature map without exaggeration.

We use three loss functions to optimize the Styler, i.e., an adversarial loss \( L_{adv} \), an image reconstruction loss \( L_{img} \), and a cycle consistency loss \( L_{cyc} \).

Adversarial Loss. We employ adversarial loss to encourage the Styler to generate fake caricatures that the discriminator \( D \) can not figure out from real ones. Here we adopt the Least Squares GAN’s [41] objective.

\[
L_{adv} = \mathbb{E}_{x_c \in X_c} [(D(x_c))^2] + \mathbb{E}_{x_p \in X_p, z_s \sim p(s)} [(1 - D(G_s(E_c(x^p)), z_s))^2]. \tag{6}
\]

Image Reconstruction Loss. For the module of style transfer, given an image from either photo domain or caricature domain, we reconstruct it after encoding and decoding.

\[
L_{img}^{rec} = \mathbb{E}_{x \in \{X_p \cup X_c\}} \| G_s(E_c(x), E_s(x)) - x \|_1. \tag{7}
\]

Cycle Consistency Loss. For the generated image \( G_s(E_c(x^p), z_s) \) with a random style code \( z_s \), the style encoder \( E_s \) should be able to encode the style code from the generated image, and the content encoder \( E_c \) should be able to encode the content feature again.

\[
L_{cyc} = L_{cyc}^{con} + L_{cyc}^{sty} = \mathbb{E}_{x^p \in X_p, z_s \sim p(s)} \| E_c(G_s(E_c(x^p), z_s)) - E_c(x^p) \|_1 + \mathbb{E}_{x_p \in X_p, z_s \sim p(s)} \| E_s(G_s(E_c(x^p), z_s)) - z_s \|_1. \tag{8}
\]

The full objective of the Styler is as below:

\[
L_S = L_{adv} + \lambda_{img}^{rec} L_{img}^{rec} + \lambda_{cyc} L_{cyc}. \tag{9}
\]

D. Training Details

We train all of our networks using the Adam optimizer [42] with \( \beta_1 = 0.5 \) and \( \beta_2 = 0.999 \). For data augmentation, we horizontally flip the images, resize them to \( 288 \times 288 \), and random crop them to \( 256 \times 256 \) with a probability of 0.5. The warper is trained using photo-caricature pairs from the same identity to learn more accurate warping patterns. The Styler is trained using photo-caricature pairs from randomly sampled identities (may be different) to get better style representation. The Warper is trained for 10,000 iterations with a fixed learning rate of 0.0001 and another 10,000 iterations with the learning rate linearly decayed to 0. The Styler is trained with a fixed learning rate of 0.0001 for 250,000 iterations and another 250,000 iterations with the learning rate linearly decayed to 0. As for the hyper-parameters, we set \( \lambda_{img}^{rec} = 10 \), \( \lambda_{warp} = 10 \), \( \lambda_{cyc} = 1 \) and \( \lambda_{tv} = 0.000005 \).
IV. EXPERIMENTS

A. Experimental Settings

The experiments are performed on a large caricature dataset WebCaricature [43]. This dataset contains 5,974 photos and 6,042 caricatures from 252 identities. We use the 17 facial landmarks officially provided by the dataset to pre-process these images. To be specific, we first rotate each image by aligning the two eyes to the horizontal position according to four eye corners. Then, a bounding box can be obtained according to the face contour. Note that in many face recognition tasks, the images are usually resized without keeping the aspect ratio. But this is not suitable for generation tasks because the geometric distribution of human faces will be destroyed. Therefore, we also limit the bounding box to square to keep the aspect ratio of faces. After that, we enlarge the bounding box with a scale of 1.3 in both width and height and resize it to $256 \times 256$. For data split, following WarpGAN [6], we randomly selected 126 identities (3,036 photos and 3,005 caricatures) for training and the remaining 126 identities (2,938 photos and 3,037 caricatures) for test.

B. Comparison with Image-to-image Translation Methods

First we compare our method with several state-of-the-art image-to-image translation methods, including UNIT [33], MUNIT [34], CycleGAN [35] and StarGAN-v2 [37]. All the methods are implemented using the officially released codes.

Fig. 4 shows the results of our method compared with the above four general image-to-image translation methods. As can be seen, UNIT, MUNIT produce images with part of detailed semantic information lost. We believe this is because the large deformation gap between photos and caricatures makes the task harder for these methods. Although CycleGAN and StarGAN-v2 can generate clear outputs, the generated caricatures are somewhat identical to input photos with texture transfer but few deformations. The last three columns show the result of our proposed method. It can be seen that the output contains diverse facial exaggerations which are more like real caricatures. Table III shows the comparison of Frchet Inception Distance [44], indicating that CariMe achieves a lower FID score than other methods.

C. Comparison with Caricature Generation Methods

In this section, we make a comparison between our CariMe and three caricature generation methods CariGANS [5], WarpGAN [6] and AutoToon [15]. We implement the warping module of CariGANS (i.e., CariGeoGAN) by ourselves. For WarpGAN, we use the officially provided code[1] and train the model using our training set. For AutoToon, we use the officially pretrained model[2] as our baseline since we do not have paired data. We first take a brief introduction to these methods and make a discussion on the results.

CariGANS [5] follows the structure of CycleGAN, but the inputs are vectors in the PCA space instead of images. After training, the model is used to predict the target landmark positions for the testing photos, then perform image warping on them from source facial landmarks to the target landmarks. We use the 17 landmarks provided by the WebCaricature dataset as supervision to train our CariGANS baseline.

WarpGAN [6] uses one end-to-end model to generate both styles and warping at the same time. The style part is implemented using AdaIN [28]. The warping module of WarpGAN will predict a set of control points and their target positions.

### Table III: Comparison on Frchet Inception Distance

| Method     | FID Score |
|------------|-----------|
| UNIT [33]  | 40.84     |
| MUNIT [34] | 43.65     |
| CycleGAN [35] | 43.96   |
| StarGAN-v2 [37] | 50.28   |
| WarpGAN [6] | 50.35     |
| CariMe(ours) | 33.56     |

1https://github.com/seasonSH/WarpGAN
2https://github.com/adobe-research/AutoToon

Fig. 4: Comparison of the proposed CariMe with general image-to-image translation methods. The last three columns show the results of our CariMe using different warp codes and style codes sampled from the normal distribution.
Fig. 5: Comparison of the proposed CariMe with other state-of-the-art caricature generation methods with both exaggeration and style rendering. The input and output images of CariGANs and AutoToon are selected from their papers. Our CariMe presents various output caricatures with different style and exaggeration while maintaining the identity well.

for every input photo. The warping period is then performed using a sparse image warping method [20]. Moreover, the discriminator is pre-trained on an auxiliary dataset.

AutoToon [15] learns a direct mapping from an input photo to a warping flow from paired data. The deformation field is required to make the warped photo to be close to the ground-truth warped photo that annotated by human beings. The stylization period of AutoToon is implemented using CartoonGAN [4].

Fig. 5 shows our comparison with these methods. The input photos in the first three rows are from CariGANs paper and the last two rows are from AutoToon paper. The results for CariGANs and AutoToon are from their papers. As can be seen, the deformation of CariGANs [5] is more like some squeezes on the facial features. For example, the caricature generated by CariGANs in the third line is squeezed down, making the caricature not like the same person. WarpGAN [6] may produce unclear artifacts in some local areas. As for AutoToon [15], the caricatures are less detailed and also present incoherent lines around the facial contour. The last four columns are caricatures given by our method. Each output is generated from a randomly sampled warp code and a randomly sampled style code. More results in our test set are given in Fig. 6. We also make comparison with only image warping in Fig. 7. It can be seen that the outputs generated by CariMe are obviously in different exaggerations and the identity can also be kept. These results show the ability of our method to produce multiple and fine-grained exaggerations with fewer supervision than other methods.

D. Controlling Both Style and Deformation

Fig. 8 exhibits the interpolation results with two randomly sampled pairs of style codes \((z_{s1}, z_{s2})\) and warp codes \((z_{w1}, z_{w2})\). Then we linearly interpolate style codes and warp codes inside to generate different outputs. When observing from top to bottom, the images are gradually colored (i.e., stylized). While observing from left to right, the exaggeration changes smoothly from one exaggeration pattern to another (e.g., a small nose becomes a big one). This shows the smoothness of the learned code space.
Fig. 6: Our method allow users to control on both style and exaggeration. Input photos are shown in the first column. The second to forth column show the results with random sampled style codes and warp codes. The last three columns show the results with the exaggeration guided by the reference exaggerations and the same style code.
Fig. 7: Comparison of the proposed CariMe with other state-of-the-art caricature generation methods with only photo warping. Our CariMe can generated obvious and diverse exaggerations in the last three columns.

Fig. 8: Interpolation in both styles and deformations. The upper left corner is the input photo. The images in the same row share the same style, while the images in the same column have the same exaggeration.

Fig. 9: The warped photos and deformation fields after removing different loss terms in the Warper. The Warper cannot produce meaningful exaggerations without neither content information nor total variation loss.
photos without caricature style transfer for every testing photo. In this section, we make a discussion on the identity preservation issue. In the first row, the exaggeration tends to present a longer face, smaller noise and smaller eyes with the scale increases. In the second row, it turns to present a larger chin, more extreme laugh style since the input photo is a smile one.

E. Ablation Study

For the newly designed Multi-exaggeration Warper in the proposed CariMe, an ablation study is performed to test the influence of different components. We remove several loss functions to demonstrate effectiveness. The results of removing $L_{rec}$ and $L_{tv}$ are reported in Fig. 9. Without the photo reconstruction loss $L_{rec}$, the deformation field is less bounded and is too distorted, and even cannot be used to produce meaningful exaggerations. Without the total variation loss $L_{tv}$, there appears some cracks appear in local areas. With both of these losses, our model can produce smooth and convincing exaggerations.

F. Controlling Exaggeration Scale

Same as previous methods like WarpGAN [6] and CariGANs [5], our method also allows customization on the exaggeration extent. This can be implemented by alternatively generating the residual deformation flow for each pixel in the sampling map. The deformation flow is multiplied by a scaling coefficient to control the exaggeration scale. Fig. 10 shows the results of applying different scales on the input photo. With the scale increasing, the exaggeration becomes bigger while still being reasonable.

V. ANALYSIS

A. Identity Preservation

One concern about caricature generation methods is whether the identity can be preserved. In this section, we make a discussion on the identity preservation issue.

We first propose two assumptions in this experiment. First, a good caricature should contain both large exaggeration and high accuracy (e.g., when the shape transformation is small, the accuracy should be high, but it may not be a good caricature). Identity preservation should be measured while keeping the degree of exaggeration be the same for different methods. Second, since the caricature styles may interfere with the evaluation of deformation quality, we only generate warped photos without caricature style transfer for every testing photo.

Therefore, we should generate warped photos with the same degree of exaggeration by controlling the scale factor of different caricature generation methods. To achieve this goal, for each input photo $P \in \mathbb{R}^{W \times H \times 3}$, we first generate the corresponding deformation flow $F \in \mathbb{R}^{W \times H \times 2}$, then calculate the exaggeration degree for $P$ as follows:

$$Degree(P) = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} ||F_{i,j}||_2}{H \times W}. \quad (10)$$

The warp degree for every method is calculated by averaging the degrees for all testing photos (2,938 photos in total). This metric provides a unified standard for measuring the degree of exaggeration between different methods.

After we obtain warped photos in the same warp degree, the identity can be preserved. In this section, we make a discussion on the identity preservation issue. In the first row, the exaggeration tends to present a longer face, smaller noise and smaller eyes with the scale increases. In the second row, it turns to present a larger chin, more extreme laugh style since the input photo is a smile one.

The result shows CariMe can better preserve the identity than other methods, especially in large exaggerations. We also
find that the scale factors required to uniform the degree are different among different methods. For example, for small exaggeration, CariMe needs a scale of 0.3 to get warp degree 3.19, while AutoToon needs a scale of 1.2 to get warp degree 3.06, which also proves our first assumption above.

### B. Perceptual Study

To further evaluate our method, we conduct two perceptual studies in terms of exaggeration and overall quality. 20 volunteers are invited to our perceptual studies. The volunteers are first given 20 real caricatures to get familiar with real exaggerations, then asked to compare caricatures generated by our method with other caricature generation methods.

The first study is a warping quality test. In this study, we randomly choose 30 photos with different identities in the test set. For each photo, we apply the warping module of different methods (i.e., CariGANs [5], WarpGAN [6], AutoToon [15]) to generate warped photos without stylization to avoid the influence of texture. Since our method is the only approach for multiple exaggerations, we use two random warp codes for every photo (i.e., CariMe-random1, CariMe-random2). However, the volunteers are told that the outputs are generated from five different methods. Then we ask the volunteers to score how much the exaggerations look like real caricature-style exaggerations. The scores range from 1 to 5 where a higher score means better warping. The results are reported in Table V where our method achieves 3.36 and 3.74 for warp quality independently. We also present the percentage of each method in our perceptual study in Fig. 11 which shows a more detailed comparison.

The second study is an overall quality test. Similarly, we randomly choose another 30 photos with different identities in the test set. We first produce warped photos using different caricature generation methods for each photo (same as the first study). Then we apply a stylization network (the Styler network in this paper) on these photos. We believe it will contribute to a more fair comparison by using the same stylization method. We also use the whole WarpGAN model as our baseline to evaluate the effectiveness of the Styler (WarpGAN-baseline). The scoring period is the same as the first study. The result is given in Table VI where our method outperforms on other methods. Moreover, we can see that for the same warped photo, applying the proposed Styler network can achieve better user acceptance (2.81) than our baseline (2.15).

### C. Performance

Our core algorithm is developed in PyTorch 1.4 [47]. All of our experiments are conduct on an NVIDIA V100 GPU. To evaluate the time cost for each method, we calculate the total time to generate warped photos for all testing photos (i.e., 2938 photos) and get the average reasoning time. The runtime comparison is given in Table VI. As can be seen, CariMe earns the first order in runtime performance, taking 0.003 second to process one image. This is because the generated deformation field can be directly used to warp the photo by a simple sampling without other interpolate calculations like point-based methods. We also find that CariGANs take less time for geometric warping than AutoToon. This may be due to the use of fewer landmarks for CariGeoGAN in our experiment (68 to 17). The upscaling of deformation fields in AutoToon from $32 \times 32$ to $256 \times 256$ also increases complexity.

### VI. Conclusion And Future Work

In this paper, we propose a novel framework for multi-exaggeration unpaired caricature generation. We propose a deformation field based warping method which considers both exaggeration pattern and image content to produce diverse caricature exaggerations. Our approach supports flexible controls to change shape exaggerations, warping degrees, and appearance styles. Experiments and perceptual studies demonstrate that the proposed method generates caricatures...
that are superior to other state-of-the-art caricature generation methods. Although our method can achieve compelling results in many cases, there still exist problems that need to be tackled in caricature generation. Fig. 12 shows several typical failure cases. The first line shows a failure example in large exaggerations where some content is destroyed, which also occurs in other caricature generation methods. The second line shows a failure example where the deformation field is in chaos no matter what warp code is sampled. We analyze this is caused by a badly learned content code for some isolated outliers. A possible way to tackle this problem is to enhance the representation ability of the photo encoder and feed it with more data. We will explore these interesting but unsolved problems in the future.

REFERENCES

[1] S. B. Sadimon, M. S. Sunar, D. Mohamad, and H. Haron, “Computer generated caricature: A survey,” in Proc. Lect. Notes Comput. Sci., 2010, pp. 383–390.

[2] L. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis using convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Sys., NeuIPS, 2015.

[3] R. Yi, Y.-J. Liu, Y.-K. Lai, and P. L. Rosin, “Unpaired portrait drawing generation via asymmetric cycle mapping,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2020, pp. 8217–8225.

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

[5] K. Cao, J. Liao, and L. Yuan, “Carigans: unpaired photo-to-caricature translation,” ACM Trans. Graph., pp. 1–14, 2018.

[6] Y. Shi, D. Deb, and A. K. Jain, “Warpgan: Automatic caricature generation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2019, pp. 10762–10771.

[7] J. Yaniv, Y. Newman, and A. Shamir, “The face of art: landmark detection and geometric style in portraits,” ACM Trans. Graph., p. 60, 2019.

[8] H.-C. Shin, J. H. Park, and S.-D. Kim, “Combination of warping robust elastic graph matching and kernel-based projection discriminant analysis for face recognition,” IEEE Trans. Multimedia, vol. 9, no. 6, pp. 1125–1136, 2007.

[9] L. Liang, H. Chen, Y.-Q. Xu, and H.-Y. Shum, “Example-based caricature generation with exaggeration,” in Proc. Pacif Conf Comput Graph Appl, 2002, pp. 386–393.

[10] X. Han, C. Gao, and Y. Yu, “Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling,” ACM Trans. Graph., p. 126, 2017.

[11] Q. Wu, J. Zhang, Y.-K. Lai, J. Zheng, and J. Cai, “Alive caricature from 2d to 3d,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2018, pp. 7336–7345.

[12] X. Han, K. Hou, D. Du, Y. Qiu, S. Cui, K. Zhou, and Y. Yu, “CaricatureShop: Personalized and photorealistic caricature sketching,” IEEE Trans. Vis. Comput. Graphics, 2018.

[13] Z. Zheng, C. Wang, Z. Yu, N. Wang, H. Zheng, and B. Zheng, “Unpaired photo-to-caricature translation on faces in the wild,” Neurocomputing, vol. 355, pp. 71–81, 2019.

[14] W. Li, W. Xiong, H. Liao, J. Huo, Y. Gao, and J. Luo, “Carigan: Caricature generation through weakly paired adversarial learning,” Neural Netw., vol. 132, pp. 66–74, 2020.

[15] J. Gong, Y. Hold-Geoffroy, and J. Lu, “Autoatoon: Automatic geometric warping for face cartoon generation,” in Proc. Br. Mach. Vis. Conf., BMVC, 2020, pp. 360–369.

[16] S.-S. Lin, I.-C. Yeh, C.-H. Lin, and T.-Y. Lee, “Patch-based image warping for content-aware retargeting,” IEEE Trans. Multimedia, vol. 15, no. 2, pp. 359–368, 2012.

[17] J. Jin, A. Wang, Y.Zhao, C. Lin, and B. Zeng, “Region-aware 3-d warping for dibi,” IEEE Trans. Multimedia, vol. 18, no. 6, p. 953–966, 2016.

[18] M. Jaderberg, K. Simonyan, A. Zisserman et al., “Spatial transformer networks,” in Proc. Adv. Neural Inf. Process. Syst., NeuIPS, 2015, pp. 2017–2025.

[19] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, “Stgan: Spatial transformer generative adversarial networks for image composi- ting,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2018, pp. 9455–9464.

[20] F. Cole, D. Belanger, D. Krishnan, A. Sama, I. Mosseri, and W. T. Freeman, “Synthesizing normalized faces from facial identity features,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2017, pp. 3703–3712.

[21] W. Wu, K. Cao, C. Li, C. Qian, and C. C. Loy, “Transgaga: Geometry-aware unsupervised image-to-image translation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2019, pp. 8012–8021.

[22] Y. Ganin, D. Kononenko, D. Suntatullina, and V. Lempitsky, “Deepwarp: Photorealistic image resynthesis for gaze manipulation,” in Proc. Eur. Conf. Comput. Vis., ECCV, 2016, pp. 311–326.

[23] X. Xing, R. Gao, T. Han, S.-C. Zhu, and Y. N. Wu, “Unsupervised disentangling of appearance and geometry by deformable generator network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2019, pp. 10346–10355.

[24] X. Fan, J. Lei, Y. Fang, Q. Huang, N. Ling, and C. Hou, “Stereoscopic image stitching via disparity-constrained warping and blending,” IEEE Trans. Multimedia, vol. 22, no. 3, pp. 655–665, 2019.

[25] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, “Neural style transfer: A review,” IEEE Trans. Vis. Comput. Graphics, 2019.

[26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representa- tions, ICLR, 2015.

[27] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal style transfer via feature transforms,” in Proc. Adv. Neural Inf. Process. Syst., NeuIPS, 2017, pp. 386–396.

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

[29] J. Kim, M. Kim, H. Kang, and K. H. Lee, “U-gat-it: Unsupervised gener- ative attentional networks with adaptive layer-instance normalization for image-to-image translation,” in Proc. Int. Conf. Learn. Representations, ICLR, 2019.

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., NeuIPS, 2014, pp. 2672–2680.

[31] Y. Liu, W. Chen, L. Liu, and M. S. Lew, “Swpagan: A multistage generative approach for person-to-person fashion style transfer,” IEEE Trans. Multimedia, vol. 21, no. 9, pp. 2209–2222, 2019.

[32] Y.-F. Zhou, R.-H. Jiang, X. Wu, J.-Y. He, S. Weng, and Q. Peng, “Branchgan: Unsupervised mutual image-to-image transfer with a single encoder and dual decoders,” IEEE Trans. Multimedia, vol. 21, no. 12, pp. 3136–3149, 2019.
[33] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in Proc. Adv. Neural Inf. Process. Syst., NeurIPS, 2017, pp. 700–708.

[34] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” in Proc. Eur. Conf. Comput. Vis., ECCV, 2018, pp. 172–189.

[35] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vis., ICCV, 2017, pp. 2223–2232.

[36] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsupervised image-to-image translation,” in Proc. Eur. Conf. Comput. Vis., ECCV, 2018, pp. 172–189.

[37] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proc. Adv. Neural Inf. Process. Syst., NeurIPS, 2017, pp. 700–708.

[38] 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 Conf. Comput. Vis. Pattern Recog., CVPR, 2018, pp. 8789–8797.

[39] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2020, pp. 8188–8197.

[40] X. Yu, Y. Chen, T. Li, S. Liu, and G. Li, “Multi-mapping image-to-image translation via learning disentanglement,” in Proc. Adv. Neural Inf. Process. Syst., NeurIPS, 2019, pp. 2994–3004.

[41] T. Xiao, J. Hong, and J. Ma, “Elegant: Exchanging latent encodings with gan for transferring multiple face attributes,” in Proc. Eur. Conf. Comput. Vis., ECCV, 2018, pp. 168–184.

[42] S. Gu, J. Bao, H. Yang, D. Chen, F. Wen, and L. Yuan, “Mask-guided portrait editing with conditional gans,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2019, pp. 3436–3445.

[43] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recog., CVPR, 2017, pp. 2794–2802.

[44] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” Proc. Int. Conf. Learn. Representations, ICLR, 2015.

[45] J. Hu, W. Li, Y. Shi, Y. Gao, and H. Yin, “Webcaricature: a benchmark for caricature recognition,” in Proc. Br. Mach. Vis. Conf., BMVC, 2018, p. 223.

[46] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in Proc. Adv. Neural Inf. Process. Syst., NeurIPS, 2017, pp. 6626–6637.

[47] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. Adv. Neural Inf. Process. Syst., NeurIPS, 2017, pp. 6626–6637.

[48] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “Vggface2: A dataset for recognising faces across pose and age,” in Proc. Int. Conf. Autom. Face Gesture Recog., FGR, 2018, pp. 67–74.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., “Pytorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., NeurIPS, 2019, pp. 8026–8037.