TransferI2I: Transfer Learning for Image-to-Image Translation from Small Datasets

Yaxing Wang[1,2], Héctor Laria[2] Joost van de Weijer[2], Laura Lopez-Fuentes[3], Bogdan Raducanu[2]
[1] PCALab, Nanjing University of Science and Technology, China
[2]Computer Vision Center, Universitat Autònoma de Barcelona, Spain
[3]Universitat de les Illes Balears, Spain
{yaxing,hlaria,joost,bogdan}@cvc.uab.es, l.lopez@uib.es

Abstract

Image-to-image (I2I) translation has matured in recent years and is able to generate high-quality realistic images. However, despite current success, it still faces important challenges when applied to small domains. Existing methods use transfer learning for I2I translation, but they still require the learning of millions of parameters from scratch. This drawback severely limits its application on small domains. In this paper, we propose a new transfer learning for I2I translation (TransferI2I). We decouple our learning process into the image generation step and the I2I translation step. In the first step we propose two novel techniques: source-target initialization and self-initialization of the adaptor layer. The former finetunes the pretrained generative model (e.g., StyleGAN) on source and target data. The latter allows to initialize all non-pretrained network parameters without the need of any data. These techniques provide a better initialization for the I2I translation step. In addition, we introduce an auxiliary GAN that further facilitates the training of deep I2I systems even from small datasets. In extensive experiments on three datasets, (Animal faces, Birds, and Foods), we show that we outperform existing methods and that mFID improves on several datasets with over 25 points. Our code is available at: https://github.com/yaxingwang/TransferI2I.

1. Introduction

Image-to-image (I2I) translation aims to map an image from a source to a target domain. Several methods obtain outstanding results on paired data [21, 61], unpaired data [30, 56, 60], scalable I2I translation [11, 36, 45] and diverse I2I translation [11, 20, 32]. *Scalable I2I* translation aims to translate images between multiple domains. For example, a cat face is mapped onto other animal faces (i.e. dog, tiger, bear, etc.). The goal of *diverse I2I* translation is to synthesize multiple plausible outputs of the target domain from a single input image (i.e. translating a dog face to various plausible cat faces). Despite impressive leaps forward with paired, unpaired, scalable and diverse I2I translation, there are still important challenges. Specifically, to obtain good results existing works rely on large labelled data. When given small datasets (e.g., 10 images per domain) current algorithms suffer from inferior performance. Also, labeling large-scale datasets is costly and time-consuming, making those methods less applicable in practice.

Several works [5, 6, 12, 35, 38] have studied one-shot and few-shot I2I translation. One-shot I2I translation [5, 6, 12, 35] refers to the case where only *one source* image and *one or few* target images are available. These works fail to perform multiclass I2I translation. FUNIT [38] conducts few-shot I2I translation, but still requires large datasets at the training stage. In this paper, we focus on transfer learning for I2I translation with limited data.

Recent work [48, 54] leverages transfer learning for I2I translation. SGP [48] utilizes a pretrained classifier (e.g, VGG [50]) to initialize the encoder of an I2I model. However, the remaining networks (i.e., decoder, discriminator and adaptor layers[1]) need to be trained from scratch, which still requires a large dataset to train the I2I translation model. DeepI2I [54] uses a pretrained GAN (e.g., StyleGAN [26] and BigGAN [8]) to initialize the I2I model. However, it still requires to train the adaptor layers from scratch. The adaptor layers contains over 85M parameters (using the pretrained BigGAN) which makes their training on translation between small domains prone to overfitting. Since both SGP and DeepI2I leverage the adaptor between the encoder and the generator, one potential problem is that the generator easily uses the information from the high-resolution skip connections (connecting to the upper layers

[1]We follow [54] and call the layers which connect encoder and decoder at several levels adaptor layers.

of the generator), and ignore the deep layers of the generator, which require a more semantic understanding of the data, thus more difficult to train. Inspired by DeepI2I, we use the pretrained GANs to initialize I2I translation model. Differently, we propose a new method to train I2I translation, overcoming the overfitting and improving the training of I2I model.

In this paper, we decouple our learning process into two steps: image generation and I2I translation. The first step aims to train a better generative model, which is leveraged to initialize the I2I translation system, and contributes to improve I2I translation performance. We introduce two contributions to improve the efficiency of the transfer, especially important for small domains. (1) we improve *source-target initialization* by finetuning the pretrained generative model (e.g., StyleGAN) on source and target data. This ensures that networks are already better prepared for their intended task in the I2I system. (2) we propose a *self-initialization* to pretrain the weights of the adaptor networks (the module A in Figure 1 (b)) without the need of any data. Here, we exploit the fact that these parameters can be learned by generating the layer activations from both the generator and discriminator (by sampling from the latent variable z). From these activations the adaptor network weights can be learned. For the second step we conduct the actual I2I translation using the learned weights in the first step. Furthermore, we propose an *auxiliary generator* to encourage the usage of the deep layers of the I2I network.

Extensive experiments on a large variety of datasets confirms the superiority of the proposed transfer learning technique for I2I. It also shows that we can now obtain high-quality image on relatively small domains. This paper shows that transfer learning can reduce the need of data considerably; as such this paper opens up application of I2I to domains that suffer from data scarcity. Our main contributions are:

- We explore I2I translation with limited data, reducing the amount of required labeled data.
- We propose several novel techniques (i.e., *source-target initialization*, *self-initialization* and *auxiliary generator*) to facilitate this challenging setting.
- We extensively study the properties of the proposed approaches on two-class and multi-class I2I translation tasks and achieve significant performance improvements even for high quality images.

2. Related work

Generative adversarial networks. GANs [17] are a combination of a generator G and a discriminator D. The goal of the generator is to learn a mapping from a latent code, i.e. a noise source, to the training data distribution. Conversely, the discriminator network, or critic [3], learns to distinguish between the real data and generated instances from G in the fashion of an adaptive loss. In this opposed game, both networks improve upon each other to the point of yielding state-of-the-art image generation. Recent works [3, 18, 39] aim to overcome mode collapse and training instability problems, which frequently occur when optimizing GANs. Besides, several works [8, 13, 26] explore constructing effective architectures to synthesize high-quality images.

I2I translation. Image-to-image translation has been widely studied in computer vision. It has achieved outstanding performance on both paired [16, 22, 61] and unpaired image translation [30, 37, 40, 43, 56, 60]. These approaches, however, face two main challenges: diversity and scalability. The former aims to generate multiple plausible outputs of the target domain from a single input image [2, 20, 32, 57]. The goal of scalable I2I translation is to map images [11, 33, 57] across several domains using a single model. Several works [27, 46, 48, 55] explore the difficult task: the *shape* translation as well as *style*. TransGaGa [55] disentangles the source image into two spaces: the geometry and style. Then it conducts the translation for each latent space separately. However, none of these approaches addresses the problem of transfer learning for I2I.

Several recent works used GANs for I2I translation with few test samples. Lin *et al*. proposed a zero-shot I2I translation method, which leverages pairs of images and captions to study domain-specific and domain-invariant features. Recent work [5, 6, 12, 35] explore one-shot I2I translation, and propose one-shot specific I2I models. However, these methods cannot be used for multi-class I2I translation, since these model are designed for the two-class case where a few images of two domains can be accessed. FUNIT [38] is the first to study few-shot I2I translation, but still relies on vast quantities of labeled source domain images for training.

Transfer learning. Transfer learning aims to reduce the training time, improve the performance and reduce the amount of training data required by a model by reusing the knowledge from another model which has been trained on another, but related, task or domain. A series of recent works investigated knowledge transfer on generative models [42, 53] as well as discriminative models [14]. More recent work [48, 54] explore the knowledge transfer for I2I translation. Both methods, however, introduce a new network module which is trained from scratch, and prone to suffer from overfitting. Several other approaches [15, 23] perform the image manipulation based on the pretrained GAN. Especially, given the pretrained generator (e.g., StyleGAN) they expect to manipulate the output image attribute (e.g., the age, the hair style, the face pose etc..). However, these methods do not focus on transfer learning. Furthermore, some methods [1, 4, 59] embed a given exemplar image into the input latent space of the pretrained GAN (e.g., StyleGAN). These methods literately

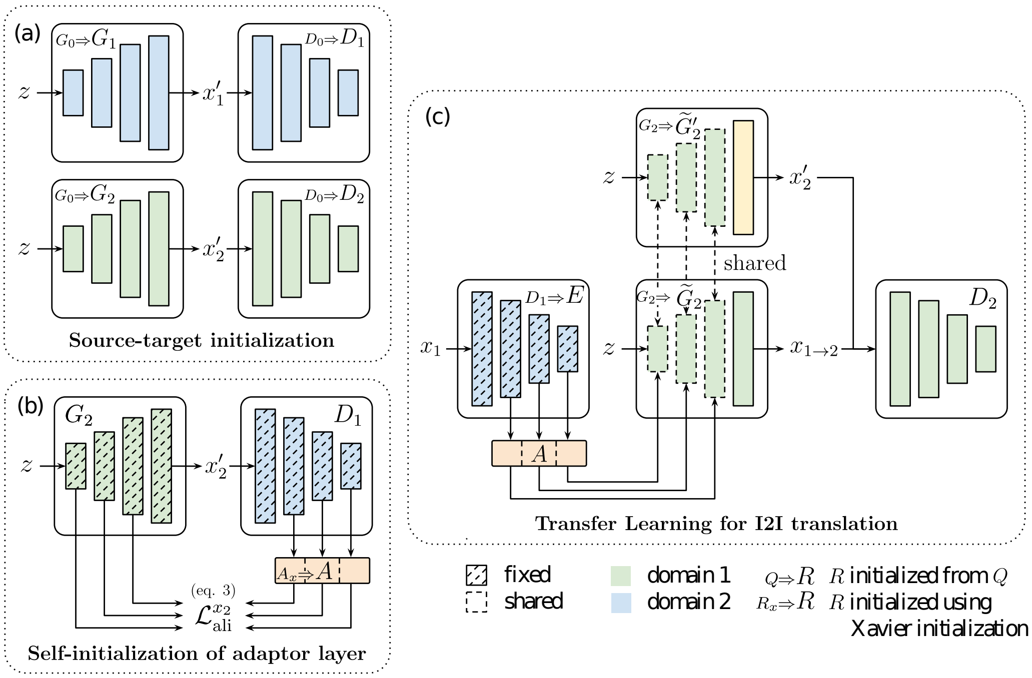

Figure 1. Model architecture and training stages. Here modules come from the immediate previous stage unless otherwise indicated. A pretrained GAN (e.g., StyleGAN [26]) is used as G_0 and D_0 to initialize the two GANs. (a) *Source-target initialization* performs finetuning on two domains (i.e., $\mathcal{X}_1$ and $\mathcal{X}_2$) to form two independent GANs (i.e., the generator G_1 and the discriminator D_1, the generator G_2 and the discriminator D_2). (b) *Self-initialization* of adaptor layer to pretrain the adaptor A and align both the generator G_2 and the discriminator D_1. We only update the adaptor layers A. (c) The I2I translation model is composed of five main parts: the encoder E, the adaptor layer A, the generator $\widetilde{G}_2$, the *auxiliary generator* $\widetilde{G}'_2$ and the discriminator D_2. Note the encoder E is initialized by the discriminator D_1. The portion of weights from $\widetilde{G}'_2$ that is not shared (in yellow), is initialized with G_2 weights.

optimize the latent space to reconstruct the provided image. In fact, they do not perform I2I translation.

3. Method

Problem setting. Our goal is to propose an unpaired I2I translation system for the case when training data is limited. Therefore, we apply transfer learning to improve the efficiency of I2I translation. We decouple our learning into an image generation step and a I2I translation step. The image generation step contains two contributions: (a) *source-target initialization*, and (b) adaptor layer *self-initialization*. In addition, for the I2I step, we introduce an *auxiliary generator* to address the inefficient usage of the deep layers of the generator when using the skip connections in I2I model.

Figure 1 provides an overview of our method. Figure 1(a) shows the *source-target initialization*, in which we learn a better generation model, thus contributing to a better initialization of the I2I translation models. Next, in Figure 1(b), we introduce *self-initialization* to overcome overfitting of the adaptor layers. In Figure1(c), we train the I2I network, and introduce an *auxiliary generator*. This additional generator shares several layers with the main generator, encouraging the usage of the under-performing deep layers. Our method is general for two-class I2I translation and multi-class I2I translation. First we introduce our method for single-class I2I (Section 3.1) and in the next section we extend our method to multi-class I2I (Section 3.2).

3.1. Method overview

We consider two domains: source domain $\mathcal{X}_1 \subset \mathbb{R}^{H\times W\times 3}$ and target domain $\mathcal{X}_2 \subset \mathbb{R}^{H\times W\times 3}$. In this work, given limited training samples from both source and target domains, we aim to map a source image $x_1 \in \mathcal{X}_1$ into a target sample $x_{1\to 2} \in \mathcal{X}_2$. Let vector $\mathbf{z} \in \mathbb{R}^{\mathbf{Z}}$ be random noise.

Source-target initialization. Given a pretrained GAN (e.g., StyleGAN) and limited training data, we propose to train two generative models on the limited available data of respectively the source and target domain (Figure 1 (a)). Especially, we train a generative model which is composed of a generator G_i and a discriminator D_i for each domain $\mathcal{X}_i, i = \{1, 2\}$. Here we apply finetuning to adapt to the source and target domains like in [53]. This step could be further improved by using recent approaches to improve the transfer on small domains [42, 41, 58, 34]. The training objective becomes:

$$\mathcal{L}_{GAN}^{x_1} = \mathbb{E}_{x_1\sim\mathcal{X}_1}\left[\log D_1\left(\mathbf{x_1}\right)\right] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log\left(1 - D_1\left(G_1\left(\mathbf{z}\right)\right)\right)\right] \tag{1}$$

$$\mathcal{L}_{GAN}^{x_2} = \mathbb{E}_{x_2\sim\mathcal{X}_2}\left[\log D_2\left(\mathbf{x_2}\right)\right] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log\left(1 - D_2\left(G_2\left(\mathbf{z}\right)\right)\right)\right]. \tag{2}$$

Here the generative models for both the source and target domains are used to provide a better initialization for the I2I translation.

Self-initialization of adaptor layer. Inspired by DeepI2I [54], we use the pretrained discriminator (Figure 1(a)) to initialize both the encoder and the discriminator of the I2I model (Figure 1(c)), and correspondingly, the pretrained generator to the I2I generator. Since in the GAN configuration there are no connections between intermediate layers of the generator and discriminator, these layers are not aligned. For that reason, [54] introduces an adaptor network (indicated by A in Figure 1(b,c)) to communicate between the various layers of the encoder and decoder. In DeepI2I, they found that the introduction of four adaptor networks is optimal. These layers contain a significant amount of the total number of parameters in the system (around 25%). They then proceed to naively optimize these parameters on the source-target data. Training the adaptor network from scratch leads to overfitting on limited data.

To overcome these drawbacks, we propose a procedure, called *self-initialization*, that leverages the previous pretrained model (Figure 1(a)) to align the adaptor networks without the need of any data. As shown in Figure 1(b), the noise $\mathbf{z}$ is taken as input for the generator G_2, from which we extract the hierarchical representation $F_g(\mathbf{z}) = \{G_2(\mathbf{z})_l\}$ as well as the synthesized image $G_2(\mathbf{z})$. Here $G_2(z)_l$ is the $l_{th}(l = m, \ldots, n, (n > m))$ ResBlock [2] output of the generator G_2. We then take the generated image $G_2(\mathbf{z})$ as input for the discriminator D_1, and similarly collect the hierarchical feature $F_d(\mathbf{z}) = \{D_1(G_2(\mathbf{z}))_l\}$. The adaptor network A finally takes the output representation $\{D_1(G(\mathbf{z}))_l\}$ as input, that is $A(F_d(\mathbf{z})) = \{A\}$. In this step, our loss is:

$$\mathcal{L}_{ali}^{x_2} = \sum_l \left\|F_g\left(\mathbf{z}\right) - A\left(D_1(G_2(\mathbf{z}))\right)\right\|_1. \tag{3}$$

In this step both the generator and the discriminator are frozen and only the adaptor layers are learned. Note that the adaptor layers are trained to take the discriminator as input, and output a representation that is aligned with the generator (opposite to the order in which they are applied in the GAN); this is done because the generator and discriminator are switched in the I2I network (see Figure 1(c)) when we use the pretrained discriminator to initialize the encoder.

Transfer Learning for I2I translation. Figure 1(c) shows how to map the image from the source domain to target domain. For example, to translate a source image $x_1 \in \mathcal{X}_1$ to $x_{1\to2} \in \mathcal{X}_2$. Our architecture consists of 5 modules: the encoder E, the adaptor A, the generator $\widetilde{G}_2$, the *auxiliary generator* $\widetilde{G}'_2$ and the discriminator D_2. Let E_l be the $l_{th}(l = m, \ldots, n, (n > m))$ ResBlock output of the encoder E, which is further taken as input for the corresponding adaptor network A_l.

We aim to map the image from the source to the target domain with limited labeled data. First, the encoder E, initialized by the pretrained discriminator D_1 takes the image x_1 as input, extracting the hierarchical representation $E_g(x_1) = \{E(x_1)_l\}$ from different layers, which contains both the structural and semantic information of the input image. $E_g(x_1)$ is then fed to the adaptor network $A(x_1) = \{A(x_1)_l\}$, which in turn is taken as input for the generator $\widetilde{G}_2$ along with the noise $\mathbf{z}$ to synthesize the output image $x_{1\to2} = \widetilde{G}_2(\mathbf{z}, A(E(x_1)))$. Note we sum the output of the adaptor with the corresponding one of the generator. We employ the discriminator D_2 to distinguish real images from generated images, and preserve a similar pose in input source image x_1 and the output $\widetilde{G}_2(\mathbf{z}, A(E(x_1)))$ [38, 54].

Training the I2I translation model can lead to unused capacity of the deep layers of the generator, largely due to the skip connections. It is relatively easy for the generator to use the information from the high-resolution skip connections (connecting to the upper layers of the generator), and ignore the deep layers of the generator, which require a more semantic understanding of the data, thus more difficult to train. To address this, we propose an *auxiliary generator* which has the same network design, but only uses the noise as input. Taking the translation from the source image $x_1 \in \mathcal{X}_1$ to $x_{1\to2} \in \mathcal{X}_2$ as example. The *auxiliary generator* G'_2 takes the noise $\mathbf{z}$ as input, and synthesizes the output image $x'_2 \in \mathcal{X}_2$. We propose to share the deep layers of this *auxiliary generator* with the ones following the skip connection in the main generator $\widetilde{G}_2$ (the dashed layers in Figure 1(c)). Since $\widetilde{G}'_2$ has no access to skip connections, it is forced to use its deep layers, and since we share these, the main I2I generator is also driven to use them.

Our loss function for I2I translation is a multi-task objective comprising: (a) *adversarial loss* which classifies the real image and the generated image. (b) *reconstruction loss* guarantees that both the input image x_1 and the synthesized image $x_{1\to2} = \widetilde{G}_2(\mathbf{z}, A(E(x_1)))$ keep the similar structural information.

Adversarial loss. We employ GAN [17] to optimize this problem as follows:

$$\begin{aligned}\mathcal{L}_{GAN}^{x_2} = \ & \mathbb{E}_{x_2\sim\mathcal{X}_2}\left[\log D_2\left(\mathbf{x_2}\right)\right] \\ & + \mathbb{E}_{\mathbf{x_1}\sim\mathcal{X}_1,\mathbf{z}\sim p(\mathbf{z})}\left[\log(1 - D_2\left(\widetilde{G}_2\left(A\left(E\left(\mathbf{x_1}\right)\right),\mathbf{z}\right)\right)\right] \\ & + \lambda_{aux}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log\left(1 - D_2\left(\widetilde{G}'_2\left(\mathbf{z}\right)\right)\right)\right],\end{aligned} \tag{4}$$

where $p\left(\mathbf{z}\right)$ follows the normal distribution. The hyper-

[2]After each ResBlock the feature resolution is half of the previous one in both encoder and discriminator, and two times in generator

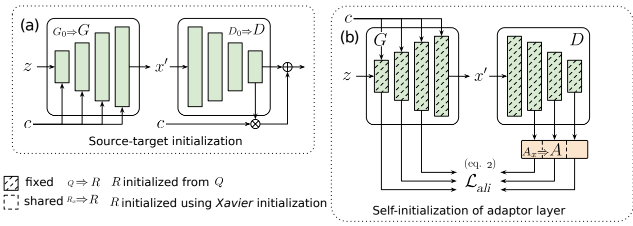

Figure 2. Conditional model architecture and training stages. c is the conditional embedding. (a) *Source-target initialization* and (b)*Self-initialization*.

parameter λ_{aux} is to balance the importance of each terms. We set $\lambda_{aux} = 0.01$. The discriminator D_1 and loss $\mathcal{L}_{GAN}^{x_1}$ are similar.

Reconstruction loss. We use reconstruction to preserve the structure of both the input image x_1 and the output image $x_{1\to2}$. In the same fashion as results for photo-realistic image generation [24, 25, 49], we use the discriminator output to achieve this goal through the following loss:

$$\mathcal{L}_{rec}^{x_1} = \sum_l \alpha_l \left\| D_2\left(\mathbf{x_1}\right) - D_2\left(\mathbf{x_{1\to2}}\right) \right\|_1, \tag{5}$$

where parameters α_l are scalars which balance the terms. Note we set $\alpha_l = 1$.

Full Objective. The full objective function of our model is:

$$\min_{\substack{E_1,E_2,A_1,A_2\\ \widetilde{G}_1,\widetilde{G}_1',\widetilde{G}_2,\widetilde{G}_2'}} \max_{D_1,D_2} \mathcal{L}_{GAN}^{x_1} + \mathcal{L}_{GAN}^{x_2} + \lambda_{rec}\left(\mathcal{L}_{rec}^{x_1} + \mathcal{L}_{rec}^{x_2}\right) \tag{6}$$

where λ_{rec} is hyper-parameters that balance the importance of each terms. We set $\lambda_{rec} = 1$. Note $\widetilde{G}_1$ and $\widetilde{G}_1'$ are the corresponding the generator and the auxiliary generator from domain $\mathcal{X}_2$ to domain $\mathcal{X}_1$.

3.2. Multi-class I2I translation

Our method can also be applied to multi-class I2I translation. As shown in Figure 2 we have one conditional generator and one conditional discriminator by using a class embedding like BigGAN [8]. Especially, we perform *source-target initialization* (Figure 2(a)) from the pretrained conditional GAN (e.g., BigGAN), and obtain a single generator and discriminator for all data. The following steps, the *self-initialization of the adaptor* (Figure 2(b)) and the I2I translation with the *auxiliary generator* (Figure 1(c)), are similar to the ones of two-class I2I system except for the conditioning for both generator and discriminator. The framework for multi-class I2I translation is shown in Supp. Mat. A.

4. Experiments

In this section, we first introduce the experimental settings (Section 4.1): the training details, evaluation measures, datasets and baselines. We then evaluate our method

source-target initialization	Self-initialization	mKID ×100↓	mFID ↓
×	×	11.48	137.11
√	×	9.63	114.23
×	√	10.03	122.12
√	√	9.40	109.7

Table 1. Influence of *source-target initialization* and *self-initialization* of the adaptor on *Animal faces*.

on two cases: *multi-class I2I translation* (Section 4.2) and *two-class I2I translation* (Section 4.3).

4.1. Experiment setting

Training details. We adapt the structure of the pretrained GAN (i.e., StyleGAN for two-class I2I translation and BigGAN for multi-class I2I translation) to our architecture. Especially, both the generator G and the discriminator D directly duplicate the ones of the GAN (i.e., StyleGAN or BigGAN). The auxiliary generator G' is same to the generator, and the encoder E copy the structure of the discriminator. The adaptor network A contains four sub-adaptor networks. In multi-class I2I system, each of the sub-adaptor consists of one Relu, two convolutional layers (Conv) with 3×3 filter size and stride of 1, and one Conv with 1×1 filter and stride of 1, except for the fourth sub-adaptor (corresponding to the deepest layer of the encoder) which only contains two Convs with 3×3 filter and stride of 1. In two-class I2I system, each sub-adaptor network is composed of Conv with 3×3 filter size and stride of 1. The proposed method is implemented in Pytorch [44]. The configure of the experiment is reported in Suppl. Mat. Section A (Table 1). We use 1× Quadro RTX 6000 GPUs (24 GB VRAM to conduct all our experiments.

Evaluation metrics. We use several GAN metrics. The first one is Fréchet Inception Distance (FID) [19], which compares the distributions of the real and fake images using the Fréchet distance. The second one is Kernel Inception Distance (KID) [7], which calculates the maximum mean discrepancy (MMD) of the same embedded features and is proven to be a converging estimator, contrary to FID. To account for all categories, we calculate the mean FID and KID as *mFID* and *mKID*. Finally, we train a real (*RC*) and a fake classifier (*FC*) [47] to evaluate the ability to generate class-specific images. RC is trained on real data and evaluated on the generated data and vice versa for FC.

Datasets. We evaluate our method on five datasets. For multi-class I2I translation, we use three datasets: *Animal faces* [38], *Birds* [51] and *Foods* [28]. To evaluate the two-class I2I model, we use two datasets: *cat2dog-200* [33] and *cat2dog-2035* [33]. The *Animal faces* dataset contains 1,490 images and 149 classes in total, *Birds* has 48,527 images and 555 classes in total, *Foods* consists of 31,395 images and 256 classes in total. We resized all images from the *Animal faces*, *Birds* and *Foods* to 128×128, and split each data into a training (90 %) and test set (10 %) except for the *Animal faces* in which the number of test images is

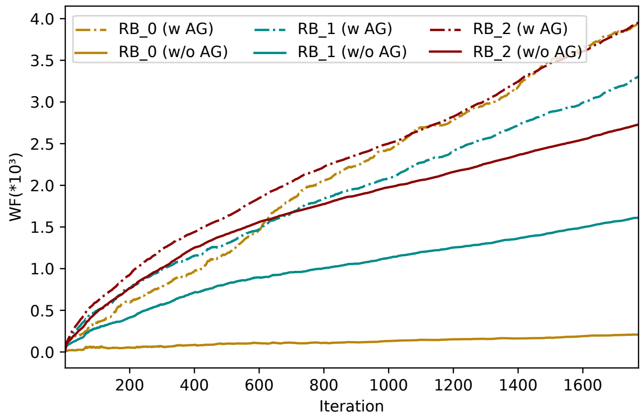

Figure 3. The change of the weights w (w/o) the *auxiliary generator* (AG). *RB_i* ($i = 0, 1, 2$) is the index of ResBlock layer of the generator from input to the output. *WF* is the weight fluctuation.

#ResBlock	mKID ×100↓	mFID ↓
1	9.48	115.47
2	9.39	110.31
3	9.34	105.98
4	9.25	103.55

Table 2. Ablation on number of the shared layers between the generator G and the *auxiliary generator* G'. Note we we account for the shared ResBlock layer from the bottom layer of the generator.

1,490 (10/per class). The *cat2dog-200* is composed of 200 images (100 images/per class). The *cat2dog-2035* contains 771 images for the cat category and 1264 images for the dog category. The test dataset for both *cat2dog-200* and *cat2dog-2035* is the same, and has 200 images (100 images/per class) with an image size of 256 × 256. Note for any used dataset, both training and test splits do not overlap.

The baselines for two-class I2I. We compare to several baselines for this setting. *CycleGAN [60]* first perform unpaired I2I translation by leveraging a cycle consistency loss to reconstruct the input image. *UNIT [37]* presents an unsupervised I2I translation method under the shared-latent space assumption. The related methods, including *MUNIT [20]*, *DRIT++* [33] and *StarGANv2* [11], propose disentanglement to control separately the pose and style information. *UGATIT [29]* aims to handle the geometric changes, and introduce two techniques: an attention module and a new normalization. CUT [43] introduces contrastive learning for I2I translation. DeepI2I [54] uses pretrained GANs to initialize the I2I model.

The baselines for multi-class I2I. We compare to StarGAN [10], StarGANv2 [11], SDIT [52], DRIT++ [33], DMIT [57] and DeepI2I [54], all of which perform image-to-image translation between multi-class domains. StarGANv2 [11] obtains the stability by introducing a class-specific network. *SDIT [52]* leverages the class label and random noise to achieve scalability and diversity in a single model. A similar idea is also explored in *DMIT [57]*.

4.2. Multi-class I2I translation

Ablation study. We now evaluate the effect of each independent contribution on the performance of TransferI2I. First we ablate the *source-target initialization* and *self-initialization* without the *auxiliary generator*. Next, we evaluate the performance gain when adding the *auxiliary generator*.

Source-target initialization and self-initialization. Table 1 reports the performance of both techniques in terms of both mFID and mKID on *Animal faces*. Note that the second row of Table. 1 is equal to DeepI2I. Adding one of the techniques (*source-target initialization* and *self-initialization* of the adaptor layer) improves performance of I2I translation compared to DeepI2I. Furthermore, performing *source-target initialization* achieves a larger advantage than *self-initialization*, *e.g.* for *mFID*: 114.23 vs. 122.12. This seems to indicate the former is more important. Finally, using both techniques obtains the best mFID score, indicating that our method successfully performs I2I translation with few images.

Auxiliary generator. In this paper, we propose to leverage the *auxiliary generator* to encourage the usage of the deep layers of the generator. We conduct an experiment to evaluate the effect of the number of shared layers between the generator $\widetilde{G}$ and the *auxiliary generator* $\widetilde{G}'$. As reported in Table. 2, we found that more shared layers result in better performance (e.g., the mFID value reduces with an increasing number of shared layers).

To measure the distance between two models we use weight fluctuation (WF), defined for two models with parameters θ_1 and θ_2 as $\mathrm{WF} = (\theta_1 - \theta_2)^T \mathrm{FM}_{\theta_1} (\theta_1 - \theta_2)$ where FM_{θ_1} is the Fisher matrix [31]. This distance takes into account the importance of the weights to the loss computation. As shown in Figure 3, using the *auxiliary generator* leads to larger weight changes in the deep layers than not using it, clearly demonstrating improved utilization and a beneficial effect in the overall system performance. The drastic change (i.e., *RB_0 (w Aug.) vs. RB_0 (w/o Aug.)*) appears in the first ResBlock of the generator, which means we are able to learn the semantic information. Moving towards the upper layers, the gap of the corresponding layers of the two generators (w and w/o the *auxiliary generator*) becomes smaller. The most likely reason is that the upper layers (influencing the structural information) use more information from the skip connections.

Quantitative results. As reported in Table 3, we compare the proposed method with the baselines on the *Animal faces* [38], *Birds* [51] and *Foods* [28] datasets. Our approach outperforms all baselines in terms of mFID/mKID (joint quality and diversity) and RC/FC (the ability to generate class-specific images). We obtain a drop in mFID of around 30 points for both *Animal faces* and *Birds*, and 23

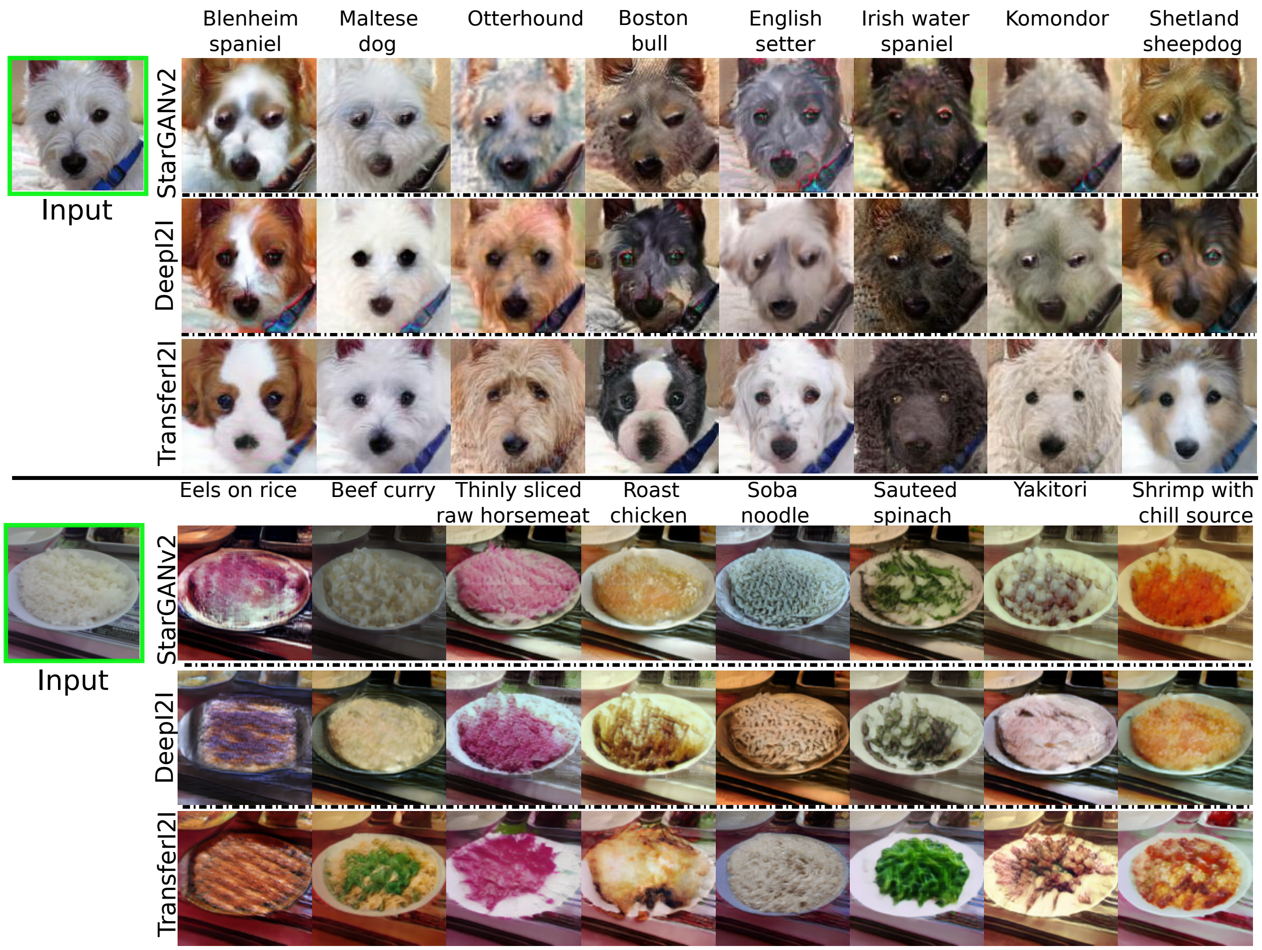

Figure 4. Qualitative comparison on the *Animal faces* and *Foods*. The input images are in the first column and the remaining columns show the class-specific translated images.

Datasets / Method	*Animal faces* (10/per class)				*Birds* (78/per class)				*Foods* (110/per class)			
	mKID×100↓	mFID↓	RC↑	FC↑	mKID×100↓	mFID↓	RC↑	FC↑	mKID×100 ↓	mFID↓	RC↑	FC↑
StarGAN [10]	28.4	276.5	4.89	5.12	21.4	214.6	9.61	10.2	20.9	210.7	10.7	12.1
SDIT [52]	31.4	283.6	5.51	4.64	22.7	223.5	8.90	8.71	23.7	236.2	11.9	11.8
DMIT [57]	29.6	280.1	5.98	5.11	23.5	230.4	12.9	11.4	19.5	201.4	8.30	10.4
DRIT++ [33]	26.6	270.1	4.81	6.15	24.1	246.2	11.8	13.2	19.1	198.5	10.7	12.7
StarGANv2 [11]	11.38	131.2	12.4	14.8	10.7	152.9	25.7	21.4	6.72	142.6	34.7	22.8
TransferI2I (scratch)	41.37	356.1	3.47	1.54	30.5	301.7	3.24	5.84	26.5	278.2	5.83	4.67
DeepI2I [54]	11.48	137.1	10.3	9.27	8.92	146.3	20.8	22.5	6.38	130.8	30.2	19.3
TransferI2I	**9.25**	**103.5**	**22.3**	**25.4**	**6.23**	**118.3**	**27.1**	**28.4**	**3.62**	**107.8**	**43.2**	**24.8**

Table 3. Comparison with baselines. TransferI2I obtains superior results on three datasets. We still obtain satisfactory advantage on both the bird and the food dataset, even they have more samples.

points on *Foods*. This indicates an advantage of the proposed method on small datasets (e.g., the number of images per class is 10 on *Animal faces*). The advantage is less on larger datasets (e.g. on the *Food* dataset with 110 images per class). Training the same architecture from scratch (*Transfer(scratch)*) obtains inferior results. Both StarGANv2 and DeepI2I exhibit similar performance, albeit inferior to TransferI2I on all metrics. Apart from the improvement on mFID, also the classification scores RC and FC of TransferI2I show that both quality (RC/FC) and diversity (FC) are improved.

We also evaluate the methods when using 100 images per class on the *Animal faces* dataset. We obtain the 149.4, 153.3 and 127.6 mFID for StarGANv2, DeepI2I and our model respectively. Our method still obtains a large advantage compared to its competitors even for a larger test set.

Qualitative results. Figure 4 shows the comparison to baselines on *Animal faces* and *Foods* dataset. Although both StarGANv2 and DeepI2I are able to perform multi-class I2I translation to each class, they fail to generate highly realistic images. Taking *Animal faces* as an example, given the target class label our method is able to provide high visual quality images. The qualitative results of the *Foods* dataset also confirm our conclusion: the images of TransferI2I are

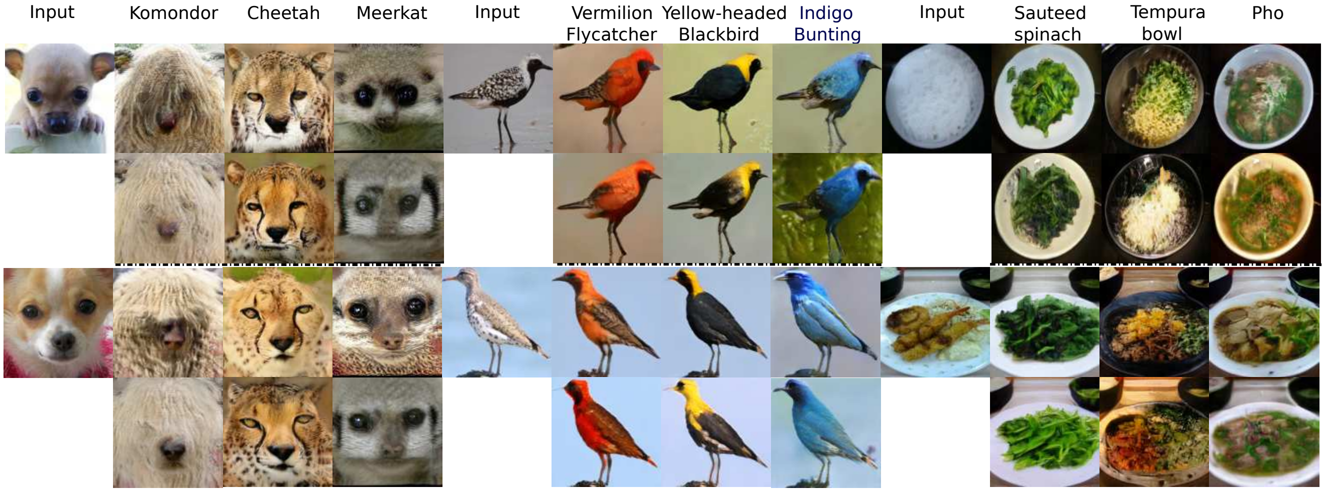

Figure 5. Qualitative results of TransferI2I. The input image is in the first column and the class-specific outputs are in the remaining columns. For each specific target class, we show two images.

Method \ Dataset	(cat,dog):(100,100)				(cat,dog):(771,1264)			
	dog → cat		cat → dog		dog → cat		cat → dog	
	FID↓	KID ↓	FID↓	KID ↓	FID↓	KID ↓	FID↓	KID ↓
CycleGAN [60]	210.7	14.33	284.6	28.14	119.32	4.93	125.30	6.93
UNIT [37]	189.4	12.29	266.3	25.51	59.56	1.94	63.78	1.94
MUNIT [20]	203.4	13.63	270.6	26.17	53.25	1.26	60.84	7.25
NICEGAN [9]	104.4	6.04	156.2	10.56	48.79	1.58	44.67	1.20
UGATIT-light [29]	133.3	6.51	206.6	15.04	80.70	3.22	64.36	2.49
CUT [43]	197.1	12.01	265.1	25.83	45.41	1.19	48.37	6.37
StarGANV2 [11]	336.4	40.21	339.8	41.32	**25.41**	0.91	**30.1**	**1.03**
DeepI2I [54]	83.71	4.26	112.4	5.67	43.23	1.37	39.54	1.04
TransferI2I (ours)	**55.2**	**3.97**	**83.6**	**4.59**	27.0	**0.84**	37.13	1.12

Table 4. The metric results on both *cat2dog-200* and *cat2dog-2035* datasets. Note we multiply 100 for *KID*.

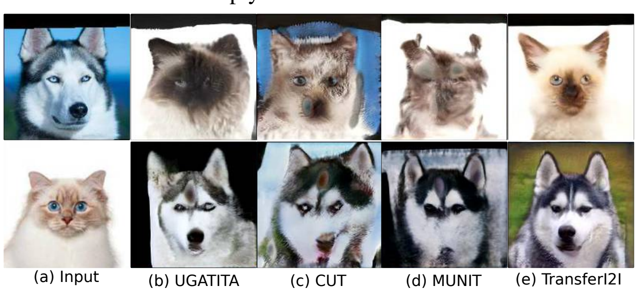

Figure 6. Examples of generated outputs on *cat2dog-200* dataset.

in general of higher quality than those of the baselines.

We further validate whether our method has both scalability and diversity in a single model. As shown in Figure 5, given the target class label (e.g., *Komondor*) our method successfully synthesizes diverse images by varying the noise z (i.e., the second column of the figure). The results show that by changing the target class label (i.e., scalability) the generator produces the corresponding target-specific output.

4.3. Two-class I2I translation

To evaluate the generality of our method, here we validate the proposed algorithm for two-class I2I translation on a two-category dataset: cats and dogs. We use the pre-trained StyleGAN to initialize our model (see Figure 1 (a)). Figure 6 shows the generated image of both the baselines and the proposed method on *cat2dog-200* dataset. We can easily observe that the baselines fail to synthesize realistic image, although they learn the style information of the target domain. We can see that TransferI2I generates more realistic target images. For quantitative evaluation, we report the results in terms of FID and KID. As shown in Table 4, TransferI2I obtains the best score on the small *cat2dog-200* dataset, improving FID by around 30 points with respect to DeepI2I. This clearly demonstrates that our method successfully conducts I2I translation when given limited data. On the much larger *cat2dog-2035* dataset, where transfer learning is less crucial, we obtain comparable performance to StarGANv2 but significantly outperform DeepI2I (which uses a similar architecture).

5. Conclusions

We have proposed an approach to benefit from transfer learning for image-to-image methods. We decoupled our learning process into an image generation and I2I translation step. The first step, including the *source-target initialization* and *self-initialization* of the adaptor, aims to learn a better initialization for the I2I translation (the second step). Furthermore, we introduce an *auxiliary generator* to overcome the inefficient usage of the deep layers of the generator. In this paper we still suffer from challenge that the domain gap between the target domain and the source domain influences the transfer effectiveness: the smaller the domain gap is, the better the I2I translation performance is. We will focus on this limimation in our future work.

Acknowledgement We acknowledge the support from Huawei Kirin Solution, and the project PID2019-104174GB-I00 (MINECO, Spain), the CERCA Programme of Generalitat de Catalunya and EU Project CybSpeed MSCA-RISE-2017-777720.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *ICCV*, pages 4432–4441, 2019. 2

[2] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *ICML*, 2018. 2

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. In *ICLR*, 2017. 2

[4] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *ICLR*, 2018. 2

[5] Sagie Benaim, Ron Mokady, Amit Bermano, and L Wolf. Structural analogy from a single image pair. In *Computer Graphics Forum*. Wiley Online Library. 1, 2

[6] Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In *NeurIPS*, pages 752–762, 2019. 1, 2

[7] M Bińkowski, DJ Sutherland, M Arbel, and A Gretton. Demystifying mmd gans. In *ICLR*, 2018. 5

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 1, 2, 5

[9] Runfa Chen, Wenbing Huang, Binghui Huang, Fuchun Sun, and Bin Fang. Reusing discriminators for encoding towards unsupervised image-to-image translation. In *CVPR*, 2020. 8

[10] 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 *CVPR*, June 2018. 6, 7

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 1, 2, 6, 7, 8

[12] Tomer Cohen and Lior Wolf. Bidirectional one-shot unsupervised domain mapping. In *ICCV*, pages 1784–1792, 2019. 1, 2

[13] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NeurIPS*, pages 1486–1494, 2015. 2

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014. 2

[15] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*, pages 5744–5753, 2019. 2

[16] Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *NeurIPS*, pages 1294–1305, 2018. 2

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. 2, 4

[18] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, pages 5767–5777, 2017. 2

[19] 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 *NeurIPS*, pages 6626–6637, 2017. 5

[20] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, pages 172–189, 2018. 1, 2, 6, 8

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 1

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[23] Ali Jahanian, Lucy Chai, and Phillip Isola. On the"steerability" of generative adversarial networks. In *ICLR*, 2020. 2

[24] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *NeurIPS*, pages 4016–4027, 2018. 5

[25] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, pages 1219–1228, 2018. 5

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 2, 3

[27] Oren Katzir, Dani Lischinski, and Daniel Cohen-Or. Cross-domain cascaded deep feature translation. *arXiv*, pages arXiv–1906, 2019. 2

[28] Yoshiyuki Kawano and Keiji Yanai. Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In *ECCV*, pages 3–17. Springer, 2014. 5, 6

[29] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019. 6, 8

[30] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 1, 2

[31] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 6

[32] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 1, 2

[33] 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. *IJCV*, pages 1–16, 2020. 2, 5, 6, 7

[34] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. In *NeurIPS*, 2020. 3

[35] Jianxin Lin, Yingxue Pang, Yingce Xia, Zhibo Chen, and Jiebo Luo. Tuigan: Learning versatile image-to-image translation with two unpaired images. In *European Conference on Computer Vision*, pages 18–35. Springer, 2020. 1, 2

[36] Alexander H Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *NeurIPS*, pages 2590–2599, 2018. 1

[37] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, pages 700–708, 2017. 2, 6, 8

[38] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *CVPR*, pages 10551–10560, 2019. 1, 2, 4, 5, 6

[39] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pages 2794–2802, 2017. 2

[40] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*, pages 3693–3703, 2018. 2

[41] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. In *CVPR AI for Content Creation Workshop*, 2020. 3

[42] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. *ICCV*, 2019. 2, 3

[43] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for conditional image synthesis. In *ECCV*, 2020. 2, 6, 8

[44] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[45] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. In *NeurIPS*, 2016. 1

[46] Kuniaki Saito, Kate Saenko, and Ming-Yu Liu. Coco-funit: Few-shot unsupervised image translation with a content conditioned style encoder. *arXiv preprint arXiv:2007.07431*, 2020. 2

[47] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. How good is my gan? In *ECCV*, pages 213–229, 2018. 5

[48] Assaf Shocher, Yossi Gandelsman, Inbar Mosseri, Michal Yarom, Michal Irani, William T Freeman, and Tali Dekel. Semantic pyramid for image generation. In *CVPR*, pages 7457–7466, 2020. 1, 2

[49] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, pages 2107–2116, 2017. 5

[50] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1

[51] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*, pages 595–604, 2015. 5, 6

[52] Yaxing Wang, Abel Gonzalez-Garcia, Joost van de Weijer, and Luis Herranz. SDIT: Scalable and diverse cross-domain image translation. In *ACM MM*, 2019. 6, 7

[53] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *ECCV*, pages 218–234, 2018. 2, 3

[54] Yaxing Wang, Lu Yu, and Joost van de Weijer. Deepi2i: Enabling deep hierarchical image-to-image translation by transferring from gans. *NeurIPS*, 2020. 1, 2, 4, 6, 7, 8

[55] Wayne Wu, Kaidi Cao, Cheng Li, Chen Qian, and Chen Change Loy. Transgaga: Geometry-aware unsupervised image-to-image translation. In *CVPR*, pages 8012–8021, 2019. 2

[56] Zili Yi, Hao Zhang, Ping Tan Gong, et al. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2017. 1, 2

[57] Xiaoming Yu, Yuanqi Chen, Shan Liu, Thomas Li, and Ge Li. Multi-mapping image-to-image translation via learning disentanglement. In *NeurIPS*, pages 2990–2999, 2019. 2, 6, 7

[58] Miaoyun Zhao, Yulai Cong, and Lawrence Carin. On leveraging pretrained gans for limited-data generation. *ICML*, 2020. 3

[59] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. *ECCV*, 2020. 2

[60] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 1, 2, 6, 8

[61] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, pages 465–476, 2017. 1, 2