Complex & Intelligent Systems (2023) 9:4111–4122
https://doi.org/10.1007/s40747-022-00924-1

ORIGINAL ARTICLE

Multi-feature contrastive learning for unpaired image-to-image translation

Yao Gou[1] · Min Li[1] · Yu Song[1] · Yujie He[1] · Litao Wang[1]

Received: 20 January 2022 / Accepted: 6 November 2022 / Published online: 26 December 2022

Abstract
Unpaired image-to-image translation for the generation field has made much progress recently. However, these methods suffer from mode collapse because of the overfitting of the discriminator. To this end, we propose a straightforward method to construct a contrastive loss using the feature information of the discriminator output layer, which is named multi-feature contrastive learning (MCL). Our proposed method enhances the performance of the discriminator and solves the problem of model collapse by further leveraging contrastive learning. We perform extensive experiments on several open challenge datasets. Our method achieves state-of-the-art results compared with current methods. Finally, a series of ablation studies proved that our approach has better stability. In addition, our proposed method is also practical for single image translation tasks. Code is available at https://github.com/gouayao/MCL.

Keywords Generative model · Image translation · Contrastive learning · Multi-feature

Arabic Keywords إنشاء النموذج · ترجمة الصور · التقابلي التعلم · متعدد الميزات

Introduction

Generative adversarial networks (GANs) [14] usually include two models: a generator and a discriminator. The generator aims to capture the real data distribution to generate new samples. The discriminator aims to judge an input sample's realness to identify whether it is real or fake. Because of their solid generative capability, GANs have become one of the most promising methods in the family of generative models [13]. It is widely applied in various sectors [9,37], especially in the field of image generation.

Many problems can be summarized as image-to-image translation tasks in the image generation field, such as image denoising [5], dehazing [3,28], coloring [46], makeup [29], and super-resolution [26,34,40]. The image-to-images translation aims to find a mapping between a source domain $\mathcal{X}$ and a target domain $\mathcal{Y}$ and "translate" the input image into the corresponding output image. In general, image-to-image translation tasks can be categorized into two groups: paired (supervised) [20,35,39] and unpaired (unsupervised) [19,24,27,30,43,47]. Pix2pix [20] investigated conditional GANs (cGANs) as a general-purpose solution to image-to-image translation problems and developed a common framework for all these problems. Wang et al. [39] and Park et al. [35] extended pix2pix and further improved the quality of the generated images. These approaches require paired data for training. However, for many tasks, paired training data are challenging to obtain. It significantly limits the application of image-to-image translation. To address this problem, Zhu et al. [47] presented a Cycle-consistency GAN (CycleGAN) for learning an inverse mapping between two domains $\mathcal{X}$ and $\mathcal{Y}$ to realize image-to-image translation tasks in the absence of paired examples. Similarly, literature [24,43] also used cycle-consistency to realize unpaired image-to-image translation.

Although cycle-consistency does not require the training data to be paired, it assumes that the relationship between the two domains $\mathcal{X}$ and $\mathcal{Y}$ is a bijection, which is often too restrictive. More recently, some methods [1,2,11,31,36] have attempted to use one-sided mapping instead of two-sided

This paper is supported by National Natural Science Foundation of China (62006240).

✉ Yao Gou
gouayao@163.com

✉ Min Li
proflimin@163.com

[1] Xi'an High-Tech Research Institute, Xi'an 710025, China

mapping. In literature [36], Park et al. first applied contrastive learning to image-to-image translation tasks by learning the correspondence between input and output patches, achieving a performance superior to those based on cycle-consistency. This method is named CUT. To further leverage contrastive learning and avoid the drawbacks of cycle-consistency, Han et al. [16] proposed a dual contrastive learning approach to infer an efficient mapping between unpaired data, referred to as Dual Contrastive Learning GAN (DCLGAN). Both CUT and DCLGAN only introduce contrastive learning into the generator, making the discriminator prone to overfitting and even suffering mode collapse during training.

Previous approaches either had strict restrictions on training datasets (paired) or mapping functions (bijective), or merely considered enhancing the performance of the generator. In this paper, we propose a multi-feature contrastive learning method. Our method is a one-sided mapping method for unpaired image-to-image translation, considering enhancing the performance of the generator and discriminator. In summary, this work aims to make two contributions:

(1) Our proposed method can further enhance the performance of the discriminator, prevent the discriminator overfitting issue during training. This method benefits from multi-feature contrastive learning, which is called MCL. Large amounts of experiments show that the quantitative and qualitative aspects of our method are better than those of other methods on various unpaired translation tasks. In addition, our proposed method is also applicable to single image translation tasks, as shown in Fig. 1.
(2) We analyze the feature information of the discriminator output layer and construct a contrastive loss using this feature information. Our proposed loss is simple, effective, and universally applicable, called MCL loss. Experiments show that MCL loss can be directly added to most image-to-image translation methods (such as CycleGAN, CUT, and DCLGAN) to improve the quality of the generated images. In addition, since we did not utilize additional model parameters, MCL loss adds little additional training time and computational resources.

Related work

Unpaired translation

GANs [14] usually consist of two models: (a) a generator $G : Z \to X$, (b) and a discriminator $D : X \to [0, 1]$. The generator G maps a potential variable $z \sim p(z)$ to X to generate a sample $G(z)$ of realness, where $p(z)$ represents a specific prior distribution. The discriminator D maps the input sample to a probability space to distinguish between the real and the generated sample. The training process of G and D follows the following objective function:

$$\begin{aligned}\min_G \max_D V(G, D) &= E_{x\sim p_{\text{data}}}\left[\log D(x)\right] \\ &\quad + E_{z\sim p_z}\left[\log\left(1 - D(G(z))\right)\right] \\ &= E_{x\sim p_{\text{data}}}\left[\log D(x)\right] \\ &\quad + E_{x\sim p_g}\left[\log\left(1 - D(x)\right)\right] \end{aligned} \tag{1}$$

where p_{data}, p_z, and p_g represent the data distribution of real samples, input potential variables, and generated samples, respectively.

For unpaired image-to-image translation tasks [16,36,47], an unpaired dataset is given: $X = \{x \in \mathcal{X}\}$ and $Y = \{y \in \mathcal{Y}\}$. On the one hand, the generator G wants to learn a mapping $G : X \to Y$ from the source domain $\mathcal{X}$ to the target domain $\mathcal{Y}$. On the other hand, the discriminator D hopes to distin-

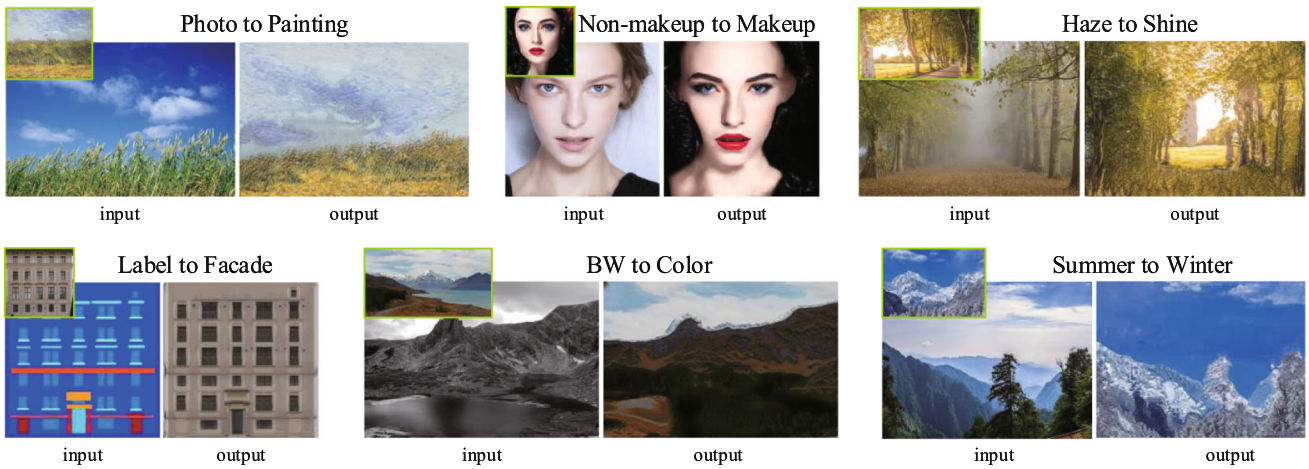

Fig. 1 Single image translation tasks. We try to solve several different issues with the same architecture and objective from a single image. Here the results of our approach are shown on these issues, including style transfer, makeup, dehazing, label2facade, coloring, and summer2winter

guish the transformed image $G(x)$ from the target domain image $\mathcal{Y}$. At this point, the objective function of training G and D is as follows:

$$\min_G \max_D V(G, D) = E_{y\sim Y}\left[\log D(y)\right] + E_{x\sim X}\left[\log\left(1 - D(G(x))\right)\right] \quad (2)$$

Contrastive learning

Contrastive learning was found to be effective in state-of-the-art unsupervised visual representation learning tasks [6,17,21,38,42]. It aims to learn a mapping function that makes representations of associated samples closer and keeps representations of other samples away. These associated samples are named positive samples, and others are named negative. For contrastive learning, how to properly construct positive and negative samples is crucial.

Some recent works investigate the use of contrastive learning for image translation [1,16,31,36]. TUNIT [1] adopts contrastive losses to simultaneously separate image domains and translates input images into the estimated domains. DivCo framework [31] uses contrastive losses to properly constrain both "positive" and "negative" relations between the generated images specified in the latent space. CUT [36] uses a noise contrastive estimation framework to maximize the mutual information between input and output for improving the performance of unpaired image-to-image translation. DCLGAN [16] extends one-sided mapping to two-sided mapping to further leverage contrastive learning, performing better in learning embeddings and thus achieving state-of-the-art results.

Note that all the above methods only introduce contrastive learning into the generator, that leads to the discriminator overfitting issue in the training process. Our proposed MCL is a novel contrastive learning strategy, which uses the feature information of the discriminator output layer to construct the contrastive loss. We further demonstrate the superiority of our method compared to several state-of-the-art methods through extensive experiments. Our method only uses existing feature information, so almost no additional computing resources and training time will be added. The specific method is described below.

Methods

Given a dataset of $X = \{x \in \mathcal{X}\}$ and $Y = \{y \in \mathcal{Y}\}$, we aim to learn a mapping that translates an image x from a source domain $\mathcal{X}$ to a target domain $\mathcal{Y}$. For a 70×70 PatchGAN discriminator [20], its output layer is a 30×30 matrix $A = (a_{i,j})_{30\times 30}$-each element $a_{i,j}$ aims to classify whether 70×70 overlapping image patches are real or fake. The discriminator determines whether an input image is real or fake by the expectation of all elements.

Different from previous methods [11,16,20,36,47], we also consider how to use the feature information of the discriminator output layer to construct the contrastive loss and

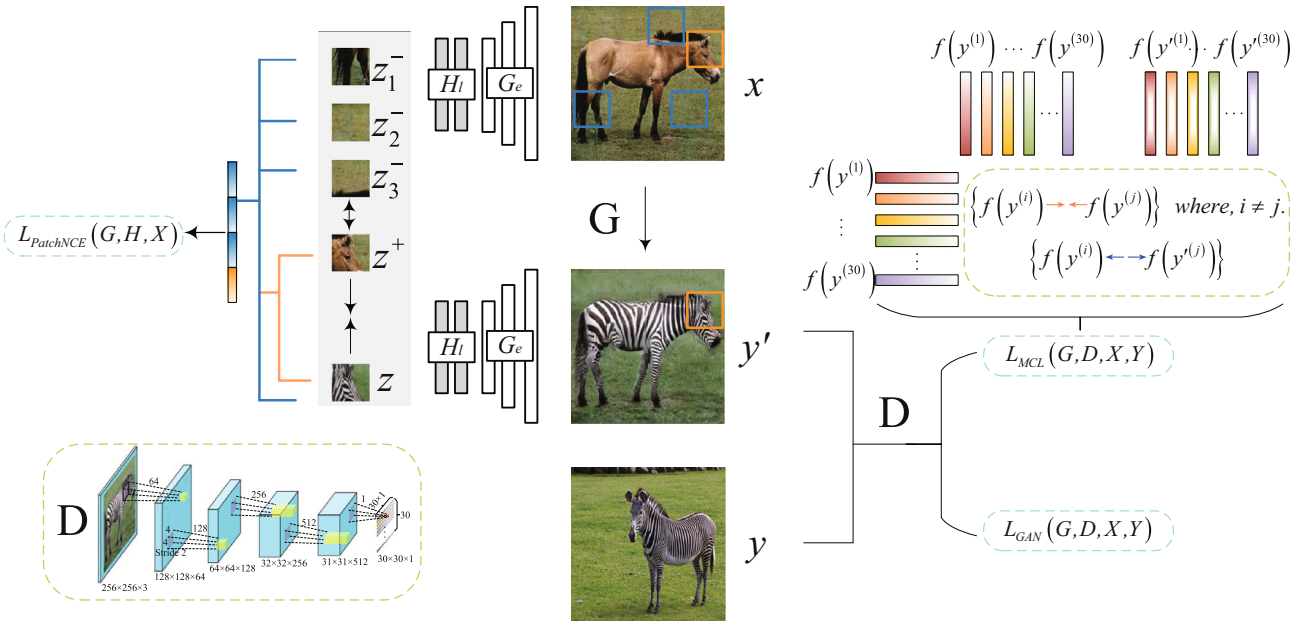

Fig. 2 Overall architecture of our approach. We consider constraining the generator and discriminator based on contrastive learning. Our approach includes four loss items altogether: adversarial loss, two PatchNCE losses, and MCL loss. Adversarial loss is encouraged to control the translation style. PatchNCE loss and MCL loss are used to enhance the performance of the generator and discriminator, respectively. We omit a similar PatchNCE loss $L_{\text{PatchNCE}}(G, H, Y)$ here

thus enhance the generalization performance of the discriminator. Figure 2 shows the overall architecture of our approach. We combine four losses, including adversarial loss, two PatchNCE losses, and MCL loss. The details of our objective are described below.

Adversarial loss

We use an adversarial loss [14] to encourage the translated images to be visually similar enough to images from the target domain, as described below:

$$L_{\mathrm{GAN}}(G, D, X, Y) = E_{y\sim Y}\left[\log D(y)\right] + E_{x\sim X}\left[\log\left(1 - D(G(x))\right)\right] \quad (3)$$

PatchNCE loss

We use a noise contrastive estimation framework [38] to maximize the mutual information between the input and output patches. That is, a generated output patch should appear closer to its corresponding input patch and keep away from other random patches.

Following CUT [36], a query, a positive and N negatives are, respectively, mapped to K-dimensional vectors, which are defined as v, $v^+ \in R^K$, and $v^- \in R^{N\times K}$. Note that $v_n^- \in R^K$ represents the n-th negative. In this paper, query, positive and negative refer to output, corresponding input, and noncorresponding input, respectively. Our goal is to associate positive and stay away from negatives, which can be expressed mathematically as a cross-entropy loss [15]:

$$l\left(v, v^+, v^-\right) = -\log\left[\frac{\exp\left(v\cdot v^+/\tau\right)}{\exp\left(v\cdot v^+/\tau\right) + \sum_{n=1}^{N}\exp\left(v\cdot v_n^-/\tau\right)}\right] \quad (4)$$

We normalize vectors onto a unit sphere to prevent the space from collapsing or expanding. We use a temperature parameter $\tau = 0.07$ as default.

Like CUT [36], the generator is divided into two components: an encoder G_e and a decoder G_d, applied sequentially to produce the output image $y' = G(x) = G_d(G_e(x))$. We select L layers from $G_e(x)$ and send it to a small two-layer MLP network H_l, producing a stack of features $\{z_l\}_L = \left\{H_l\left(G_e^l(x)\right)\right\}_L$, where $G_e^l(x)$ represents the output of the lth chosen layer. Then, we index into layers $l \in \{1, 2, \ldots, L\}$ and denote $s \in \{1, \ldots, S_l\}$, where S_l is the number of spatial locations in each layer. We refer to the corresponding feature("positive") as $z_l^s \in R^{C_l}$ and the other features("negatives") as $z_l^{S\backslash s} \in R^{(S_l-1)C_l}$, where C_l is the number of channels at each layer. Similarly, we encode the output image y' into $\left\{\hat{z}_l\right\}_L = \left\{H_l\left(G_e^l(G(x))\right)\right\}_L$. We aim to match corresponding input-output patches at a specific location. In Fig. 2, for example, the head of the output zebra should be more strongly associated with the head of the input horse than the others, such as legs and grass. Thus, the PatchNCE loss can be expressed as

$$L_{\mathrm{PatchNCE}}(G, H, X) = E_{x\sim X}\sum_{l=1}^{L}\sum_{s=1}^{S_l} l\left(\hat{z}_l^s, z_l^s, z_l^{S\backslash s}\right) \quad (5)$$

In addition, $L_{\mathrm{PatchNCE}}(G, H, Y)$ is computed on images from the domain $\mathcal{Y}$ to prevent the generator from making unnecessary changes.

MCL loss

PatchNCE loss enhances the performance of the generator by learning the correspondence between input and output image patches. We further improve the performance of the discriminator using the feature information of the discriminator output layer, which is named MCL loss.

Generally, the discriminator estimates the realness of an input sample using a single scalar. However, this simple mapping undoubtedly misses some important feature information. Therefore, it is easy to overfit because the discriminator is not strong enough. To make full use of the feature information of the discriminator output layer, we use it to construct a contrastive loss instead of simply mapping it to a probability space. We treat the feature information of the discriminator output layer into a $n\times n$ matrix $A = \left(a_{i,j}\right)_{n\times n}$. Then, we process each row of elements of the matrix as a feature vector, that is $A = \left(\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(n)}\right)^T$, where $\alpha^{(i)} = \left(a_{i,1}, a_{i,2}, \ldots, a_{i,n}\right)$. And we normalize each feature vector to obtain $f(A) = \left(f\left(\alpha^{(1)}\right), f\left(\alpha^{(2)}\right), \ldots, f\left(\alpha^{(n)}\right)\right)^T$. Next, we construct the MCL loss by studying the relationship between different feature vectors.

As shown in Fig. 2, for an output image $y' = G(x)$ and an image y from the target domain $\mathcal{Y}$, we have $f(A_{(y')}) = (f(y'^{(1)}), f(y'^{(2)}), \ldots, f(y'^{(n)}))^T$ and $f(A_{(y)}) = (f(y^{(1)}), f(y^{(2)}), \ldots, f(y^{(n)}))^T$ by the discriminator (here, n=30). Naturally, we want any feature vector $f(y^{(i)})$ of y to be as close as possible to others of y and far away from the feature vectors of y'. We let $r = \{r^{(i)}\} = \{f(y^{(i)})\}$, $f = \{f^{(i)}\} = \{f(y'^{(i)})\}$, and $r^{(-i)} = r\backslash r^{(i)}$. Formally, the contrastive loss is defined by

$$L_{con}\left(r^{(i)}, f, r^{(-i)}\right) = -\frac{1}{\left|r^{(-i)}\right|}\sum_{r^{(j)}\in r^{(-i)}}\log \times\frac{\exp\left(r^{(i)}\cdot r^{(j)}/\omega\right)}{\sum_{r^{(k)}\in r^{(-i)}}\exp\left(r^{(i)}\cdot r^{(k)}/\omega\right) + \sum_{f^{(k)}\in f}\exp\left(r^{(i)}\cdot f^{(k)}/\omega\right)} \quad (6)$$

where $\omega = 0.1$.

According to Eq. 6, the MCL loss of the discriminator is defined as follows:

$$L_{MCL}(G, D, X, Y) = \frac{1}{n}\sum_{i=i}^{n} L_{con}\left(r^{(i)}, f, r^{(-i)}\right) \tag{7}$$

Final objective loss

Our final objective loss includes adversarial loss, two PatchNCE losses, and MCL loss, as follows:

$$\begin{aligned} L = {} & L_{\text{GAN}}(G, D, X, Y) + \lambda_X \cdot L_{\text{PatchNCE}}(G, H, X) \\ & + \lambda_Y \cdot L_{\text{PatchNCE}}(G, H, Y) + \lambda_M \cdot L_{MCL}(G, D, X, Y) \end{aligned} \tag{8}$$

If not specified, we choose $\lambda_X = \lambda_Y = 1$ and $\lambda_M = 0.01$.

Compared with the existing methods, MCL achieves state-of-the-art results. In addition, to further reduce the model training parameters and improve the training speed, we also propose a lighter and faster version, named FastMCL. In FastMCL, we no longer consider the effect of $L_{\text{PatchNCE}}(G, H, Y)$ on the training process, that is to make $\lambda_Y = 0$. Surprisingly, even so, FastMCL achieves slightly worse performance compared to CUT [36]. All experimental results are shown in Sect. 3.3.

Experiments

We evaluated the performance of different methods on several datasets. And we introduced the training details, datasets, and evaluation protocols of the experiments in turn. Extensive experiments were performed on unpaired image translation tasks. Furthermore, our proposed method was extended to single image translation tasks. Finally, we performed an ablation study and analyzed the influence of different loss terms on the experimental results. All the experimental results prove that our proposed method is superior to existing methods.

Training details

In this paper, we mainly follow the setup of CUT [36] for training. Our full model MCL is trained up to 400 epochs, while the fast variant FastMCL is trained up to 200 epochs. Both MCL and FastMCL include a ResNet-based generator with 9 residual blocks [22] and a PatchGAN discriminator [20]. We choose the LSGAN loss [33] as an adversarial loss and train models at 256×256 resolution. The learning rate is set to 0.0002 and starts to decay linearly after half of the total epochs.

For a single image translation task, we adopt StyleGAN2-based architecture [23] for training, named SinMCL. The generator of SinMCL consists of 1 downsampling block of StyleGAN2 discriminator, 6 StyleGAN2 residual blocks, and 1 StyleGAN2 upsampling block. The discriminator of SinMCL has the same architecture as StyleGAN2. Since we do not use style code, the style modulation layer of StyleGAN2 was removed. Note that the coefficient of MCL loss λ_M is set to 0.03.

Datasets

Horse→Zebra contains 2401 training and 260 test images, all collected from ImageNet [10]. It was introduced in CycleGAN [47].

Cat→Dog contains 5000 training images and 500 test images for each domain from the AFHQ dataset [7].

CityScapes [8] contains 2975 training and 500 test images for each domain, a city label dataset.

Monet→Photo [36] contains only a high-resolution image in each domain, which is used for single image translation.

Van Gogh→Photo contains only a high-resolution image in each domain, which is also used for single image translation.

Evaluation protocol

Fréchet Inception Distance (FID) [18] is an evaluation metric mainly used in this paper. FID was proposed by Heusel et al. and is used to measure the distance between two data distributions. That is, a lower FID indicates better results. For cityscapes, we leverage its corresponding labels to calculate the semantic segmentation scores. We use a pre-trained FCN-8s model [20,32] and score three metrics including pixel-wise accuracy (pixAcc), average class accuracy (classAcc), and mean class Intersection over Union (IoU). In addition, we compare the model parameters and training times of different methods.

We compare our proposed method with current state-of-the-art unpaired image translation methods, including CycleGAN [47], GcGAN [11], FastCUT [36], CUT [36], SimDCL [16], and DCLGAN [16]. All the experimental results show that the quality of the images generated by our method is superior to others. Moreover, our method can produce better results with a lighter computational cost of training.

Unpaired image translation

Table 1 shows the evaluation results of our proposed method and all baselines on Horse→Zebra, Cat→Dog, and CityScapes datasets, and their visual effects are shown in

Table 1 Comparison with all baselines

Method	Horse→Zebra			Cat→Dog		CityScapes		
	sec/iter↓	Model parameters↓	FID	FID	FID	pixAcc↓	classAcc↓	IoU↓
CycleGAN [47]	0.40	28.286M	77.2	85.9	76.3	0.52	0.17	0.11
GcGAN [11]	0.26	16.908M	86.7	96.6	105.2	0.55	0.20	0.13
FastCUT [36]	**0.15**	**14.703M**	73.4	94.0	68.8	0.65	0.21	0.15
CUT [36]	0.24	**14.703M**	45.5	76.2	56.4	0.70	0.24	0.17
SimDCL [16]	0.47	28.852M	47.1	65.5	51.3	0.69	0.21	0.15
DCLGAN [16]	0.41	28.812M	43.2	**60.7**	49.4	0.74	0.22	0.17
FastMCL(ous)	0.15	**14.703M**	46.5	88.8	55.3	0.76	0.25	0.19
MCL(ous)	0.25	**14.703M**	**40.7**	70.2	**47.3**	**0.78**	**0.26**	**0.21**

Best values are in bold

We compared our approach on several open datasets, primarily using FID [18] evaluation metric. For CityScapes, we leverage its corresponding labels to show the semantic segmentation scores (pixAcc, classAcc, IoU). MCL produces state-of-the-art results, and takes equal or slightly worse resources than CUT [36] in model parameters and training speed (seconds per sample). Our variant FastMCL also produced desirable results

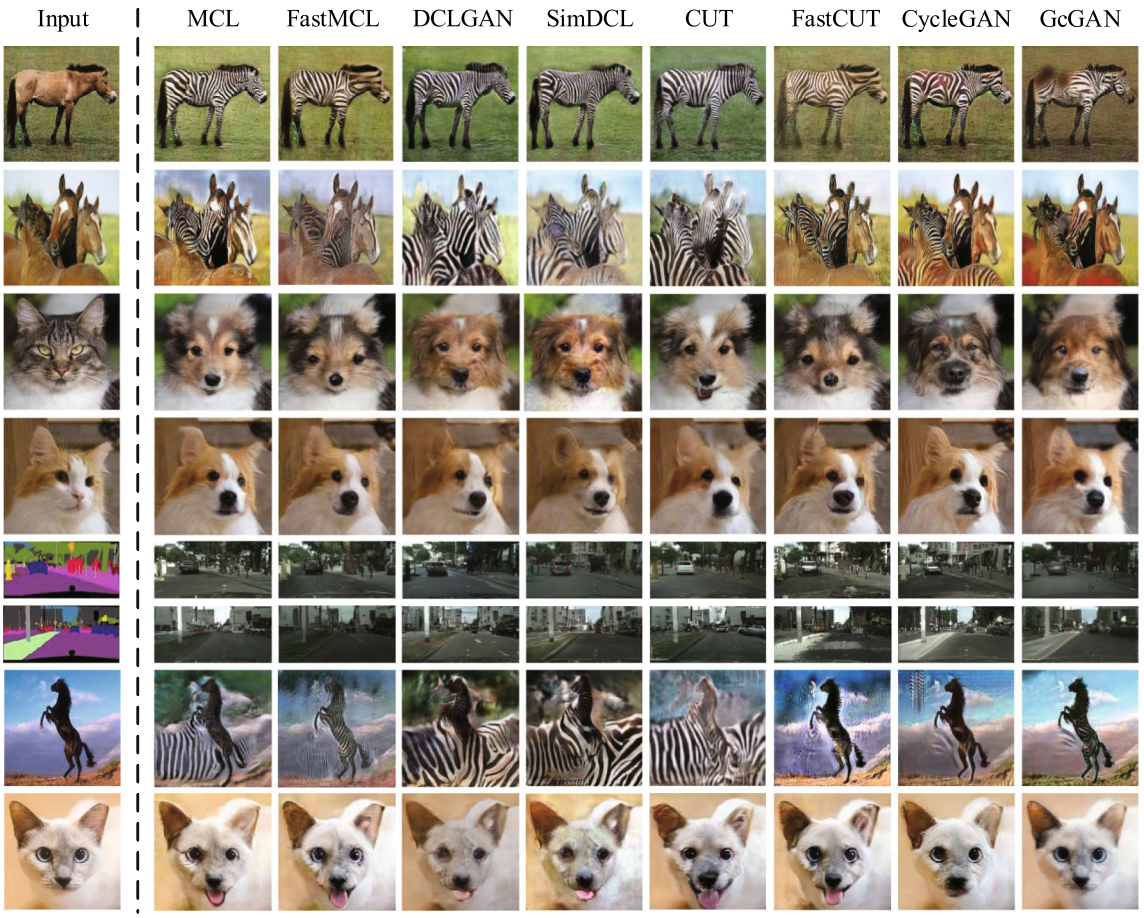

Fig. 3 Visual results of different methods. CycleGAN [47] and GcGAN [11] are cycle-consistency methods. CUT [36], FastCUT [36], DCLGAN [16], and SimDCL [16] introduce contrastive learning into the generator. Our versions MCL and FastMCL further leverage contrastive learning to enhance the performance of the discriminator. The last two rows show failing cases of other methods, and our method yielded relatively satisfactory results

Fig. 3. It is clear that our algorithms perform superior to all the baselines. As shown in Table 1, our MCL version produces state-of-the-art results, and takes equal or slightly worse resources than CUT [36] in model parameters and training speed (seconds per sample). Our variant FastMCL also produced desirable results. The last two rows of Fig. 3

Fig. 4 Example results of our MCL compared to DCLGAN [16] and CUT [36] on the cityscapes dataset with 256 × 256 resolution. The left column represents the ground truth and label, and the right three columns represent generated images and semantic labels by different methods. MCL achieves generated images more like the ground truth, and the semantic labels obtained through the pre-trained FCN-8s model are more like the real labels

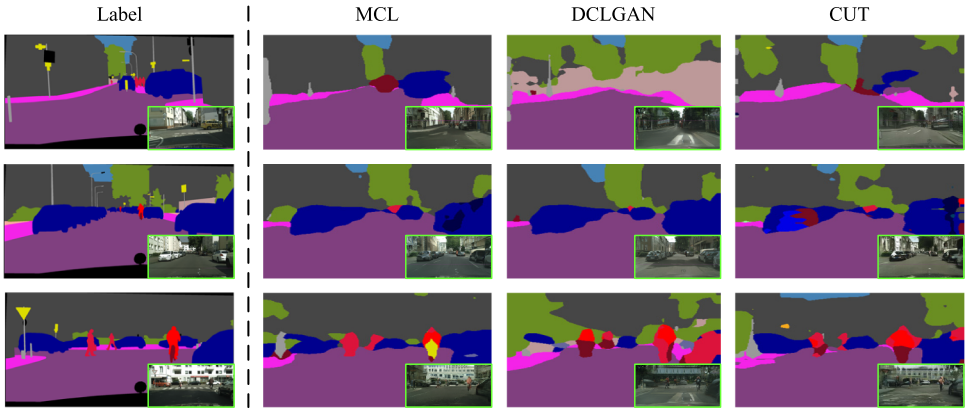

Fig. 5 Example results of our MCL compared to DCLGAN [16] and CUT [36] on the cityscapes dataset with 256 × 256 resolution. The left column represents the ground truth and label, and the right three columns represent generated images and semantic labels by different methods. MCL achieves generated images more like the ground truth, and the semantic labels obtained through the pre-trained DRN model [45] are more like the real labels

show failing cases of other approaches, and our approach yielded relatively satisfactory results.

For cityscapes, Table 1 reports the semantic segmentation metrics on a pre-model FCN-8s model [20,32], and our method achieves the highest performance on three metrics (pixAcc, classAcc, IoU) compared to all the baselines. Figures 4 and 5 show qualitative comparison results of our method with the two most advanced unpaired methods [16,36] on semantic labels to real tasks (Cityscapes dataset). Our MCL achieves generated images more similar to the ground truth, and the semantic labels obtained through the pre-trained FCN-8s model are more similar to the real labels.

We further compare our methods with three popular paired(supervised) methods, Pix2Pix [20], photo-realistic image synthesis system CRN [4] and discriminative region proposal adversarial network DRPAN [41] on the Cityscapes dataset. The quantitative comparison results of our method with other baselines are shown in Table 2. We leverage a pre-trained FCN-8s model [20,32] to calculate three semantic segmentation metrics. Our two versions outperform supervised methods and even approach the ground truth on three metrics (pixAcc, classAcc, IoU). It shows the superiority of our method for semantic labels to real tasks.

Single image translation

Although SinCUT, another variant of CUT [36], has beaten current methods [12,25,44] in the single image translation tasks, the detailed textures of the generated images do not seem realistic enough.

Like SinCUT, our method is also suitable for single image translation, named SinMCL. Experiments are performed on the Monet→Photo and Van Gogh→Photo datasets. Figure 6 shows a qualitative comparison between SinMCL and SinCUT. It is not difficult to find that our generated image has superior visual performance. For example, SinCUT gen-

Table 2 Quantitative comparison of our MCL and FastMCL with other models [4,20,41] on semantic labels to real tasks (Cityscapes dataset) by FCN-8s score

Method	pixAcc	classAcc	IoU
Pix2Pix [20]	0.66	0.23	0.17
CRN [4]	0.69	0.21	0.20
DRPAN [41]	0.72	0.22	0.19
FastMCL(ous)	0.76	0.25	0.19
MCL(ous)	**0.78**	**0.26**	**0.21**
Ground Truth	0.80	0.26	0.21

Best values are in bold
Our two versions outperform supervised methods and even approach the ground truth in three metrics, indicating the superiority of our method

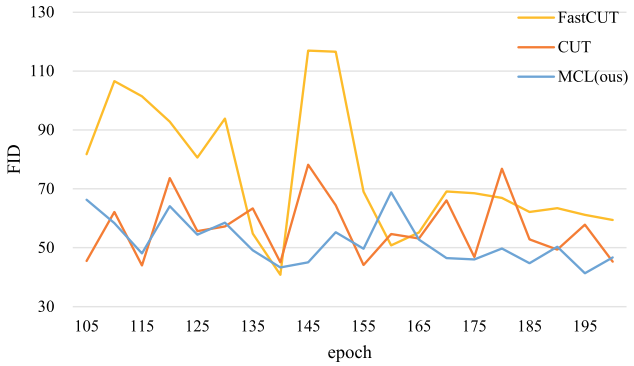

Fig. 7 Training curves of different methods in terms of FID on the Horse→Zebra dataset. When the MCL loss is not considered, our method degenerates into CUT [36]. If the PatchNCE loss $L_{\text{PatchNCE}}(G, H, Y)$ is not considered, it degrades to FastCUT [36]. It is not difficult to see how adding the MCL loss can stabilize the training process

erated some redundant noises in the red box area on the Monet→Photo dataset, but our application can eliminate these noises well. On the Van Gogh→Photo dataset, SinMCL successfully translated it into a real pear tree. Moreover, more details can be seen after magnification.

Ablation study

Compared with all baselines, our proposed method achieves superior performance on image translation tasks. Next, we consider the influence of different loss terms on the experimental results. To save computing resources and training time, we performed an ablation study on the Horse→Zebra dataset. The final objective loss in this paper consists of four loss items, including one adversarial loss, two PatchNCE losses, and one MCL loss, as shown in Eq. 8. The coefficients of these four loss terms are 1, λ_X, λ_Y, and λ_M, respectively. When $\lambda_M = 0$, our proposed method degenerates into CUT. When $\lambda_M = \lambda_X = 0$, the method degenerates into FastCUT. When $\lambda_M = \lambda_X = \lambda_Y = 0$, the method degenerates into standard GAN, which can no longer adapt to image translation tasks. Figure 7 shows the training curves of different methods on the Horse→Zebra dataset, and it shows that increasing the MCL loss term can stabilize the training process.

In Table 3, we further show the quantitative results of different methods on the Horse→Zebra dataset. We calculated the minimum, maximum, mean, and standard deviation of FID during the training process. The mean and standard deviation of FID obtained by our proposed method are the smallest, which indicates that our method has better stability. Although during the first 200 epochs of the training process, FastCUT achieved the best FID with a score of 40.8. How-

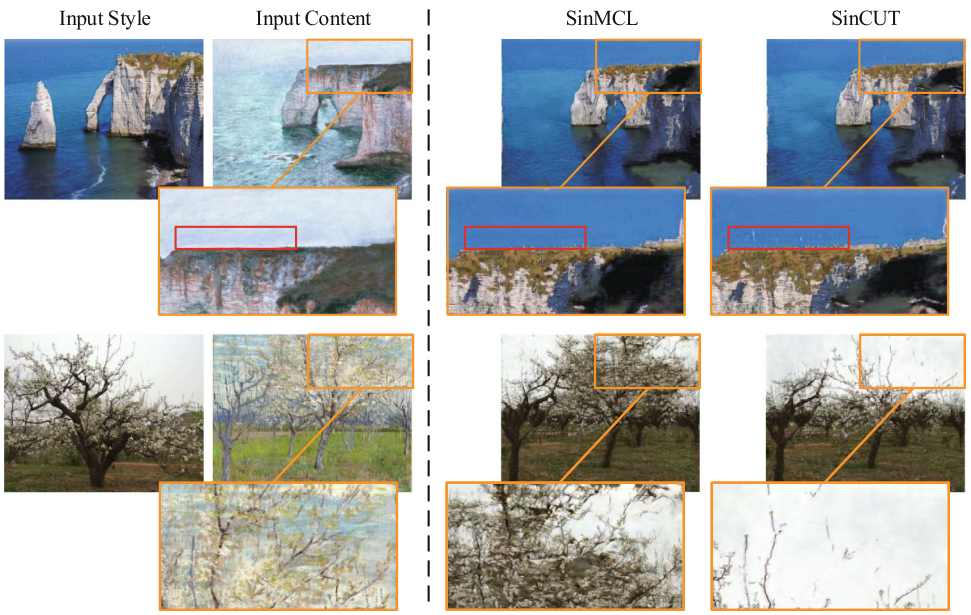

Fig. 6 Single painting to photo translation. We transferred the paintings of Claude Monet and Van Gogh to a nature photograph. There is only one high-resolution image per domain in the dataset used. Our approach shows superior performance in detail. More details can be seen after magnification

Table 3 Minimum (min), maximum (max), mean and standard deviation (SD) of FID on the Horse→Zebra dataset, calculated at 105, 110, ... , 200 epochs

Method	Min	Max	Mean	SD
FastCUT [36]	**40.8**	116.9	75.6	21.7
CUT [36]	44.0	78.2	56.8	10.7
MCL(ous)	41.4	**68.8**	**51.9**	**7.6**

Best values are in bold
Our method achieves a slightly worse minimum of FID than FastCUT [36] over the first 200 epochs. However, MCL obtains a much smaller SD of FID compared to CUT [36] and FastCUT. This shows that our method is more stable than others

Table 4 Influence of different values of hyperparameters on experimental results. We conducted an ablation study on the Horse→Zebra dataset

Model	λ_X	λ_Y	λ_M	FID
FastCUT [36]	1	×	×	73.4
FastMCL	1	×	0.1	70.0
FastMCL (ours)	1	×	0.01	**46.5**
CUT [36]	1	1	×	45.5
MCL	1	1	0.1	43.9
MCL(ours)	1	1	0.01	**40.7**

Best values are in bold
Experiments show that the best results are obtained when $\lambda_X = \lambda_Y = 1$ and $\lambda_M = 0.01$

ever, its training process is volatile, and the next FID would jump to a large value, as shown in Fig. 7. Compared to FastCUT, our method achieves a slightly worse FID with a score of 41.4. Nevertheless, our training process is more stable.

As shown in Fig. 8, we further provide a visual evaluation of the best FID achieved by different methods. During the 115th training process, CUT received the best FID with a score of 44.0. However, as shown in the red box, it appears unnatural of the head and buttock of the generated zebra by CUT. The generated zebra has no eyes on its head, and the stripes of the buttock do not match those of its body. During the 140th training process, FastCUT received the best FID with a score of 40.8. It has similar problems, such as the stripes of the head do not match those on other parts of its body. During the 195th training process, MCL received the best FID with a score of 41.4. Compared to other methods, the generated zebra by MCL looks more realistic.

Next, we explained the value of hyperparameter in this paper. First of all, in Eq. 6, ω aims to scale the distance between feature vectors, which is directly set to 0.1. Then, to ensure the balance between each loss term, we conducted an ablation study for the values of λ_X, λ_Y, and λ_M, as shown in Table 4 and Fig. 9. It is not difficult to find that adding our MCL loss can effectively improve the FID value and the quality of the generated images, and the effect is best when λ_X, λ_Y, and λ_M are 1, 1, 0.01, respectively. Therefore, unless otherwise specified, we choose $\lambda_X = \lambda_Y = 1$ and $\lambda_M = 0.01$.

Many experiments show that our method is superior to previous methods in image-to-image translation tasks. This is

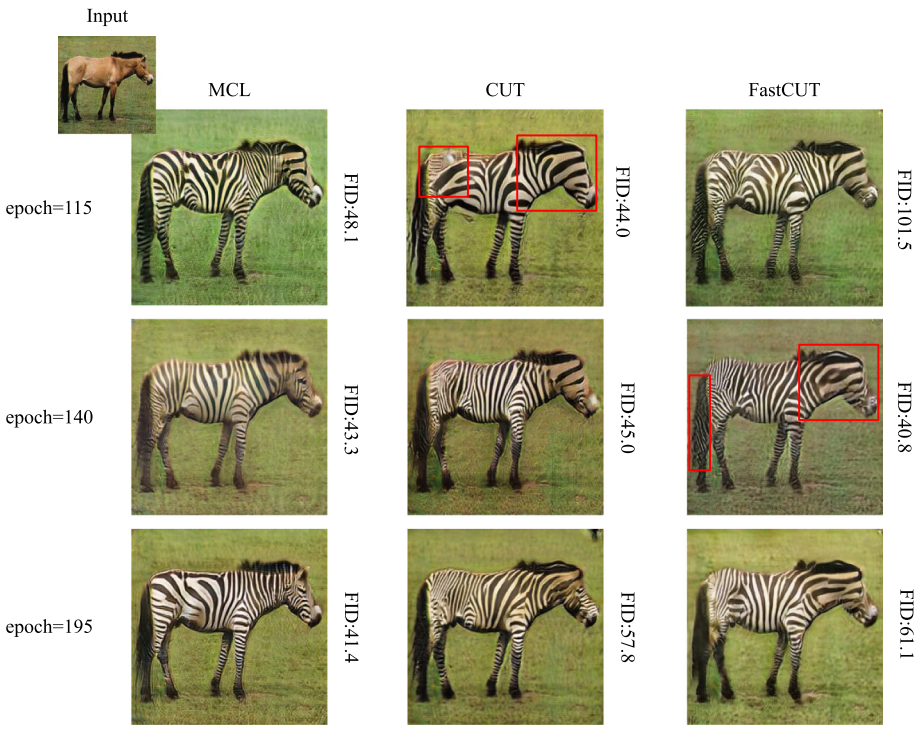

Fig. 8 Visual evaluation of different methods on the Horse→Zebra dataset. We show the visual effects of each approach in turn on three crucial epochs. When CUT [36] or FastCUT [36] reaches the minimum of FID, the generated image does not look realistic, as shown in the red box. Instead, our approach achieves a more realistic image. Furthermore, only the FID of our approach decreases with epochs, while the FID of other methods fluctuates

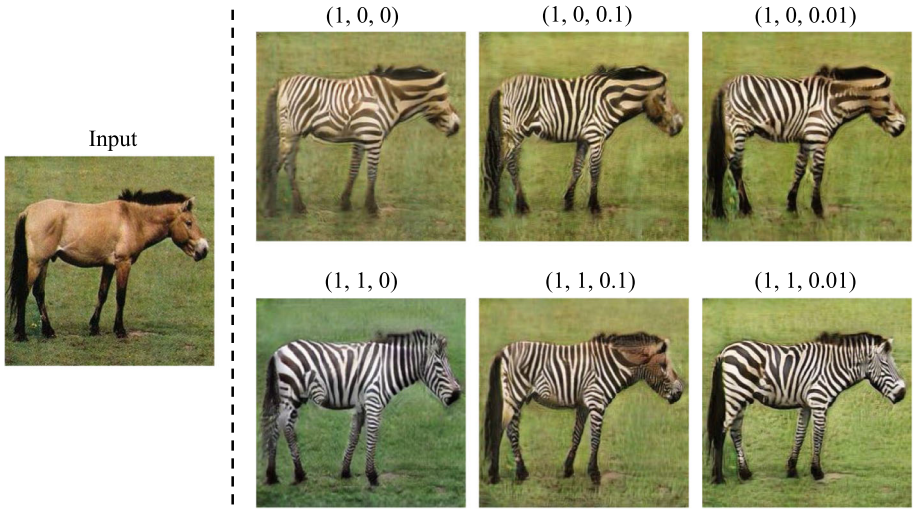

Fig. 9 Visual results of different values of hyperparameters on the Horse→Zebra dataset. $(\lambda_X, \lambda_Y, \lambda_M)$ represents the values of the hyperparameters λ_X, λ_Y, and λ_M. It can be seen that when $\lambda_X = \lambda_Y = 1$ and $\lambda_M = 0.01$, the generated image's quality is superior to others, specifically reflected in its zebra stripes are clearer and more realistic

Table 5 FID values of different methods on the Horse→Zebra dataset

Method	sec/iter↓	Model Parameters↓	FID
CycleGAN [47]	0.40	28.286M	77.2
CycleGAN + MCL loss	0.41	28.286M	**70.1**
DCLGAN [16]	0.41	28.812M	43.2
DCLGAN + MCL loss	0.42	28.812M	**39.6**
SimDCL [16]	0.47	28.852M	47.1
SimDCL + MCL loss	0.48	28.852M	**39.7**

Best values are in bold
When our MCL loss was directly added to CycleGAN, DCLGAN and SimDCL, the FID value increased by 7.1, 3.6 and 7.4, respectively. Simultaneously, the training time and model parameters are hardly increased

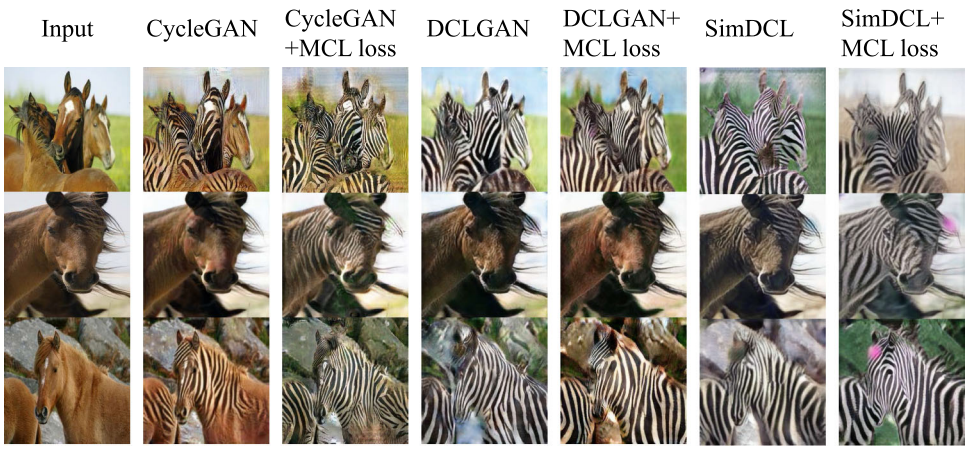

Fig. 10 Visual evaluation of different methods on the Horse→Zebra dataset. When our MCL loss was directly added to CycleGAN, DCLGAN, and SimDCL, the visual effects of the generated images were significantly improved. In general, adding MCL loss resulted in varying degrees of more realistic zebra stripes in the generated images

mainly due to our proposed MCL loss. We skillfully construct MCL loss using the feature information of the discriminator output layer, which already exists, so the MCL loss hardly increases the training time and computing resources. MCL loss is simple and efficient, and to verify this, we conducted experiments on the Horse→Zebra dataset. We directly added MCL loss to the existing methods. All experimental results showed that adding our MCL loss could effectively improve the quality of the generated images, as shown in Table 5 and Fig. 10.

Conclusion

We propose a straightforward method to construct a contrastive loss using the feature information of the discriminator output layer, which is named MCL. Our proposed method enhances the performance of the discriminator and solves the problem of model collapse effectively. Extensive experiments show that our method achieves state-of-the-art results in unpaired image-to-image translation by making better use of contrastive learning. Moreover, our method performs com-

parably or superior to paired methods on semantic labels to real tasks. In addition, we also propose two MCL variants, namely FastMCL and SinMCL. The former is a faster and lighter version for unpaired image-to-image translation tasks, and the latter is suitable for single image translation tasks. FastMCL and SinMCL have achieved great results in their tasks, respectively.

A Pseudo-code

Here, we provide the pseudo-code of MCL loss in the PyTorch style.

```
import torch
import torch.nn.functional as F
# real, fake and netD represent real image, fake image, and discriminator, respectively.
pred_real = netD(real)
pred_fake = netD(fake.detach())
mcl_fake = F.normalize(pred_fake.view(-1, 30))
mcl_real = F.normalize(pred_real.view(-1, 30))
loss_mcl = mcl(mcl_fake, mcl_real, ω)
# Input: mcl_fake is f(A_(y')).
# Input: mcl_real is f(A_(y)).
# Input: ω is the hyperparameter used in MCL loss.
# Output: MCL loss.
def mcl(mcl_fake, mcl_real, ω)
  N = mcl_fake.size(0)
  _out = [mcl_fake, mcl_real]
  outputs = torch.cat(_out, dim = 0)
  sim_matrix = outputs @ outputs.t()
  sim_matrix = sim_matrix / ω
  sim_matrix.fill_diagonal_(-5e4)
  mask = torch.zeros_like(sim_matrix)
  mask[N:, N:] = 1
  mask.fill_diagonal_(0)
  sim_matrix = sim_matrix[N:]
  mask = mask[N:]
  mask = mask / mask.sum(1, keepdim = True)
  lsm = F.log_softmax(sim_matrix, dim = 1)
  lsm = lsm * mask
  d_loss = -lsm.sum(1).mean()
  return d_loss
```

References

1. Baek K, Choi Y, Uh Y, Yoo J, Shim H (2021) Rethinking the truly unsupervised image-to-image translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 14154–14163
2. Benaim S, Wolf (2017) One-sided unsupervised domain mapping. In: NIPS, pp. 752–762 http://papers.nips.cc/paper/6677-one-sided-unsupervised-domain-mapping
3. Chaitanya B, Mukherjee S (2021) Single image dehazing using improved cyclegan. J Vis Commun Image Represent 74:103014
4. Chen Q, Koltun V (2017) Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1511–1520
5. Chen J, Chen J, Chao H, Yang M (2018) Image blind denoising with generative adversarial network based noise modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3155–3164
6. Chen T, Kornblith S, Norouzi M, Hinton GE (2020) A simple framework for contrastive learning of visual representations. CoRR. arXiv:2002.05709
7. Choi Y, Uh Y, Yoo J, Ha JW Stargan (2020) v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8188–8197
8. Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3213–3223
9. Dash A, Ye J, Wang G (2021) A review of generative adversarial networks (gans) and its applications in a wide variety of disciplines—from medical to remote sensing
10. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 https://doi.org/10.1109/CVPR.2009.5206848
11. Fu H, Gong M, Wang C, Batmanghelich K, Zhang K, Tao D (2019) Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In: CVPR, pp. 2427–2436
12. Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414–2423
13. GM, H., Gourisaria, M.K., Pandey, M., Rautaray, S.S. (2020) A comprehensive survey and analysis of generative models in machine learning. Comput Sci Rev 38:100285
14. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial nets. In: NIPS, pp. 2672–2680 . http://papers.nips.cc/paper/5423-generative-adversarial-nets
15. Gutmann M, Hyvärinen A (2010) Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Y.W. Teh, M. Titterington (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol. 9, pp. 297–304. PMLR, Chia Laguna Resort, Sardinia, Italy https://proceedings.mlr.press/v9/gutmann10a.html
16. Han J, Shoeiby M, Petersson L, Armin MA (2021) Dual contrastive learning for unsupervised image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 746–755
17. He K, Fan H, Wu Y, Xie S, Girshick R (2020) Momentum contrast for unsupervised visual representation learning. In: Proceedings of

the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9729–9738
18. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NIPS, pp. 6629–6640 http://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium
19. Huang X, Liu MY, Belongie S, Kautz J (2018) Multimodal unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 172–189
20. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1125–1134
21. Jeong J, Shin J (2021) Training gans with stronger augmentations via contrastive discriminator. In: International Conference on Learning Representations . https://openreview.net/forum?id=eo6U4CAwVmg
22. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision - ECCV 2016. Springer International Publishing, Cham, pp 694–711
23. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T (2020) Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8110–8119
24. Kim T, Cha M, Kim H, Lee JK, Kim J (2017) Learning to discover cross-domain relations with generative adversarial networks. In: D. Precup, Y.W. Teh (eds.) Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 70, pp. 1857–1865. PMLR https://proceedings.mlr.press/v70/kim17a.html
25. Kolkin N, Salavon J, Shakhnarovich G (2019) Style transfer by relaxed optimal transport and self-similarity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10051–10060
26. Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4681–4690
27. Lee HY, Tseng HY, Huang JB, Singh M, Yang, MH (2018) Diverse image-to-image translation via disentangled representations. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 35–51
28. Li R, Pan J, Li Z, Tang J (2018) Single image dehazing via conditional generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8202–8211
29. Li T, Qian R, Dong C, Liu S, Yan Q, Zhu W, Lin L (2018) Beautygan: Instance-level facial makeup transfer with deep generative adversarial network. In: Proceedings of the 26th ACM International Conference on Multimedia, MM '18, p. 645-653. Association for Computing Machinery, New York, NY, USA https://doi.org/10.1145/3240508.3240618
30. Liu MY, Breuel T, Kautz J (2017) Unsupervised image-to-image translation networks. In: Guyon I, Luxburg UV, . Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf
31. Liu R, Ge Y, Choi CL, Wang X, Li H (2021). Divco: Diverse conditional image synthesis via contrastive generative adversarial network. CoRR. arXiv:2103.07893
32. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3431–3440
33. Mao X, Li Q, Xie H, Lau RY, Wang Z, Smolley SP (2017) Least squares generative adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2813–2821. https://doi.org/10.1109/ICCV.2017.304
34. Monday HN, Li J, Nneji GU, Nahar S, Hossin MA, Jackson J, Oluwasanmi A (2022) A wavelet convolutional capsule network with modified super resolution generative adversarial network for fault diagnosis and classification. Complex Intell Syst: 1–17
35. Park T, Liu MY, Wang TC, Zhu JY (2019) Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2337–2346
36. Park T, Efros AA, Zhang R, Zhu JY (2020) Contrastive learning for unpaired image-to-image translation. In: Vedaldi A, Bischof H, Brox T, Frahm JM (eds) Computer Vision - ECCV 2020. Springer International Publishing, Cham, pp 319–345
37. Salehi P, Chalechale A, Taghizadeh M (2020) Generative adversarial networks (gans): An overview of theoretical model, evaluation metrics, and recent developments. CoRR. arXiv:2005.13178
38. van den Oord A, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. CoRR. arXiv:1807.03748
39. Wang TC, Liu MY, Zhu JY, Tao A, Kautz J, Catanzaro B (2018) High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8798–8807
40. Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Qiao Y, Change Loy C (2018) Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops
41. Wang C, Zheng H, Yu Z, Zheng Z, Gu Z, Zheng B (2018) Discriminative region proposal adversarial networks for high-quality image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV)
42. Wu Z, Xiong Y, Yu SX, Lin D (2018) Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3733–3742
43. Yi Z, Zhang H, Tan P, Gong M (2017) Dualgan: Unsupervised dual learning for image-to-image translation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2849–2857
44. Yoo J, Uh Y, Chun S, Kang B, Ha JW (2019) Photorealistic style transfer via wavelet transforms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9036–9045
45. Yu F, Koltun V, Funkhouser T (2017) Dilated residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 472–480
46. Zhang R, Isola P, Efros AA (2016) Colorful image colorization. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision - ECCV 2016. Springer International Publishing, Cham, pp 649–666
47. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2223–2232