Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6216048, 8 pages
https://doi.org/10.1155/2020/6216048

Research Article

Weakly Supervised GAN for Image-to-Image Translation in the Wild

Zhiyi Cao, Shaozhang Niu, and Jiwei Zhang

Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 68545849, China

Correspondence should be addressed to Zhiyi Cao; 68545849@qq.com

Received 19 October 2019; Revised 19 December 2019; Accepted 27 January 2020; Published 9 March 2020

Academic Editor: Ioannis Kostavelis

Generative Adversarial Networks (GANs) have achieved significant success in unsupervised image-to-image translation between given categories (e.g., zebras to horses). Previous GANs models assume that the shared latent space between different categories will be captured from the given categories. Unfortunately, besides the well-designed datasets from given categories, many examples come from different wild categories (e.g., cats to dogs) holding special shapes and sizes (short for adversarial examples), so the shared latent space is troublesome to capture, and it will cause the collapse of these models. For this problem, we assume the shared latent space can be classified as global and local and design a weakly supervised Similar GANs (Sim-GAN) to capture the local shared latent space rather than the global shared latent space. For the well-designed datasets, the local shared latent space is close to the global shared latent space. For the wild datasets, we will get the local shared latent space to stop the model from collapse. Experiments on four public datasets show that our model significantly outperforms state-of-the-art baseline methods.

1. Introduction

The unsupervised image-to-image translation is the process of learning an arbitrary mapping between two categories, domains, or classes images without labels. This greatly reduces the reliance on paired datasets and extends the range of applications for image translation tasks. For example, we can translate zebras into horses. The unsupervised image translation tasks can meet a variety of needs. Previous models assume that the shared latent space between different categories will be captured from the given categories (e.g., zebras to horses). Unfortunately, besides the well-designed datasets from given categories, many examples come from wild categories (e.g., cats to dogs) holding special shapes and sizes (short for adversarial examples), so the shared latent space is troublesome to capture and it will cause the collapse of these models.

Prior works such as CycleGAN [1] enable capturing the cyclic space between well-designed categories (e.g., zebras to horses and summer to winter) without paired data. However, the cyclic mapping of CycleGAN is a one-to-one mapping, in its limitation section; some unnatural results were discovered on special datasets. Recent works such as MUNIT [2] enable us to capture the shared latent content space between well-designed categories (e.g., cats to dogs) without paired data. Though MUNIT enables us to build multiple mapping and generate natural results, it is troublesome to capture the shared latent content space for wild categories with many adversarial examples.

In this paper, we take a further step towards unsupervised image-to-image translation research for wild datasets with many adversarial examples. Our global model can be divided into two parts to explain. For the first stage, inspired by the facenet [3], we use SSIM (structural similarity) [4] algorithm to calculate the distance between different examples. We observed that the average SSIM distance between the most examples and other examples in the well-designed dataset that belong to the same category is greater than 0.3. For the wild dataset, the average SSIM distance between many examples and other examples is less than 0.3, and these examples will cause the model to collapse. Note, it is just one way to discover adversarial examples. In order to automatically reduce the number of adversarial

examples within each category, we introduce a Sim loss based on SSIM (structural similarity) distance. For a wild dataset, we first select 5 to 10 normal examples with a SSIM distance greater than 0.3 from each other as the weakly supervised examples. We assume that a wild dataset can be divided into two parts. One part is the normal examples similar to the weakly supervised examples; the other part is the adversarial examples. Then we use Sim loss to control the learning objects of our weakly supervised model to reduce the number of adversarial examples. Through the first stage, a normal example should learn from itself and an adversarial example should learn from a weakly supervised example.

The Sim loss is beneficial to reduce the number of adversarial examples, but it cannot eliminate all the adversarial examples. For the second stage, we assume the shared latent space can be classified as global and local and design an image-to-image translation model to capture the local shared latent space rather than the global shared latent space. For the well-designed datasets, different examples have similar latent space, so the local shared latent space is close to the global shared latent space. For the wild datasets, only part of examples has similar latent space; thus we only capture the local shared latent space to stop the model from collapse. Inspired by the multimapping [5], we introduce the category codes to constraint the local shared features across categories and the encoders to capture the local shared latent space for image-to-image translation. Experiments on four public datasets show that our model significantly outperforms state-of-the-art baseline methods.

In summary, we propose a similar GAN (short for Sim-GAN) for wild datasets with many adversarial examples. The main contributions of this study are summarized as follows.

(i) We introduce a Sim loss based on SSIM (structural similarity) distance with weakly supervised examples for Sim-GAN to automatically reduce the number of adversarial examples within each category

(ii) We assume that the shared latent space can be classified as global and local and design a model to capture the local shared latent space rather than the global shared latent space

(iii) We introduce the category codes to constraint the local shared features across categories and two encoders to capture the local shared latent space for image-to-image translation

2. Related Work

2.1. Generative Adversarial Networks. There have been large GAN-family methods since the seminal work by Goodfellow et al. [6]. These GANs models [7–9] can map from noise inputs to realistic images. These GANs models have produced promising results in image translation. The Pix2pix model [10] applies a conditional GAN to model the mapping function. Although high-quality results have been shown, model training requires paired training data. It is applied to numerous tasks such as sketch to photo, image colorization, and photo to map.

2.2. Image-to-Image Translation. The CycleGAN [1] model is proposed for unpaired image translation that relies on a cycle consistency loss term. The CycleGAN model showed some success when applied to a range of classic image translation tasks like zebras to horses. Some of its failure cases include overrecognizing objects and not being able to change the shape of the object during translation (e.g., outputting a cat-shaped dog). Other works tackle the greater shape change problems. The Contrast-GAN [11] model introduces an adversarial distance comparison objective for optimizing one conditional generator and several semantic-aware discriminators. The MUNIT [2] model assumes that images in different domains share a common content space but not the style space. The ganimorph [12] model introduces dilated convolutions in their discriminator architecture. Then their discriminator output facilitates more fine-grained information flow from the discriminator to the generator. However, for the wild datasets with many adversarial examples, all of the above models will collapse.

3. Proposed Method

Given examples from a category, such as the images of cats, our goal is to translate them into dogs. In this paper, x and y indicate the examples from categories X and Y. Our method consists of two stages. In the first stage, a weakly supervised model to automatically reduce the number of adversarial examples within each category is trained. Because the dogs' dataset (http://www.recod.ic.unicamp.br/~rwerneck/datasets/flickr-dog/) contains many adversarial examples, the previous models suffered a model collapse when processing the cats-to-dogs task. Therefore, it is necessary to reduce the number of adversarial examples. x_w and y_w indicate the weakly supervised examples from categories X and Y. As long as the SSIM distance between the examples $\{x_w\}_{w=1}^{n}$ is greater than 0.3, we can get the global shared latent space. In our weakly supervised model, the source category and target category are the same. Figure 1(a) shows the network structure of the first stage.

In the second stage, an image-to-image translation model is trained. C_x (e.g., identity matrix) and C_y (e.g., inverse identity matrix) indicate the inverse category codes of the categories X and Y. Inspired by the multimapping [5], we assume that the two-dimensional C_x and C_y can constraint the local shared features across categories X and Y. For the wild datasets with many adversarial examples, the global shared latent space is troublesome to capture. We use two inverse category codes to hide some detailed features for each example and reduce the effect of the adversarial examples to establish the correct mapping. Though we only get the local shared features across categories, this will stop the model from collapse. Then we use two encoders E_x and E_y to capture the local shared latent space for unsupervised image-to-image translation. Figure 1(b) shows the network structure of the second stage.

3.1. Objective. In the first stage, our model contains two mappings $G: X \longrightarrow Y$ and its discriminator D_Y, $F: Y \longrightarrow X$ and its discriminator D_X. To automatically

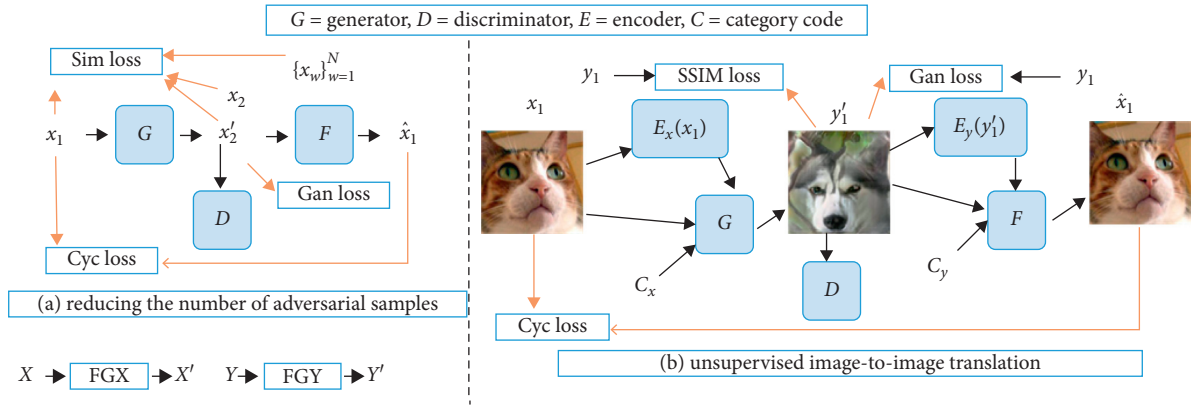

Figure 1: An overview of the two stages of our Sim-GAN. (a) In the first stage, x_1, x_2 indicate the examples of category X. Given an example x_1, our goal is to translate it into an example x_w or x_2. At first, generator G is used to translate an example x_1 into an example x_2'. Then discriminator D distinguishes between the generated example x_2' and real one x_2 or x_w. At last, generator F is used to translate x_2' into $\hat{x}_1$. $\hat{x}_1$ indicates the generated result. In this stage, FGX and FGY mean that we use the first stage to reduce the number of adversarial examples for categories X and Y. Here, Cyc loss denotes the cycle consistency loss between x_1 and $\hat{x}_1$. (b) In the second stage, E_x and E_y indicate the encoders of the categories X and Y. At first, generator G is used to translate x_1, the category code C_x, and the encoded representation of image x_1 into an example y_1'. Then the discriminator D distinguishes between the generated example y_1' and real example y_1.

reduce the number of adversarial examples for each category separately, when we train the model, x_1, x_2 come from the same category and y_1, y_2 come from the same category in this stage. We introduce the adversarial losses firstly. The adversarial losses are usually used to judge the true and false probability of the generated images and the input images. The adversarial losses can be expressed as follows:

$$\mathscr{L}_{\text{Gan}}(G, D_Y, X, Y) = E_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] + E_{x\sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))]. \tag{1}$$

CycleGAN argues that, for each example x from category X, the image translation cycle should be able to bring x back to the original image; that is, $x \longrightarrow G(x) \longrightarrow F(G(x)) \approx x$. It is called forward cycle consistency. Similarly, for each image y from category Y, G and F should also satisfy backward cycle consistency: $y \longrightarrow F(y) \longrightarrow G(F(y)) \approx y$. The cycle consistency loss can be expressed as follows:

$$\mathscr{L}_{\text{Cyc}}(G, F) = E_{x\sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + E_{y\sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]. \tag{2}$$

For the first stage, the source examples and the target examples are selected randomly from the same dataset. To automatically reduce the number of adversarial examples, we use the weakly supervised example x_w (e.g., x_1, x_2 rather x) to generate more normal examples from category X. For an example, given x_1 from category X, the generated example $x_2' = G(x_1)$ should be similar to x_2 when the minimum SSIM distance between x_w and x_2 is bigger than 0.3. In other cases, the generated example $x_2' = G(x_1)$ should be similar to x_w. This leads us to propose a Sim loss (s represents the structural similarity of two tensors):

$$\mathscr{L}_{\text{sim}}(x_2', x_1, x_2, x_W) = \begin{cases} \|x_2' - x_{w_1}\|_s, & M_s < 0.3, \\ \|x_2' - x_2\|_s, & M_s \geq 0.3, \end{cases} \tag{3}$$

where $x_W = \{x_w\}_{w=1}^n$ (it represents all the weakly supervised examples) and $M_s = \text{Min}\{\text{Min}\{\|x_2' - x_w\|_s\}_{w=1}^n, \|x_1 - x_2\|_s\}$ (this represents the minimum SSIM distance between the input examples x_1, x_2, the output of the generator network $G(x_1)$, and all the weakly supervised examples). In the first stage, our Sim-GAN model learns maps from X to X and Y to Y. After the above process, we have obtained the first stage objective:

$$G^{\text{Sim-GAN}_{\text{FGX}}} = \mathscr{L}_{\text{Gan}}(G(x), D_x, x_1, x_2) + \lambda_2 \mathscr{L}_{\text{Sim}}(x_2', x_1, x_2, x_W) + \lambda_1 \mathscr{L}_{\text{Cyc}}(G(x), F(x), x_1, x_2). \tag{4}$$

In this stage, our global objective function consists of three parts: the Gan losses, the Cyc losses, and the Sim losses. The parameter value we used is $\lambda_1 = 10$, $\lambda_2 = 60$. FGX and FGY mean that we use the first stage to reduce the number of adversarial examples for categories X and Y.

In the second stage, our Sim-GAN model learns maps from two categories X and Y. We introduce encoders E_x and E_y and category codes C_x and C_y for our model. Here, we define $G = G(x, E_x(x), C_x)$, $F = F(y, E_y(y), C_y)$ to indicate two generator networks G, F for two examples x, y of categories X, Y. In the previous papers, the SSIM (structural similarity) algorithm is used to compare the structural similarities of images. To reduce the effect of adversarial examples, because other loss calculation methods do not reflect the special structural similarity between different adversarial examples from different categories, therefore, we have improved the algorithm so that it can calculate the distance between two tensors. We apply s to represent the

structural similarity of the two tensors. The SSIM loss can be expressed as follows:

$$\mathscr{L}_{\mathrm{SSIM}}(G(x),F(y)) = E_{x\sim p_{\mathrm{data}}(x)}\left[\left\|G(x)-y_s\right\|\right] + E_{y\sim p_{\mathrm{data}}(y)}\left[\left\|F(y)-x_s\right\|\right]. \tag{5}$$

In this stage, we introduce variational autoencoders (VAEs) [13] type encoders E_x and E_y to get the local shared latent space. Our goal is to use the random Gaussian distribution ($N(0,\mathrm{I})$) to represent the local shared features. The VAE loss can be expressed as follows:

$$\mathscr{L}_{\mathrm{Vae}}(X,Y) = E_{y\sim p_{\mathrm{data}}(y)}\left[\log D_{\mathrm{KL}}\left(E_y(y)\parallel \mathscr{N}(0,I)\right)\right] + E_{x\sim p_{\mathrm{data}}(x)}\left[\log D_{\mathrm{KL}}\left(E_x(x)\parallel \mathscr{N}(0,I)\right)\right]. \tag{6}$$

where $D_{\mathrm{KL}}(p\parallel q) = -\int p(z)\log(p(z)/q(z))\mathrm{d}z$ (here, p and q are the latent distributions and z is the latent vector from VAE-like encoder. D_{KL} is the Kullback–Leibler divergence). To enforce the generator utilizing the latent vector z_x, z_y, the reconstruction latent vector loss is expressed as follows:

$$\mathscr{L}_{\mathrm{Latent}} = E_{y,z_y}\left[\left\|E_y(y')-z_y\right\|_1\right] + E_{x,z_x}\left[\left\|E_x(x')-z_x\right\|_1\right]. \tag{7}$$

Specifically, when y is input to E_y, we will get z_y. Then z_y can be input to the generator network $G(y,z_y,C_y)$. The reason for the paired z_y' and z_y is that $y' = G(y,z_y,C_y)$. After the above process, we have obtained the following objective function for the second stage:

$$G^{\mathrm{Sim-GAN_{TGX}}} = \mathscr{L}_{\mathrm{Gan}}(G,D_Y,X,Y) + \mathscr{L}_{\mathrm{Gan}}(F,D_X,Y,X) + \lambda_1\mathscr{L}_{\mathrm{Cyc}}(G,F) + \lambda_2\mathscr{L}_{\mathrm{Ssim}}(G,F) + \lambda_3\mathscr{L}_{\mathrm{Vae}}(X,Y) + \lambda_4\mathscr{L}_{\mathrm{Latent}}. \tag{8}$$

TGX means that we use the second stage to learn maps from category X to category Y. In this stage, our global objective function consists of five parts: the Gan losses, the Cyc losses, the SSIM losses, the VAE losses, and the reconstruction losses. The parameter value we used is $\lambda_1 = 10$, $\lambda_2 = 7, \lambda_3 = 0.01, \lambda_4 = 10$. Finally, the workflow of example x can be expressed as follows: $x \longrightarrow G_{\mathrm{stage1}}(x) \longrightarrow x' \longrightarrow G_{\mathrm{stage2}}(x') \longrightarrow \widehat{x} \longrightarrow F_{\mathrm{stage2}}(\widehat{x}) \approx x'$.

3.2. Network Architecture

3.2.1. Generator Network. The goal of the generator network is to generate learned features. For the first stage, we use the ResNet structure with an encoder-decoder framework, which contains two stride-2 convolution layers for downsampling, six residual blocks, and two stride-2 transposed convolution layers for upsampling. In order to get more local features, we use local response normalization [14] for all the convolutional layers. For the second stage, some details and spatial information may be lost in the downsampling process. We use the ResNet structure with a decoder framework, which contains two stride-2 convolution layers for downsampling, six residual blocks, and two stride-2 transposed convolution layers for upsampling. We replace all normalization layers except upsampling layers with CBIN (the central biasing instance normalization) layers [5]. The CBIN aims to adjust the different distributions of input feature maps adaptively with learnable parameters, which makes the category code able to manage the different tasks. We use the category code to label the different mapping in the generator.

3.2.2. Discriminator Network. For the first stage, we use one discriminator networks to make a distinction between the real example and the weakly supervised example. For the second stage, we use two discriminator networks to discriminate the real and fake images in different scales.

3.2.3. Encoder. Our encoders consists of three convolution layers followed by four residual blocks to down example the input examples. In order to get more features, we use instance normalization for all the convolutional layers. It should be noted that the output of the encoder will be used in our generator network.

4. Experiments

To explore the generality of the Sim-Gan model, we test the method on a variety of tasks including human faces to animes, human faces to cats, human faces to dogs, and cats to dogs. We carry on the experiment for unpaired image-to-image translation on four open source datasets. We implement the Sim-Gan model in the open source Tensorflow framework, which uses GTX1080Ti GPUs for both training and testing. We first optimize the dataset and deal with the problem that the dataset does not converge and then use the trained model to process the input data in the second stage and perform the image translation task. Furthermore, we used our model to handle the four tasks above. Then we performed experimental comparisons with the most advanced models to accomplish the same tasks. Finally, we recorded various performance indicators for testing [15–18].

4.1. Datasets and Preprocessing. Before starting the experiment, we should resize the image to 256×256. Each batch of training randomly loads 1 image from the source category and then randomly loads 1 image from the target category. We use a total of four public datasets for testing and comparison. The CELEBA dataset [19] with 202,599 celebrity face images (short for faces). The Getchu dataset [19] contains 26,752 anime character face images with a clean background (short for animes). The Flickr-Dog dataset (http://www.recod.ic.unicamp.br/~rwerneck/datasets/flickr-dog/) has 42 classes and 374 photos (short for dogs). The cAT dataset [20] (short for cats) includes 10,000 cat images. For each image, they annotate the head of a cat with nine points, two for eyes, one for the mouth, and six for ears.

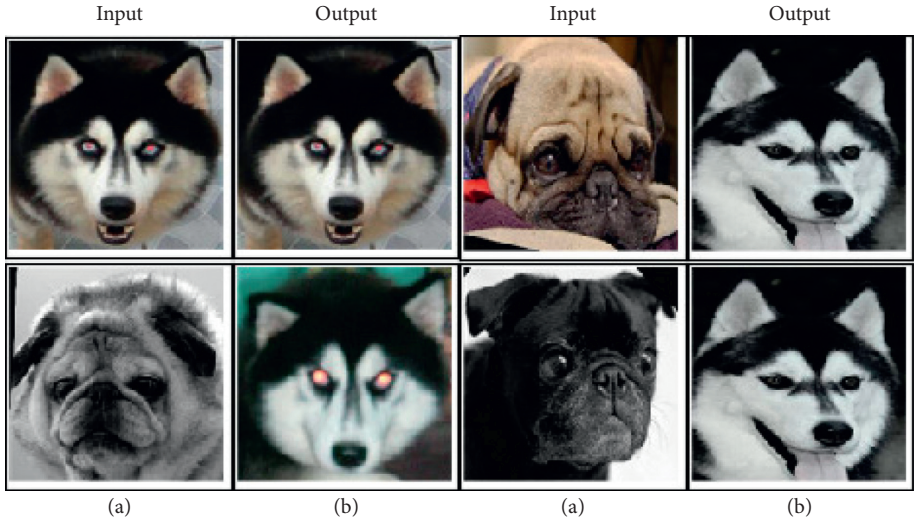

Figure 2: Some experimental results of the first stage by our model, which takes dogs to dogs. In each column from left to right: (a) input image (256×256 pixels) and (b) the output of the first stage.

We conduct cats to dogs, human faces to cats, human faces to dogs, and human faces to animes task separately. The experimental results of first stage on dogs to dogs tasks are shown in Figure 2.

As shown in Figure 2, we note that Sim-GAN can generate dogs that are close to the weakly supervised dogs. In this way, we can automatically reduce the number of adversarial dogs. The experimental results of the second stage on these tasks are shown in Figure 3:

In Figure 3, we find that Sim-GAN can generate objects closer to the target objects for four tasks. It means that Sim-GAN can get the local shared latent space to stop the model from collapse. In the first stage, we reduce more than 10% adversarial examples for dogs, 4% adversarial examples for cats, 3% adversarial examples for animes, and 2% adversarial examples for human faces.

4.2. Baselines. To compare the performance of our Sim-GAN model, we adopt the CycleGAN [1] model, the ganimorph [12] model (Imo-GAN for short), and the MUNIT [2] model as our baseline models.

4.3. Evaluation Index. Using the same evaluation metrics, we compare our method against several baselines qualitatively and quantitatively.

4.3.1. AMT. For these tasks, we run "real vs fake" perceptual studies on Amazon Mechanical Turk (AMT) to assess the realism of our outputs. We follow the same perceptual study protocol from Isola et al. [10], and we gather data from 50 participants per algorithm we tested. Participants were shown a sequence of pairs of images, one a real image and one fake (generated by our algorithm or a baseline), and asked to click on the image they thought was real.

4.3.2. Classification (Cf for Short). We train three Xception [21] based binary classifiers for each image datasets. The baseline is the classification accuracy in real images. Higher classification accuracy means that the generated images may more easy to distinguish.

4.3.3. Consistency (Cs for Short). We compared the domain consistency between real images and generated images by computing average distance in feature space. We use the cosine similarity to evaluate the perceptual distance in the feature space of the VGG-16 network [22] pretrained in ImageNet [23]. We sum across the five convolution layers preceding the pool layers. The larger the value, the more the similarities between the two images. In the test stage, we randomly example the real image and the generated image from the same domain to make up the data pair. Then we compute the average distance between each pair.

4.4. Base Model Comparison. Here, we evaluate the performance of different models. In order to be fair, we use the same dataset to ensure that each model reaches a convergence state. The experimental results of the Sim-GAN model on these tasks are shown in Figure 4 (the result ours$_2$ indicates that only the model of our second stage is used to generate data).

In Figure 4, it is shown that our model generates better and closer target category examples than other models. The CycleGAN model cannot handle these tasks; it only learns part of the style mapping. The Imo-GAN model enables completing a few tasks but lacks some details; it only learns most of the content mapping. The MUNIT model enables completing most tasks, it learns the right content mapping and style mapping. The experimental results show that for the cats to human faces task, besides CycleGAN, all the models produce natural results and it means that the local shared latent space is close to some of the global shared latent space. This is because the cats and faces datasets

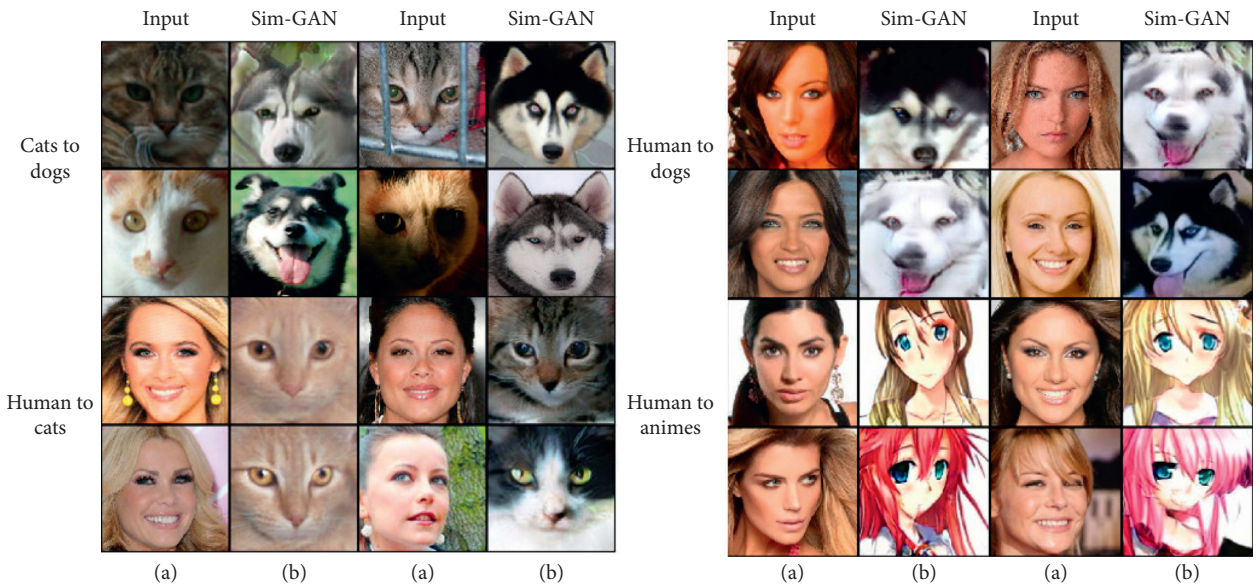

Figure 3: Some image-to-image translation results by our model, which takes cats to dogs, human faces to cats, human faces to dogs, and human faces to animes. In each column from left to right: (a) input image (256×256 pixels) and (b) image-to-image translation result.

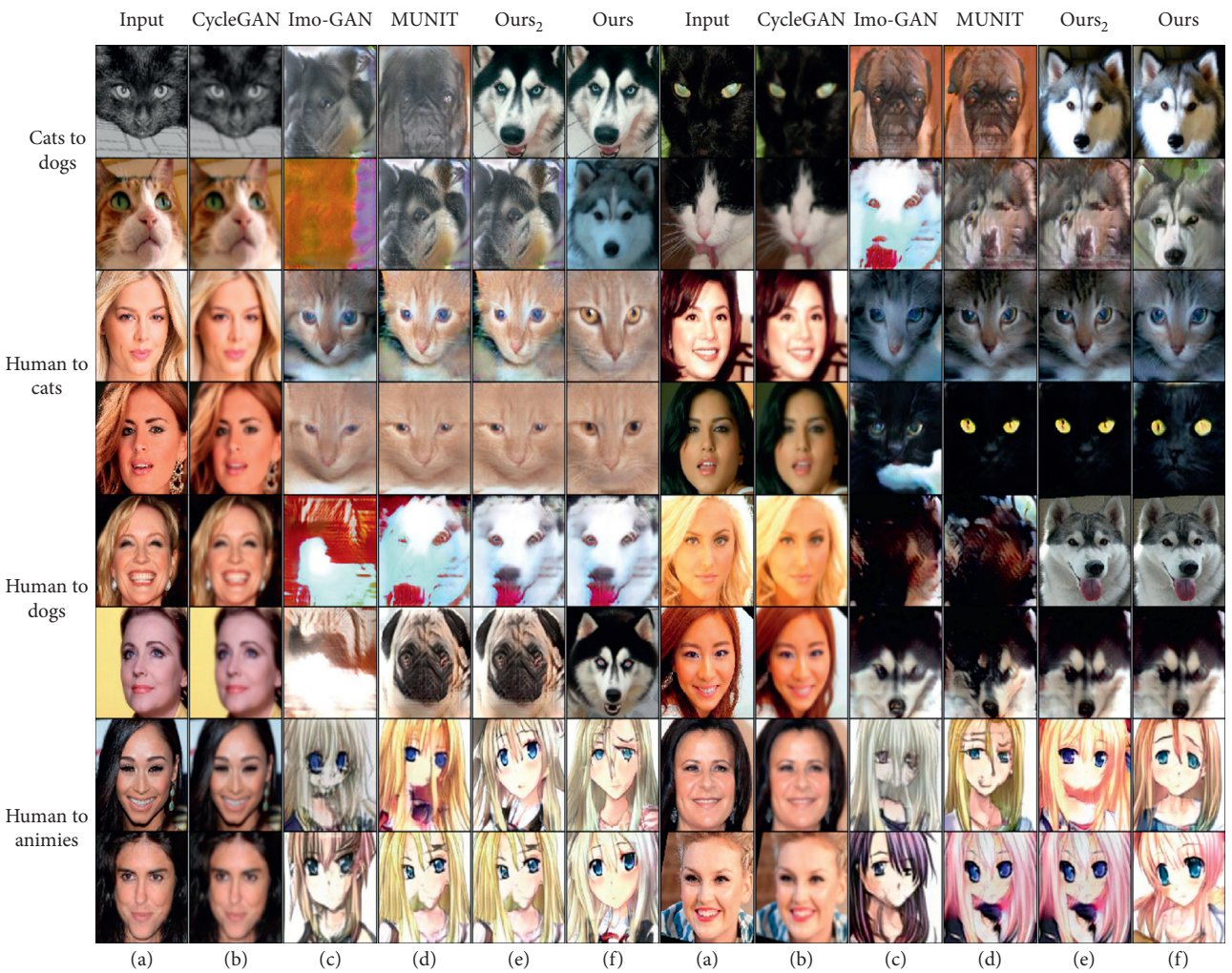

Figure 4: The image-to-image translation results for cats to dogs, human faces to cats, human faces to dogs, and human faces to animes tasks. (a) Input images. (b) Translation results by CycleGAN. (c) Translation results by Imo-GAN. (d) Translation by MUNIT. (e) Translation results only using the model of our second stage. (f) Translation results by our method.

Table 1: Quantitative evaluations in terms of cats to dogs (A), human faces to cats (B), human faces to dogs (C), and human faces to animes (D) task from different models.

Tasks	Baselines	AMT (%)	Cf	Cs
A	CycleGAN	2% ± 1.0	0.15	1.46
	Imo-GAN	26% ± 1.0	0.76	1.27
	MUNIT	23% ± 1.0	0.71	1.28
	$Ours_2$	24% ± 1.0	0.73	1.20
	Ours	**39% ± 1.0**	**0.92**	**1.06**
B	CycleGAN	2% ± 1.0	0.15	1.48
	Imo-GAN	23% ± 1.0	0.72	1.36
	MUNIT	25% ± 1.0	0.76	1.34
	$Ours_2$	26% ± 1.0	0.78	1.29
	Ours	**28% ± 1.0**	**0.82**	**1.26**
C	CycleGAN	2% ± 1.0	0.11	1.62
	Imo-GAN	20% ± 1.0	0.46	1.43
	MUNIT	23% ± 1.0	0.61	1.40
	$Ours_2$	24% ± 1.0	0.66	1.38
	Ours	**26% ± 1.0**	**0.79**	**1.24**
D	CycleGAN	2% ± 1.0	0.11	1.63
	Imo-GAN	20% ± 1.0	0.46	1.43
	MUNIT	23% ± 1.0	0.61	1.40
	$Ours_2$	25% ± 1.0	0.67	1.37
	Ours	**26% ± 1.0**	**0.78**	**1.26**

contain fewer adversarial examples. For the cats to dogs task, besides Sim-GAN, all the models produce unnatural results and it means that the local shared latent space enables stopping the model from collapsing. This is because the dogs dataset contains many adversarial examples. In addition, the image translation results of $ours_2$ model shows that if there is no weakly supervised training in the first stage, $ours_2$ model also produces natural results. Under most conditions, the image translation results of $ours_2$ model are better than the MUNIT model and get the local shared latent space. Furthermore, the reason for some similar image translation results between MUNIT and our method is that the local shared latent space and the global shared latent space may intersect under certain conditions.

The contrast effect for four tasks on three evaluation indicators is shown in Table 1.

As can be seen from Table 1, our model achieved leading numerical results than other models. This means that we not only reduce the number of adversarial examples but also successfully capture the local shared latent space for unsupervised image-to-image translation.

4.5. Limitations. Although our model is able to generate semantically plausible and visually pleasing examples for wild datasets with many adversarial examples, it has some limitations. The first limitation is that we are not able to translate desired results based on conditions. These will be addressed in the next study. Our model can avoid collapse for the wild dataset, but the weakly supervised model reduces the number of examples. Therefore, our image translation results lack diversity, which will be discussed in future work. The second limitation is that the image translation results are not similar in the pose of the head. The main reason for the not similarity in the pose of the head is the dataset. The similarity is global latent spaces. The not similarity in the pose of the head is the local latent spaces. The last limitation is the pretrained ImageNet for the consistency evaluation. Though we use pretrained ImageNet for the consistency evaluation of dogs, cats, and animes, the VGG-Face model is very critical for face consistency evaluation. We will use it for the consistency evaluation of cats to human faces, dogs to human faces, and animes to human faces tasks.

5. Conclusion

This paper studies the use of a novel Generative Adversarial Networks model for image-to-image translation when other models collapse. We assume the shared latent space can be classified as global and local and design a weakly supervised Similar GANs (Sim-GAN for short) to capture the local shared latent space rather than the global shared latent space. We first introduce a loss based on SSIM (structural similarity) distance with weakly supervised examples for Sim-GAN to automatically reduce the number of adversarial examples within each category. Then we introduce the category codes to constraint the local shared features across categories and the encoders to capture the local shared latent space for unsupervised image-to-image translation. Experiments on four public datasets show that our model significantly outperforms state-of-the-art baseline methods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the BUPT Excellent PhD Students Foundation (no. CX2018206).

References

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[2] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-toimage translation," 2018, https://arxiv.org/abs/1804.04732.

[3] F. Schroff, D. Kalenichenko, and P. James, "Facenet: a unified embedding for face recognition and clustering," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, Boston, MA, USA, June 2015.

[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[5] X. Yu, Z. Ying, Li Ge, and W. Gao, "Multi-mapping image-to-image translation with central biasing normalization," 2018, https://arxiv.org/pdf/1806.10050.pdf.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 2672–2680, Montreal, Canada, December 2014.

[7] A. Martin, S. Chintala, and L. Bottou, "Wasserstein gan," 2017, https://arxiv.org/abs/1701.07875.

[8] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821, IEEE, Venice, Italy, October 2017.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, https://arxiv.org/abs/1511.06434.

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976, IEEE, Honolulu, HI, USA, July 2017.

[11] X. Liang, H. Zhang, and E. P. Xing, "Generative semantic manipulation with contrasting gan," 2017, https://arxiv.org/abs/1708.00315.

[12] A. Gokaslan, V. Ramanujan, D. Ritchie, K. In Kim, and T. James, "Improving shape deformation in unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, https://arxiv.org/abs/1312.6114.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, Springer, Berlin, Germany, 2012.

[15] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning*, pp. 448–456, Lille, France, July 2015.

[16] D. Kingma and J. Lei Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/pdf/1412.6980.pdf.

[17] Y. Xu, J. Ren, G. Wang, C. Zhang, J. Yang, and Y. Zhang, "A blockchain-based non-repudiation network computing service scheme for industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3632–3641, 2019.

[18] X. Yan, B. Cui, Y. Xu, P. Shi, and Z. Wang, "A method of information protection for collaborative deep learning under gan model attack," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, p. 1, 2019.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, Santiago, Chile, December 2015.

[20] W. Zhang, J. Sun, and X. Tang, "Cat head detection-how to effectively exploit shape and texture features," in *Proceedings of the European Conference on Computer Vision*, pp. 802–816, Springer, Marseille, France, October 2008.

[21] F. Chollet, "Xception: deep learning with depthwise separable convolutions," 2017, https://arxiv.org/abs/1610.02357.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[23] F.-F. Li, "Imagenet: crowdsourcing, benchmarking & other cool things," *CMU VASC Seminar*, vol. 16, pp. 18–25, 2010.