Dual Contrastive Learning for Unsupervised Image-to-Image Translation

Junlin Han[1,2] Mehrdad Shoeiby[1] Lars Petersson[1] Mohammad Ali Armin[1]
[1]DATA61-CSIRO, [2]Australian National University
{junlin.han, mehrdad.shoeiby, lars.petersson, ali.armin}@data61.csiro.au

Abstract

Unsupervised image-to-image translation tasks aim to find a mapping between a source domain X and a target domain Y from unpaired training data. Contrastive learning for Unpaired image-to-image Translation (CUT) yields state-of-the-art results in modeling unsupervised image-to-image translation by maximizing mutual information between input and output patches using only one encoder for both domains. In this paper, we propose a novel method based on contrastive learning and a dual learning setting (exploiting two encoders) to infer an efficient mapping between unpaired data. Additionally, while CUT suffers from mode collapse, a variant of our method efficiently addresses this issue. We further demonstrate the advantage of our approach through extensive ablation studies demonstrating superior performance comparing to recent approaches in multiple challenging image translation tasks. Lastly, we demonstrate that the gap between unsupervised methods and supervised methods can be efficiently closed.

1. Introduction

The image-to-image translation task aims to convert images from one domain to another domain, e.g., horse to zebra, low-resolution images to high-resolution images, image to label, photography to painting, and vice versa. Image-to-image translation has drawn considerable attention due to its wide range of applications including style-transfer [46, 19, 24, 34, 1], image in-painting [36], colourisation [44], super-resolution [21, 43], dehazing [27], underwater image restoration [13], and denoising [2].

In unsupervised image-to-image translation without paired data, the main problem is that the adversarial loss [11] is significantly under-constrained, that is, there exist multiple possible mappings between the two domains which make the training unstable and, hence, the translation unsuccessful. To restrict the mapping, the contemporary approaches CycleGAN [46], DiscoGAN [22], and DualGAN [42] use a similar idea, the assumption of cycle-consistency [46] which learns the reverse mapping from the target domain back to the source domain and measures whether the reconstruction image is identical to the input image. The cycle-consistency [46] assumption ensures that the translated images have similar texture information to the target domain, failing to perform geometry changes. Also, the cycle-consistency [46] assumption forces the relationship between the two domains to be a bijection [26]. This is usually not ideal. For example, in the horse to zebra image translation task, the reconstruction is constrained via a fidelity loss, compromising image diversity.

To address this constraint, recently contrastive learning between multiple views of the data has achieved state-of-the-art performance [15, 5, 17, 33] in the field of self-supervised representation learning. This was followed by CUT [34] introducing contrastive learning for unpaired image-to-image translation with a patch-based, multi-layer PatchNCE loss to maximize the mutual information between corresponding patches of input and output images.

While CUT [34] demonstrated the efficiency of contrastive learning, we believe certain design choices are limiting its performance. For example, one embedding was used for two distinct domains which may not efficiently capture the domain gap. To further leverage contrastive learning and avoid the drawbacks of cycle-consistency [46], we propose our dual contrastive learning approach which is referred to as DCLGAN.

DCLGAN aims to maximize mutual information by learning the correspondence between input and output image patches using separate embeddings. By employing different encoders and projection heads for different domains, we learn suitable embeddings to maximize agreement. The dual learning setting [42] also helps to stabilize training. Besides, we revisit some design choices and find that removing RGB pixels representing small patches, in the PatchNCE loss, can be beneficial. We show that cycle-consistency [46] is unnecessary and in fact counter-intuitive when there is no strict constraint on geometrical structure. Lastly, a variant of DCLGAN, referred to as SimDCL, significantly avoids mode collapse.

This paper presents a novel framework and its' variant that can break the limitations of CycleGAN [46] (limited

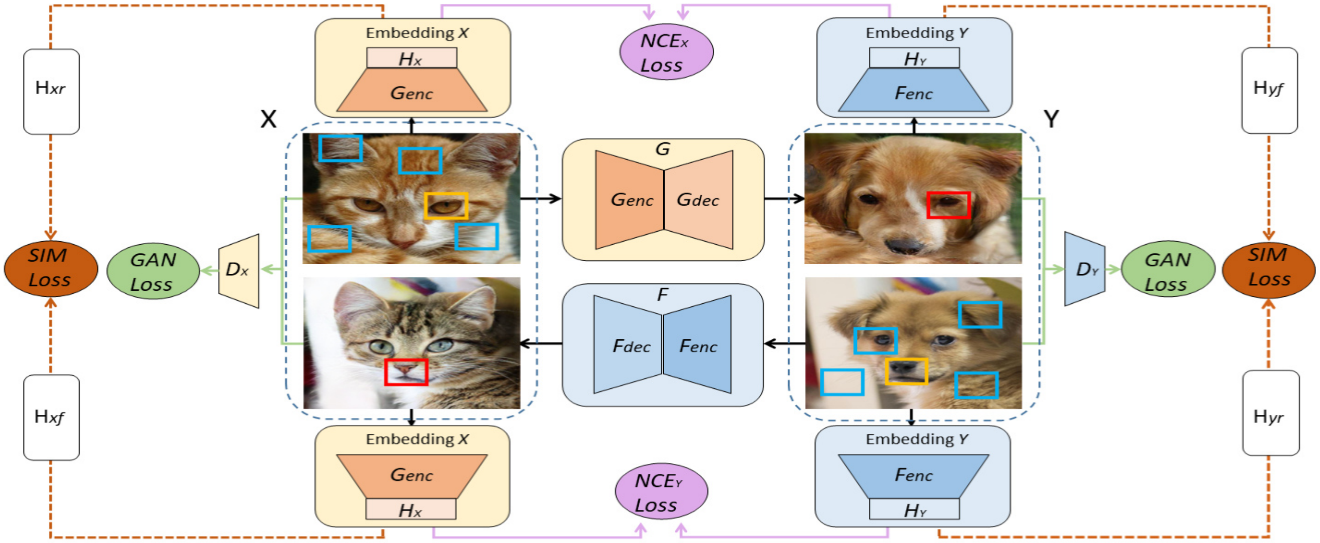

Figure 1. Overall architecture of DCLGAN: By dually learning two mappings $G : X \rightarrow Y$ and $F : Y \rightarrow X$, we successfully enable unpaired image-to-image translation without cycle-consistency. We define the encode half of G and F to be G_{enc}, F_{enc}. We use G_{enc} and H_X as embedding X and F_{enc} and H_Y as embedding Y. We depict here the GAN loss (green line), the patch-based multiplayer PatchNCE loss (purple line). We omit the identical loss here. For variant SimDCL, we add the similarity loss between real images and fake images belonging to the same domain (dashed orange line). PatchNCE loss helps the generated fake image red patch to be similar to its real input image yellow patch while dissimilar to other blue patches.

performance in geometry changes) and CUT [34] (suffering mode collapse and a few inferior results). Through extensive experiments, we demonstrate the quantitative and qualitative superiority of our method compared to several state-of-the-art methods on various popular tasks. Additionally, we show that our method successfully closes the gap between unsupervised and supervised methods, as contrastive learning has done in the field of self-supervised learning. A comprehensive ablation study demonstrates the effectiveness of DCLGAN. Our code is available at GitHub.

2. Related Work

Image-to-image Translation. GANs [11] have been applied to a multitude of image applications, especially in image-to-image translation. The key to the success of GANs is the idea of adversarial loss [11], which forces the generated image to be indistinguishable in principle from the real image. Generally, image-to-image translation can be categorized into two groups: a paired setting (supervised) [41, 20, 35] and an unpaired setting (unsupervised) [46, 22, 42]. Paired setting means the training set is paired, every image from domain X has a corresponding image from domain Y.

Supervised methods. In this line, Pix2Pix [20] first achieved task-agnostic image translation supporting multiple image-to-image translation tasks using only a general method. It has then been extended to Pix2PixHD [41] enabling synthesizing high-resolution photo-realistic images. SPADE [35] introduces the spatially-adaptive normalization layer to further improve the quality of generated images. These supervised approaches require paired data for training, which imposes a limitation on their usage.

Unsupervised methods. In unsupervised settings, the current methods [46, 19, 34, 24, 7, 42, 1, 22, 10, 29, 25, 42, 4, 45] are mainly developed based on two assumptions: a shared latent space [29] and a cycle-consistency assumption [46]. UNIT [29] proposes a shared latent space assumption which assumes a pair of corresponding images in different domains can be mapped to the same latent representation in a shared-latent space. Recent works [19, 24, 7, 25, 6] further enable multi-modal and multi-domain synthesis to bring diversity in the translated outputs. MUNIT [19] disentangles domain-specific features by splitting the latent space into style code and content code. DRIT [24, 25] embeds images onto two spaces including a domain-specific attribute space and a content space capturing shared information. StarGAN [6, 7] employs a unified model architecture to translate images across multiple domains.

Break the cycle. CycleGAN [46] learns two mappings simultaneously via translating an image to the target domain and back preserving the fidelity of the input and the reconstructed image. This leads it to be too restrictive. Recently, a few methods [34, 32, 1, 10] have tried to break the cycle to alleviate the problem of cycle-consistency [46]. CouncilGAN [32] uses more than two generators and discriminators along with the council loss. DistanceGAN [1] and GCGAN [10] enable one-way translation. They employ different constraints from different aspects. We take the advantages of both CycleGAN [46] and CUT [34], employing the idea of mutual information maximization to enable two-sided unsupervised image-to-image translation based on the architectures of CycleGAN [46].

Contrastive learning. In the field of unsupervised representation learning [5, 15, 17, 33], contrastive learning aims to learn an embedding where associated signals are pulled together while other samples in the dataset are pushed away. Signals may vary depending on specific tasks. In general, the objective is to discriminate its transformed version against other samples. To get the transformed version, data augmentation shows the most successful results [5, 15]. However, natural transformations can draw comparable results [17, 14] if ideal natural sources such as those arising audio and optical flow in videos are available. For image-to-image translations, patches are ideal natural sources for instance discrimination since they are easy to track and use [17, 33, 34]. CUT [34] first applies noise contrastive estimation to image-to-image translation tasks by learning the correspondence between input image patches and the corresponding generated image patches, achieving a performance superior to those based on cycle-consistency [46]. We further rethink several design choices of leveraging contrastive learning, making it more beneficial to employ contrastive learning for unsupervised image-to-image translation. We extend one-sided mapping to two-sided, performing better in learning embeddings and thus achieving new state-of-the-art results. We additionally address the mode collapse problem that previous methods based on mutual information maximization can not handle.

3. Method

Given two domains $\mathcal{X} \subset \mathbb{R}^{H\times W\times C}$ and $\mathcal{Y} \subset \mathbb{R}^{H\times W\times 3}$ and a dataset of unpaired instances X containing some images x and Y containing some images y. We denote them $X = \{x \in \mathcal{X}\}$ and $Y = \{y \in \mathcal{Y}\}$. We aim to learn two mappings $G : X \rightarrow Y$ and $F : Y \rightarrow X$.

DCLGAN has two generators G, F as well as two discriminators D_X, D_Y. G enables the mapping from domain X to domain Y and F enables the reverse mapping. D_X and D_Y ensure that the translated images belong to the correct image domain. The first half of the generators are defined as encoder while the second half are decoders and presented as G_{enc} and F_{enc} followed by G_{dec} and F_{dec} respectively.

For each mapping, we extract features of images from four layers of the encoder and send them to a two-layer MLP projection head (H_X and H_Y). Such a projection head learns to project the extracted features from the encoder to a stack of features. Note that we use G_{enc} and H_X as the embedding for domain X and use F_{enc} and H_Y as the embedding for domain Y. If two domains share common semantic information such as horse and zebra, using one encoder can provide reasonable results. However, this may fail to capture the variability in two distinctive domains with a large gap. Additionally, we introduce four light networks to capture the common information within one domain and form a similarity loss.

Figure 1 shows the overall architecture of DCLGAN and SimDCL. DCLGAN combines three losses including adversarial loss [11], PatchNCE loss, and identity loss [46] whereas SimDCL has one additional similarity loss to address mode collapse. The details of our objective are described below.

3.1. Adversarial loss

An adversarial loss [11] is employed to encourage the generator to generate visually similar images to images from the target domain, for the mapping $G : X \rightarrow Y$ with discriminator D_Y , the GAN loss is calculated by:

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}\left(G, D_Y, X, Y\right) = &\ \mathbb{E}_{y\sim Y}\left[\log D_Y(y)\right] \\ &+ \mathbb{E}_{x\sim X}\left[\log\left(1 - D_Y(G(x))\right)\right],\end{aligned} \tag{1}$$

where G tries to generate images $G(x)$ that look similar to images from domain Y , while D_Y aims to distinguish between translated samples $G(x)$ and real samples y. A similar adversarial loss for the mapping $F : Y \rightarrow X$ and its discriminator D_X is introduced as:

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}\left(F, D_X, X, Y\right) = &\ \mathbb{E}_{x\sim X}\left[\log D_X(x)\right] \\ &+ \mathbb{E}_{y\sim Y}\left[\log\left(1 - D_X(G(y))\right)\right].\end{aligned} \tag{2}$$

3.2. Patch-based multi-layer contrastive learning

Mutual information maximization. Our goal is to maximize the mutual information between corresponding patches of the input and the output. For instance, for a patch showing the eye of a generated dog (top-right of Figure 1), we should be able to more strongly associate it with the eye of the input real cat other than the rest of the patches of the cat.

Following the setting of CUT [34], we employ a noisy contrastive estimation framework [33] to maximize the mutual information between inputs and outputs. The idea behind contrastive learning is to correlate two signals, i.e., the "query" and its' "positive" example, in contrast to other examples in the dataset (referred to as "negatives").

We map query, positive, and N negatives to K-dimensional vectors and denote them $v, v^+ \in R^K$ and $v^- \in R^{N\times K}$ respectively. Note that $v_n^- \in R^K$ denotes the n-th negative. We normalize vectors with L2-normalization then set up an $(N+1)$-way classification problem and compute the probability that a "positive" is selected over "negatives". Mathematically, this can be expressed as a cross-entropy loss [12] which is computed by:

$$\ell\left(\boldsymbol{v}, \boldsymbol{v}^{+}, \boldsymbol{v}^{-}\right) = -\log(\frac{\exp\left(\boldsymbol{sim}(v, \boldsymbol{v}^{+})/\tau\right)}{\exp\left(\boldsymbol{sim}(v, \boldsymbol{v}^{+})/\tau\right) + \sum_{n=1}^{N} \exp\left(\boldsymbol{sim}(v, \boldsymbol{v}_{n}^{-})/\tau\right)}), \tag{3}$$

where $\operatorname{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^{\top}\boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$ denotes the cosine similarity between $\boldsymbol{u}$ and $\boldsymbol{v}$. τ denotes a temperature parameter to scale the distance between the query and other examples, we use 0.07 as default.

PatchNCE loss. We use G_{enc} and H_X to extract features from domain X and use F_{enc} and H_Y to extract features from domain Y. We do not share weights in order to learn better embeddings and capture variability in two distinct domains. We select L layers from $G_{enc}(X)$ and send it to H_X, embedding one image to a stack of features $\{\boldsymbol{z}_l\}_L = \left\{H_X^l\left(G_{\text{enc}}^l(\boldsymbol{x})\right)\right\}_L$, where G_{enc}^l represents the output of l-th selected layers.

Now we consider the patches. After having a stack of features, each feature actually represents one patch from the image. We take advantage of that and denote the spatial locations in each selected layer as $s \in \{1, ..., S_l\}$, where S_l is the number of spatial locations in each layer. We select a query each time, refer the corresponding feature ("positive") as $\boldsymbol{z}_l^s \in \mathbb{R}^{C_l}$ and all other features "negatives") as $\boldsymbol{z}_l^{S\backslash s} \in \mathbb{R}^{(S_l-1)\times C_l}$, where C_l is the number of channels in each layer. For the generated fake image $G(x)$ belonging to domain Y, we exploit the advantages of dual learning and use a different embedding of domain Y. Similarly, we get another stack of features $\{\hat{\boldsymbol{z}}_l\}_L = \left\{H_Y^l\left(F_{\text{enc}}^l(G(\boldsymbol{x}))\right)\right\}_L$.

We aim to match the corresponding patches of input and output images. The patch-based, multi-layer PatchNCE loss [34] for mapping $G : X \rightarrow Y$ can be expressed as:

$$\begin{gathered}\mathcal{L}_{\text{PatchNCE}_{\text{X}}}(G, H_X, H_Y, X) = \\ \mathbb{E}_{\boldsymbol{x}\sim X} \sum_{l=1}^{L} \sum_{s=1}^{S_l} \ell\left(\hat{z}_l^s, \boldsymbol{z}_l^s, \boldsymbol{z}_l^{S\backslash s}\right).\end{gathered} \tag{4}$$

Consider the reverse mapping $F : Y \rightarrow X$, we introduce a similar loss as well,

$$\begin{gathered}\mathcal{L}_{\text{PatchNCE}_{\text{Y}}}(F, H_X, H_Y, Y) = \\ \mathbb{E}_{\boldsymbol{y}\sim Y} \sum_{l=1}^{L} \sum_{s=1}^{S_l} \ell\left(\hat{z}_l^s, \boldsymbol{z}_l^s, \boldsymbol{z}_l^{S\backslash s}\right),\end{gathered} \tag{5}$$

where $\{\boldsymbol{z}_l\}_L = \left\{H_Y^l\left(F_{\text{enc}}^l(\boldsymbol{y})\right)\right\}_L$ and $\{\hat{\boldsymbol{z}}_l\}_L = \left\{H_X^l\left(G_{\text{enc}}^l(F(\boldsymbol{y}))\right)\right\}_L$ are different from $G : X \rightarrow Y$.

3.3. Similarity loss

Intuitively, images from the same domain should have some similarities. Their semantics are different but they share a common style. In the dual learning setting, we have one real and one fake image belonging to the same domain in each iteration. After getting four stacks of features, we use four light networks $(H_{xr}, H_{xf}, H_{yr}, H_{yf})$ to project them to 64-dim vectors, where x, y, r, f refers to images within domain X, images within domain Y, real, fake correspondingly. These 64-dim vectors belonging to the same domain can be measured by a similarity loss, such a loss can be formalised as:

$$\begin{aligned}&\mathcal{L}_{\text{sim}}(G, F, H_X, H_Y, H_{xr}, H_{xf}, H_{yr}, H_{yf}) \\ &= [\|H_{xr}(H_X(G_{enc}(x))) - H_{xf}(H_X(G_{enc}(F(y))))\|_1^{sum}] \\ &+ [\|H_{yr}(H_Y(F_{enc}(y))) - H_{yf}(H_Y(F_{enc}(G(x))))\|_1^{sum}],\end{aligned} \tag{6}$$

where sum means we sum them up together. Implementing a similarity loss on the deep features forces the generated images to be realistic, as opposed to mode collapsed outputs, by encouraging the deep features of the generated and real images to be similar.

3.4. Identity loss

In order to prevent generators from unnecessary changes, we add an identity loss [46]. Unlike CUT [34], We do not employ PatchNCE loss as identity loss due to training speed.

$$\begin{aligned}\mathcal{L}_{\text{identity}}(G, F) = \mathbb{E}_{x\sim X}\left[\|F(x) - x\|_1\right] \\ + \mathbb{E}_{y\sim Y}\left[\|G(y) - y\|_1\right].\end{aligned} \tag{7}$$

Such an identity loss can encourage the mappings to preserve color composition between the input and output.

3.5. General objective

DCLGAN. The generated image should be realistic and patches in the input and output images should share same correspondence. We employ identity loss [46] in the default setting. The full objective is estimated by:

$$\begin{aligned}&\mathcal{L}(G, F, D_X, D_Y, H_X, H_Y) \\ &= \lambda_{GAN}(\mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y)) \\ &+ \lambda_{NCE}\mathcal{L}_{\text{PatchNCE}_{\text{X}}}(G, H_X, H_Y, X) \\ &+ \lambda_{NCE}\mathcal{L}_{\text{PatchNCE}_{\text{Y}}}(F, H_X, H_Y, Y) \\ &+ \lambda_{idt}\mathcal{L}_{\text{identity}}(G, F).\end{aligned} \tag{8}$$

We set λ_{GAN} = 1, λ_{NCE} = 2 and λ_{idt} = 1. DCLGAN achieves superior performance to existing methods.

SimDCL. We introduce SimDCL since methods based on mutual information maximization suffer from mode collapse in some specific tasks. We add similarity loss to the full objective of DCLGAN, and name it SimDCL, where sim is short for similarity and DCL stands for dual contrastive learning. The full objective of this variant is:

$$\begin{aligned}
&\mathcal{L}(G, F, D_X, D_Y, H_X, H_Y) \\
&= \lambda_{GAN}(\mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y)) \\
&+ \lambda_{NCE}\mathcal{L}_{\text{PatchNCE}_\text{X}}(G, H_X, H_Y, X) \\
&+ \lambda_{NCE}\mathcal{L}_{\text{PatchNCE}_\text{Y}}(F, H_X, H_Y, Y) \\
&+ \lambda_{sim}\mathcal{L}_{\text{sim}}(G, F, H_X, H_Y, H_1, H_2, H_3, H_4) \\
&+ \lambda_{idt}\mathcal{L}_{\text{identity}}(G, F).
\end{aligned} \tag{9}$$

We set λ_{GAN} = 1, λ_{NCE} = 2, λ_{SIM} = 10 and λ_{idt} = 1. This variant runs slower than DCLGAN, we recommend using it for Photo → Label, semantic segmentation and similar tasks to avoid mode collapse. SimDCL achieves equal or slightly worse performance compared to DCLGAN.

4. Experiments

The training details, datasets, and our evaluation protocol along with all baselines are described as follows.

4.1. Training details

We mostly follow the setting of CUT [34] to train our proposed model. We use Hinge GAN loss [28] instead of LSGAN loss [31]. More specifically, we use the Adam optimiser [23] with β_1 = 0.5 and β_2 = 0.999. DCLGAN is trained for 400 epochs with a learning rate of 0.0001 while SimDCL is trained for 200 epochs with a learning rate of 0.0002 unless specified. The learning rate starts to decay linearly after half of the total epochs. We use a ResNet-based [16] generator with PatchGAN [20] as discriminator. We use a batch size of 1 and instance normalization [39]. All training images are loaded in 286 × 286 then cropped to 256 × 256 patches. More details on the training and the architecture are provided in the supplementary material.

4.2. Datasets

We evaluated our proposed method and baselines on six different datasets with nine tasks.

Horse ↔ Zebra was introduced in CycleGAN, it contains 1067 horse images, 1344 zebra images as the training set and 260 test images all collected from ImageNet [9].

Cat ↔ Dog contains 5000 training images and 500 test images for each domain. It was introduced in StarGAN2 [7]. DCLGAN is trained for 200 epochs only for this dataset.

CityScapes [8] contains 2975 training and 500 validation images for each domain. One domain is city scenes from German cities and the other is semantic segmentation labels. We focus on Label → City. We also leverage labels to measure how well methods discover correspondences.

Van Gogh → Photo contains 400 Van Gogh paintings and 6287 photographs from Flickr. It was collected in CycleGAN [46]. DCLGAN is trained for 200 epochs only for this task. We reuse the training set of Van Gogh paintings as the test set.

Label ↔ Facade is similar to CityScapes, it contains 400 paired training images and 106 paired test images from the CMP Facade Database [38].

Orange → Apple is also from ImageNet [9]. It contains 1019 orange images and 995 apple images in the training set. For testing, we use 248 orange images.

4.3. Evaluation

Metrics We mainly use Fréchet Inception Distance (FID) [18] to measure the quality of generated images. FID [18] shows high correspondence with human perception, it is based on the Inception Score (IS) [37]. Lower FID [18] means lower Fréchet distance between real and generated images. That is to say, lower FID [18] means generated images are more realistic. For cityscapes, following Pix2Pix [20], we use the pre-trained semantic segmentation network FCN-8 [30] and compute three metrics. They are mean class Intersection over Union (IoU), pixel-wise accuracy (pixAcc), and average class accuracy (classAcc).

Baselines. We perform qualitative and quantitative comparison between our proposed method and recent state-of-the-art unsupervised methods including CUT [34], FastCUT [34], CycleGAN [46], MUNIT [19], DRIT [24], DistanceGAN [1], SelfDistance [1] and GCGAN [10]. MUNIT [19] and DRIT [24] are able to generate diverse results for only one input image and the others only produce one result. Among them, CUT, FastCUT [34], DistanceGAN, SelfDistance [1] and GcGAN [10] are one-sided methods. The rest are two-sided methods.

5. Results

Here, we compare our algorithms (DCLGAN and SimDCL) with all baselines on different datasets. Further, we compare DCLGAN to supervised methods on the CityScapes dataset using the FCN [30] score, showing that the performance of our method is on par with supervised methods. Lastly, we show that SimDCL avoids mode collapse.

5.1. Comparison of different methods

Table 1 shows a comparison of the quantitative results of DCLGAN and SimDCL with several baselines on three challenging tasks, including CityScapes, Cat → Dog, and Horse → Zebra. We only use the FID [18] score as our quantitative metric. It is evident that our algorithms perform stronger than all the baseline. Figure 2 presents the corresponding randomly selected qualitative results. DCLGAN

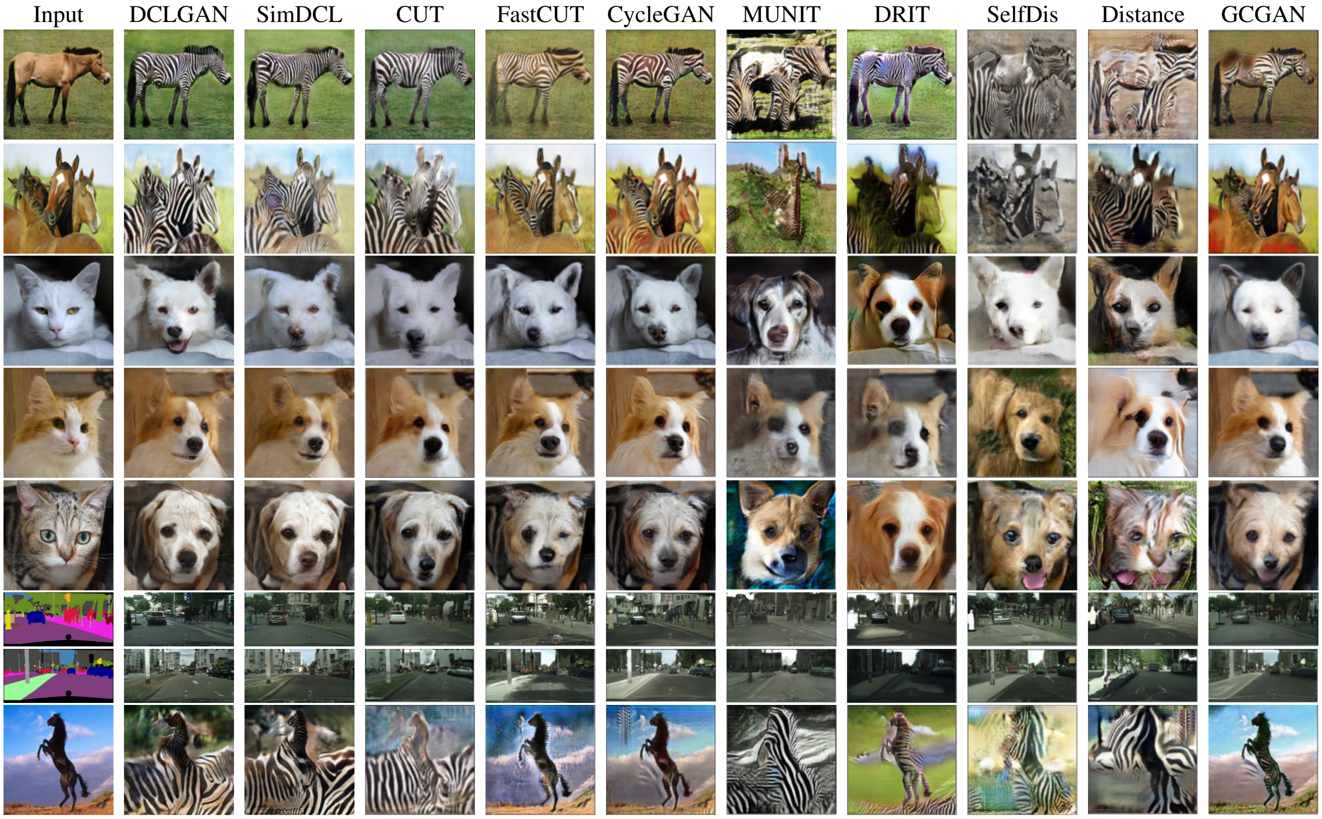

Figure 2. Comparison to all baselines on the Horse→Zebra, Cat→Dog, and CityScapes tasks. DCLGAN and SimDCL show visual satisfactory results. The last row is a failure case, our methods are unable to identify unusual pose and rare background. They fail to distinguish foreground and background, adding zebra textures to the cloud.

	CityScapes	Cat→ Dog	Horse → Zebra		Overall
Method	FID↓	FID↓	FID↓	sec/iter↓	Ranking
CycleGAN [46]	68.6	85.9	66.8	0.40	4
MUNIT [19]	91.4	104.4	133.8	0.39	9
DRIT [22]	155.3	123.4	140.0	0.70	10
Distance [1]	85.8	155.3	72.0	**0.15**	6
SelfDistance [1]	78.8	144.4	80.8	0.16	6
GCGAN [10]	105.2	96.6	86.7	0.62	8
CUT [34]	56.4	76.2	45.5	0.24	3
FastCUT [34]	68.8	94.0	73.4	**0.15**	5
DCLGAN (ours)	**49.4**	**60.7**	**43.2**	0.41	**1**
SimDCL (ours)	51.3	65.5	47.1	0.47	2

Table 1. Comparison to all baselines on the Horse→Zebra, Cat→Dog, and CityScapes tasks. DCLGAN denotes our model without Similarity loss and SimDCL denotes our model with Similarity loss. We show FID [18] score for all tasks. The overall ranking is based on the FID score among all tasks. DCLGAN generates better images with acceptable speed, runs a bit slower than CycleGAN [46]. Our variant SimDCL also shows competitive results.

performs both geometry changes and texture changes with negligible artifacts, this is especially successful in Cat → Dog while other methods can not generate realistic images. It is worth mentioning that models generating multiple outputs perform the worst.

We select the top four methods from Table 1 and set a second comparison among them by testing them in 5 more tasks: Zebra → Horse, Van Gogh → Photo, Dog → Cat, Label → Facade and Orange → Dog. We show quantitative results in Table 2 and randomly picked qualitative results in Figure 3. The results suggest that DCLGAN keeps superior performance comparing to other methods among various tasks. Methods under the cycle-consistency assumption [46] usually fail to perform geometric changes while methods based on mutual information maximization successfully enable both geometric changes and texture changes. This is explicitly shown in Dog → Cat tasks.

5.2. Comparison to supervised methods

Here we compare our DCLGAN method with three popular supervised methods, Pix2Pix [20], photo-realistic image synthesis system CRN [3] and discriminative region proposal adversarial network DRPAN [40] on the CityScapes dataset. We follow the setting in Pix2Pix [20] and use a pre-trained semantic segmentation network FCN-8 [30] to compute the FCN score. Quantitative results are shown in Table 3. DCLGAN performs best in pixACC and significantly closes the gap between unsupervised methods and supervised methods. On average, our method performs on par with supervised methods.

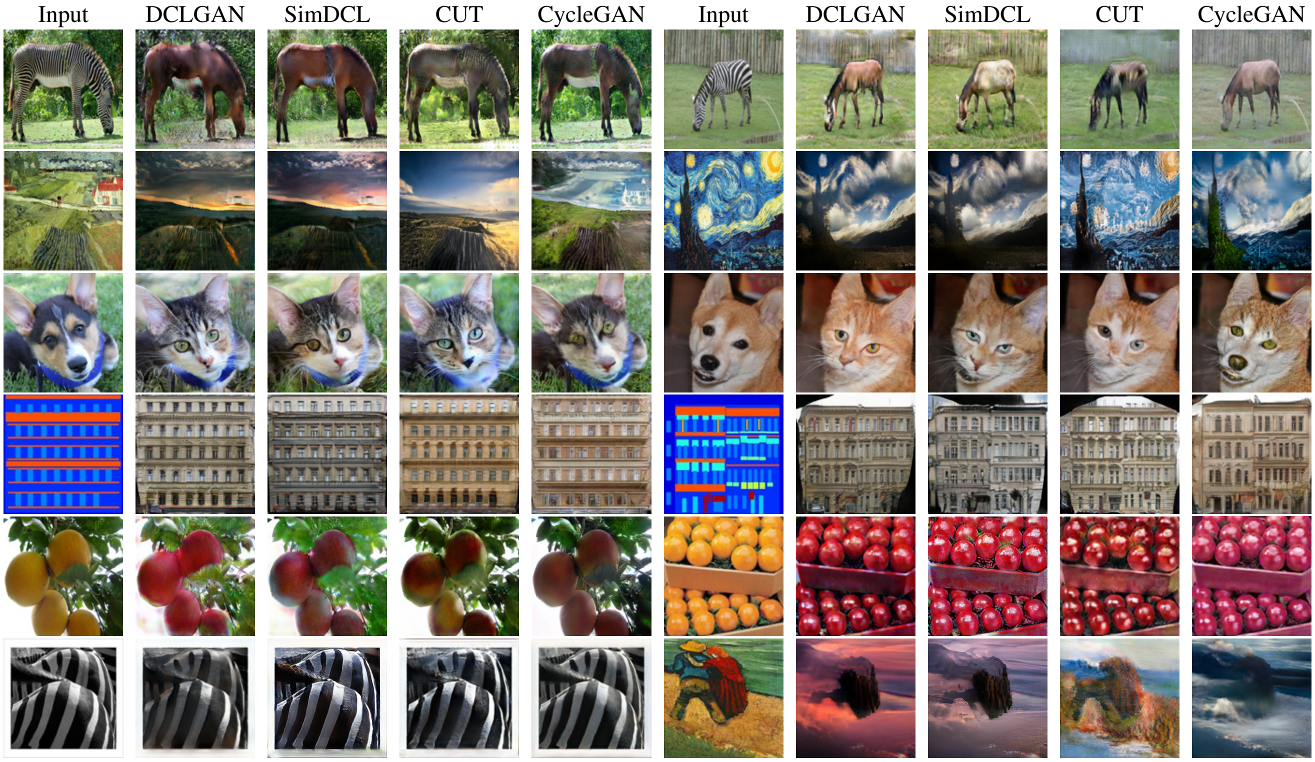

Figure 3. Comparison between the best four methods on five more tasks including Zebra → Horse, Van Gogh → Photo, Dog → Cat, Label → Facade, and Orange → Apple. We randomly pick two samples for each task. Our DCLGAN performs both geometry changes and texture changes. The last row show two typical failure cases, the first one fails to translate the image since input is only a small part of zebra, the last one fails to keep the structure of humans, translating them to Yosemite.

	Zebra→ Horse	Van Gogh → Photo	Dog → Cat		Label → Facade	Orange → Apple	Model
Method	**FID↓**	**FID↓**	**Overall Runtime↓**	**FID↓**	**FID↓**	**FID↓**	**Parameters**
CycleGAN [46]	154.3	103.0	106hr	107.7	127.5	**117.7**	28.286M
CUT [34]	170.5	96.9	125hr	26.8	119.7	127.0	14.406M
DCLGAN (ours)	**139.5**	93.7	108hr	**22.2**	**119.2**	124.9	28.812M
SimDCL (ours)	152.5	**93.5**	124hr	22.8	132.3	134.4	28.852M

Table 2. Comparison between the best four methods on Zebra → Horse, Van Gogh → Photo, Dog → Cat, and Label → Facade tasks. DCLGAN still outperforms other methods in most tasks. The overall runtimes are provided for Dog → Cat task, in hours. Note CUT is trained for 400 epochs while the rest for 200 epochs only. The overall ranking circumstances among methods compared in here are identical to the first comparison (Table 1) except for a tie with CycleGAN [46] and CUT [34].

	CityScapes		
Method	**pixAcc↑**	**classAcc↑**	**IoU↑**
DCLGAN(ours)	**0.74**	0.22	0.17
Pix2Pix [20]	0.66	0.23	0.17
CRN [3]	0.69	0.21	**0.20**
DRPAN [40]	0.73	**0.24**	0.19
Ground Truth	0.80	0.26	0.21

Table 3. Comparison between unsupervised DCLGAN and supervised Pix2Pix [20], CRN [3], DRPAN [40] on CityScapes dataset. We follow the setting of Pix2Pix [20] to compute the FCN [30] score. DCLGAN outperforms supervised methods in pixAcc, suggesting the gap between unsupervised methods and supervised methods is closing.

5.3. Addressing mode collapse via similarity loss.

Our final comparison is a stress test on mode collapse. Mode collapse in generation tasks means the outputs lack diversity, and usually, the outputs are not realistic. We find that methods based on mutual information maximization (CUT and DCLGAN) can not prevent mode collapse in Photo → Label and similar tasks. To address this issue, we design SimDCL. We test the best four methods on the Facade → Label task and show the randomly picked visual results in Figure 4. No matter what the input is, the outputs of both CUT and DCLGAN are almost identical while SimDCL generates reasonable outputs for different inputs. SimDCL is more robust to the mode collapse issue compared with other methods based on mutual information

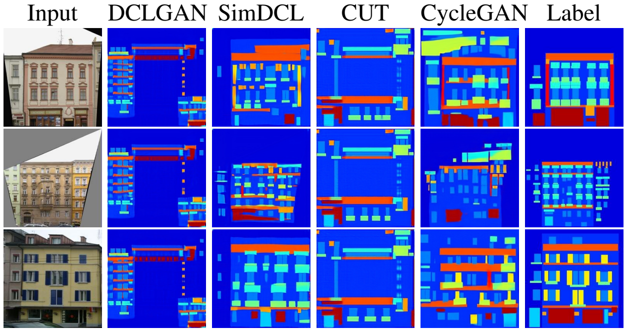

Figure 4. Comparison between the best four methods on Facade → Label task. Methods based on mutual information maximization suffer from mode collapse in this task. We address this by introducing SimDCL to prevent mode collapse. SimDCL also capture more correspondence between facade and label than CycleGAN [46].

maximization.

6. Ablation study

DCLGAN shows superior performance compared to all baselines. We explore what is making contrastive learning effective. We analyze DCLGAN by studying each of our contributions in isolation via conducting several experiments, summarized in Table 4. We use three tasks including Horse → Zebra, Zebra → Horse, and CityScapes in our ablation study.

We show the results of: (I) Adding the first RGB pixels back. (II) Drawing external negatives. (III) Using the same encoder and MLP for one mapping instead of two. (IV) Adding cycle-consistency loss. (V) Removing the dual setting.

	Horse → Zebra	**Zebra → Horse**	**CityScapes**
Ablation	**FID↓**	**FID↓**	**FID↓**
I	49.7	156.7	50.3
II	**41.7**	149.2	49.6
III	44.0	153.4	52.2
IV	44.6	140.6	55.4
V	47.0	151.3	91.5
DCLGAN	43.2	**139.5**	**49.4**

Table 4. Quantitative results for ablations.

(I) CUT [34] uses features from five layers in total including the first RGB pixels in PatchNCE loss ($l = 5$ in Equations 4 and 5). Layers and spatial locations within the feature stack represent patches of the input image. Deeper layers correspond to bigger patches. However, RGB pixels represents the smallest possible patch size (1×1), providing misleading information. We find that not including the RGB layer encourages convergence. In fact, if we adopt the strategy in CUT [34] ($l = 5$), the results deteriorate in all three tasks as demonstrated in Table 4.

(II) Effect of drawing external negatives. CUT [34] states that internal negatives (patches from an input image only) are more effective than external negatives (patches from other images). CUT [34] adds negatives using a momentum encoder [15]. We explore this in a different approach, by taking the advantage of the dual setting. DCLGAN produces four different stacks of features at each iteration. Concatenating two stacks of features belonging to the same domain provides more negatives (255 internal and 256 external) for one query while the default DCLGAN uses 255 internal negatives. We obverse better quantitative results in Horse → Zebra and very close results in CityScapes for this variant. Although the gap of FID score between the default DCLGAN and this variant is small, the visual quality is not as good as that of the default DCLGAN, that is, objects in the generated image tend to be merged together.

(III) Effect of using separate embeddings for each domain. While CUT [34] uses the same embedding for both domains, we use two separate embeddings, one for each domain. Adopting the CUT [34] strategy in our network we find that the results will deteriorate, as demonstrated in Table 4. One embedding fails to capture the variability in two distinct domains, for instance, Photo → Label.

(IV) Effect of Cycle-consistency loss. To test if the cycle-consistency loss can improve the results, we add cycle-consistency [46] loss to our objective. We did not observe any improvements (Table 4). Although cycle-consistency and mutual information maximization share some commonalities, DCLGAN is much less restrictive. DCLGAN focuses on both texture and geometry changes while CycleGAN [46] mostly focuses on texture only. We tested this variant in two tasks requiring geometry changes, Cat → Dog and Dog → Cat, the FID scores are 71.1 and 35.5 respectively, all worse than the original DCLGAN. We conclude that when strict limitations on geometry are not crucial, cycle-consistency [46] loss is better to be avoided.

(V) Dual settings stabilize the training. We remove the dual setting to demonstrate its' effect. We keep other settings the same as DCLGAN. The results are worse than DCLGAN, which shows the dual setting can learn better embeddings for different domains and stabilize the training.

7. Conclusion

We show that a dual setting can better leverage contrastive learning in unsupervised unpaired image-to-image translation. We also revise some significant designs to render contrastive learning more effective. In addition, a variant of DCLGAN, SimDCL mitigates mode collapse. Finally, we show that our method can hugely close the gap between unsupervised and supervised methods in challenging datasets such as CityScape, just as contrastive learning in the field of self-supervised representation learning.

References

[1] Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In *Advances in neural information processing systems (NIPS)*, pages 752–762, 2017. 1, 2, 3, 5, 6

[2] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3155–3164, 2018. 1

[3] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE international conference on computer vision (ICCV)*, pages 1511–1520, 2017. 6, 7

[4] Runfa Chen, Wenbing Huang, Binghui Huang, Fuchun Sun, and Bin Fang. Reusing discriminators for encoding: Towards unsupervised image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8168–8177, 2020. 2

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020. 1, 3

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 8789–8797, 2018. 2

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer vision and pattern recognitio (CVPR)*, pages 8188–8197, 2020. 2, 5

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3213–3223, 2016. 5

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 248–255. Ieee, 2009. 5

[10] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2427–2436, 2019. 2, 3, 5, 6

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*, pages 2672–2680, 2014. 1, 2, 3

[12] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 4

[13] Junlin Han, Mehrdad Shoeiby, Tim Malthus, Elizabeth Botha, Janet Anstee, Saeed Anwar, Ran Wei, Lars Petersson, and Mohammad Ali Armin. Single underwater image restoration by contrastive learning. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021. 1

[14] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in neural information processing systems (NIPS)*, 2020. 3

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020. 1, 3, 8

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer vision and pattern recognitio (CVPR)*, pages 770–778, 2016. 5

[17] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems (NIPS)*, pages 6626–6637, 2017. 5, 6

[19] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 1, 2, 5, 6

[20] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5, 6, 7

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1646–1654, 2016. 1

[22] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017. 1, 2, 6

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2014. 5

[24] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *European conference on computer vision (ECCV)*, pages 35–51, 2018. 1, 2, 5

[25] Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *International Journal of Computer Vision (IJCV)*, pages 1–16, 2020. 2

[26] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. Alice: Towards understanding adversarial learning for joint distribution matching. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5495–5503, 2017. 1

[27] Runde Li, Jinshan Pan, Zechao Li, and Jinhui Tang. Single image dehazing via conditional generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8202–8211, 2018. 1

[28] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 5

[29] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems (NIPS)*, pages 700–708, 2017. 2

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE conference on computer vision and pattern recognition (CVPR)*, pages 3431–3440, 2015. 5, 6, 7

[31] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE international conference on computer vision (ICCV)*, pages 2794–2802, 2017. 5

[32] Ori Nizan and Ayellet Tal. Breaking the cycle - colleagues are all you need. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2020. 2, 3

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 3

[34] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[35] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[36] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *IEEE conference on computer vision and pattern recognition(CVPR)*, pages 2536–2544, 2016. 1

[37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems (NIPS)*, pages 2234–2242, 2016. 5

[38] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition*, pages 364–374. Springer, 2013. 5

[39] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 5

[40] Chao Wang, Haiyong Zheng, Zhibin Yu, Ziqiang Zheng, Zhaorui Gu, and Bing Zheng. Discriminative region proposal adversarial networks for high-quality image-to-image translation. In *European conference on computer vision (ECCV)*, pages 770–785, 2018. 6, 7

[41] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[42] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2849–2857, 2017. 1, 2

[43] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 701–710, 2018. 1

[44] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision (ECCV)*, pages 649–666. Springer, 2016. 1

[45] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss. In *European conference on computer vision (ECCV)*, 2020. 2

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 3, 4, 5, 6, 7, 8