DeltaGAN: Towards Diverse Few-shot Image Generation with Sample-Specific Delta

Yan Hong, Li Niu* , Jianfu Zhang, and Liqing Zhang

MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, China
yanhong.sjtu@gmail.com, {ustcnewly,c.sis}@sjtu.edu.cn,
zhang-lq@cs.sjtu.edu.cn

Abstract. Learning to generate new images for a novel category based on only a few images, named as few-shot image generation, has attracted increasing research interest. Several state-of-the-art works have yielded impressive results, but the diversity is still limited. In this work, we propose a novel Delta Generative Adversarial Network (DeltaGAN), which consists of a reconstruction subnetwork and a generation subnetwork. The reconstruction subnetwork captures intra-category transformation, *i.e.*, "delta", between same-category pairs. The generation subnetwork generates sample-specific "delta" for an input image, which is combined with this input image to generate a new image within the same category. Besides, an adversarial delta matching loss is designed to link the above two subnetworks together. Extensive experiments on six benchmark datasets demonstrate the effectiveness of our proposed method. Our code is available at https://github.com/bcmi/DeltaGAN-Few-Shot-Image-Generation.

1 Introduction

With the great success of deep learning, existing deep image generation models [32,33,5,45,46,6,18,10,28,43] based on Variational Auto-Encoder (VAE) [35] or Generative Adversarial Network (GAN) [22] have made a significant leap forward for generating diverse and realistic images for a given category. These methods generally require amounts of training images to generate new images for a given category. For the long-tail or newly emerging categories with only a few images, directly training or finetuning on limited data may cause overfitting issue [71,19]. Besides, it is very tedious to finetune the model for each unseen category. Therefore, given a few images from an unseen category, it is necessary to consider how to generate new realistic and diverse images for this category instantly. This task is called few-shot image generation in previous literature [4,2,29,30]. In this paper, following [4,2,29,30], we target at achieving instant adaptation from multiple seen categories to unseen categories without finetuning as shown in Fig. 1, which can benefit a lot of downstream tasks like low-data classification and few-shot classification.

* Corresponding author.

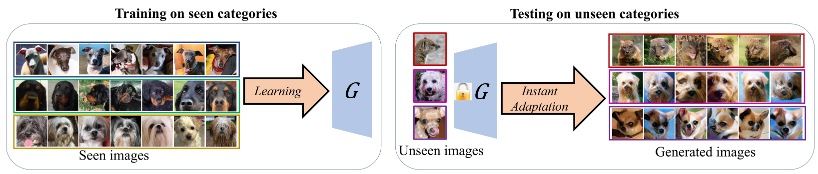

Fig. 1. The illustration of few-shot image generation task. We train a generative model on multiple seen categories. The learned generative model can be instantly applied to generate new images for unseen categories at test time. Each color indicates one category

The abovementioned few-shot image generation methods [4,2,29,30] resort to seen categories with sufficient training images to train a generative model, which can be used to generate new images for an unseen category with only a few images, which are dubbed as conditional images. For brevity, we refer to the images from seen (*resp.*, unseen) categories as seen (*resp.*, unseen) images. We classify the few-shot image generation methods into fusion-based methods [4,29,30,23] and transformation-based method [2]. However, those fusion-based methods can only produce images similar to conditional images and cannot be applied to one-shot image generation. Although transformation-based method could produce new images based on one conditional image, however, it fails to produce diverse images.

Following the research line of transformation-based methods, we propose a novel Delta Generative Adversarial Network (DeltaGAN), which can generate new images based on one conditional image by sampling random vectors. Our DeltaGAN is inspired by few-shot feature generation method Delta-encoder [57], in which intra-category transformation (*i.e.*, the difference between two images within the same category) is called "delta". The main idea of Delta-encoder is shown in Fig. 2(a). In the training stage, Delta-encoder learns to extract delta $\boldsymbol{\Delta}^r$ from same-category feature pair $\{\boldsymbol{f}_{x_1}, \boldsymbol{f}_{x_2}\}$ of image pair $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ from seen categories, in which $\boldsymbol{\Delta}^r$ is the additional information required to reconstruct $\boldsymbol{f}_{x_2}$ from $\boldsymbol{f}_{x_1}$. We refer to $\boldsymbol{x}_1$ as conditional (source) sample and $\boldsymbol{x}_2$ as target sample. In the testing stage, these extracted deltas are applied to a conditional feature $\boldsymbol{f}_y$ of image $\boldsymbol{y}$ from an unseen category to generate new feature $\tilde{\boldsymbol{f}}_y$ for this unseen category. However, Delta-encoder is a few-shot feature generation method, which cannot be directly applied to image generation. Besides, Delta-encoder relies on the deltas extracted from same-category training pairs, which does not support stochastic sampling (*i.e.*, sampling random vectors) to generate new samples in the testing stage.

In this paper, we aim to extend Delta-encoder to few-shot image generation method DeltaGAN, which supports producing diverse deltas based on random vectors. In this way, we can sample random vectors to generate diverse images without reaching training data in the testing stage. Considering that the plausibility of delta may depend on the conditional image [1], that is, a plausible

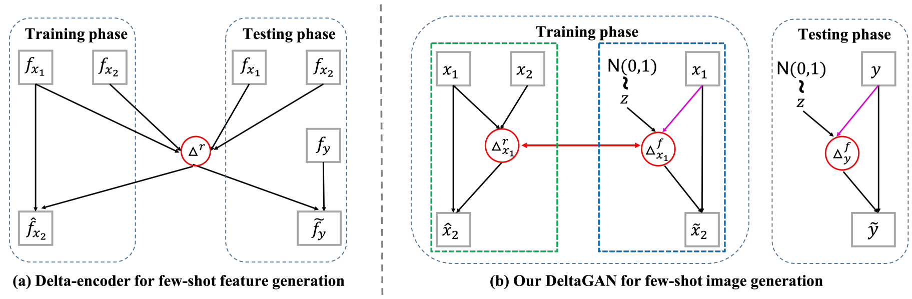

Fig. 2. The illustration of evolving from Delta-encoder to our DeltaGAN. $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ (*resp.*, $\{\boldsymbol{f}_{x_1}, \boldsymbol{f}_{x_2}\}$) is a same-category seen image pair (*resp.*, feature pair). $\boldsymbol{y}$ (*resp.*, $\boldsymbol{f}_y$) is a conditional image (*resp.*, feature) from an unseen category. $\{\hat{\boldsymbol{x}}_2, \tilde{\boldsymbol{x}}_2, \tilde{\boldsymbol{y}}\}$ (*resp.*, $\{\hat{\boldsymbol{f}}_{x_2}, \tilde{\boldsymbol{f}}_y\}$) are generated images (*resp.*, features). $\boldsymbol{z}$ is a random vector. $\{\boldsymbol{\Delta}^r, \boldsymbol{\Delta}^r_{x_1}\}$ (*resp.*, $\{\boldsymbol{\Delta}^f_{x_1}, \boldsymbol{\Delta}^f_y\}$) means real (*resp.*, fake) deltas. Red arrows indicate using adversarial delta matching loss to bridge the gap between real and fake delta. In (b), the green (*resp.*, blue) box encloses the reconstruction (*resp.*, generation) subnetwork, and pink arrows indicate the process of generating sample-specific delta

delta for one conditional image may be unsuitable for another conditional image (see Section 4.3), we aim to produce sample-specific delta. In particular, we take in a random vector and a conditional image to generate sample-specific delta, which represents the transformation from this conditional image to another possible image from the same category. We conjecture that the ability of generating sample-specific delta can be transferred from seen categories to unseen categories. To this end, we develop our DeltaGAN according to Fig. 2(b). In the training phase, we use a reconstruction subnetwork to reconstruct $\boldsymbol{x}_2$ from $\boldsymbol{x}_1$ with the delta $\boldsymbol{\Delta}^r_{x_1}$ (real delta) extracted from $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. We also use a generation subnetwork to generate sample-specific delta $\boldsymbol{\Delta}^f_{x_1}$ (fake delta) and produce new image $\tilde{\boldsymbol{x}}_2$. To ensure that fake deltas function similarly to real ones, we introduce a novel adversarial delta matching loss by using a delta matching discriminator to judge whether an input-output image pair matches the corresponding delta. Besides, we employ a variant of mode seeking loss [44] to alleviate the mode collapse issue. We also employ typical adversarial loss and classification loss to make the generated images realistic and category-preserving. In the testing stage, given a conditional unseen image $\boldsymbol{y}$, we can obtain its sample-specific delta $\boldsymbol{\Delta}^f_y$ by sampling random vector $\boldsymbol{z}$ for producing new image $\tilde{\boldsymbol{y}}$ from the category of $\boldsymbol{y}$. Because each delta represents one possible intra-category transformation, given a conditional unseen image, different deltas can produce realistic and diverse images from the same unseen category. Extensive experiments on six benchmark datasets demonstrate the effectiveness of our proposed method. Our contributions can be summarized as follows:

– We propose a novel delta-based few-shot image generation method, which has never been explored before.

- Technically, we extend few-shot feature generation method Delta-encoder to few-shot image generation with stochastic sampling and sample-specific delta. We also design a novel adversarial delta matching loss.
- Our method can produce diverse and realistic images for each unseen category based on a single conditional image, surpassing existing few-shot image generation methods by a large margin.

2 Related Work

Data augmentation: Data augmentation targets at augmenting training data with new samples. Traditional data augmentation tricks (*e.g.*, crop, flip, color jittering) only have limited diversity. Also, there are some methods [15,39,27,60] proposed to learn optimal augmentation strategies to improve the accuracy of classifiers. Similarly, neural augmentation [51,53,31,70,7] allowed a network to learn augmentations. As another research line, deep generative models can generate more diverse samples to augment training data, which can be categorized into feature-based augmentation methods [2] and image-based augmentation methods [57]. Feature-based augmentation methods [12,24] focused on generating more diverse deep features to augment the feature space of training data, while image-based augmentation methods [11,62,29,30] targeted at exploiting the distribution of training images and generating more diverse images.
Few-shot feature generation In existing few-shot feature generation methods, the semantic knowledge learned from the seen categories is transferred to compensate unseen categories in [17,24]. cCov-GAN [21] proposed a covariance-preserving adversarial augmentation network to generate more features for unseen categories. In [66], a generator subnetwork was added to a classification network to generate new examples. Intra-category diversity learned from seen categories was transferred to unseen categories to generate new features in [57,40]. Dual TriNet [12] proposed to synthesize instance features by leveraging semantics using a novel auto-encoder network for unseen categories. DTN [9] learned to transfer latent diversities from seen categories and composite them with support features to generate diverse features for unseen categories.
Few-shot image generation Compared with few-shot feature generation, few-shot image generation is a more challenging problem. Early methods can only be applied to generate new images for simple concepts, such as Bayesian program learning in [36], Bayesian reasoning in [55], and neural attention in [54].

Recently, several more advanced methods have been proposed to generate new real-world images in few-shot setting. To name a few, fusion-based method GMN [4] (*resp.*, MatchingGAN [29]) combined Matching Network [64] with Variational Auto-Encoder [52] (*resp.*, Generative Adversarial Network [22]) to generate new images without finetuning in the test phase. F2GAN [30] was designed to enhance the fusion ability of model by filling the details borrowed from conditional images. Transformation-based method DAGAN [2] proposed to produce new images by injecting random vectors into the generator conditioned on a single image. Apart from fusion-based and transformation-based methods, there

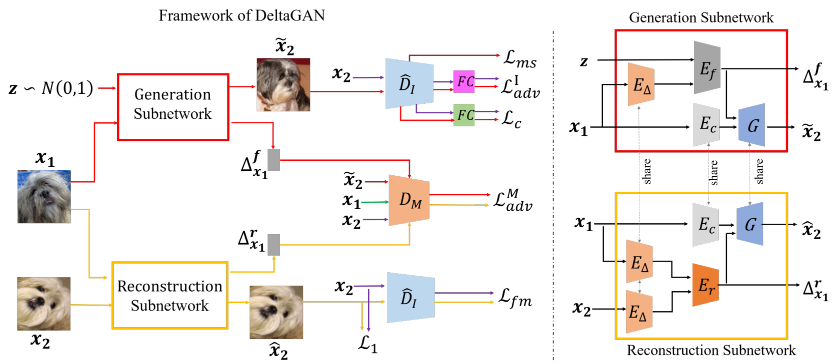

Fig. 3. Our DeltaGAN mainly consists of a reconstruction subnetwork and a generation subnetwork. Generation subnetwork learns to generate new image $\tilde{\boldsymbol{x}}_2$ based on conditional image $\boldsymbol{x}_1$ and random vector $\boldsymbol{z}$. Reconstruction subnetwork learns to produce reconstructed target image $\hat{\boldsymbol{x}}_2$ based on image pair $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. Best viewed in color

also exist optimization-based methods. For example, FIGR [13] (*resp.*, DAWSON [38]) combined adversarial learning with meta-learning method Reptile [48] (*resp.*, MAML [20]) to generate new images. However, they need to fine-tune the trained model with unseen category. Moreover, they can hardly produce sharp and realistic images. In this work, we propose a new transformation-based few-shot image generation method, which can produce more diverse images than previous methods based on a single image.

Note that some more recent works [50,37,56,65] are also called few-shot image generation. However, these works focus on adapting the generative model pretrained on a large dataset to a small dataset with a few examples, whose setting is quite different from ours. Firstly, these methods target at adapting from one source domain to another target domain, whereas our method adapts from multiple seen categories to unseen categories. Secondly, the models of these works need to be finetuned for each unseen domain, which is very tedious. Instead, the model of our method can be instantly applied to unseen categories without finetuning.

3 Our Method

We split all categories into seen categories and unseen categories, which have no overlap. Our DeltaGAN mainly consists of a reconstruction subnetwork and a generation subnetwork as shown in Fig. 3. The detailed architecture of each encoder/decoder is reported in Supplementary. In the training stage, given a same-category seen image pair $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ where $\boldsymbol{x}_1$ is the conditional image and $\boldsymbol{x}_2$ is the target image, the reconstruction subnetwork extracts real delta $\boldsymbol{\Delta}^{\boldsymbol{r}}_{\boldsymbol{x}_1}$ from this pair, and reconstructs the target image $\boldsymbol{x}_2$ based on $\boldsymbol{x}_1$ and $\boldsymbol{\Delta}^{\boldsymbol{r}}_{\boldsymbol{x}_1}$. In the generation subnetwork, a random vector $\boldsymbol{z}$ and the conditional image $\boldsymbol{x}_1$ are used to obtain fake sample-specific delta $\boldsymbol{\Delta}^{\boldsymbol{f}}_{\boldsymbol{x_1}}$, which collaborates with $\boldsymbol{x}_1$ to generate a new image $\tilde{\boldsymbol{x}}_2$. Moreover, we design an adversarial delta matching loss to bridge the gap between real delta and fake delta. In the testing stage, given an unseen image $\boldsymbol{y}$, only generation subnetwork is used to produce diverse and realistic images $\{\tilde{\boldsymbol{y}}_k\}$ belonging to the same category of $\boldsymbol{y}$.

3.1 Reconstruction Subnetwork

In the reconstruction subnetwork (see Fig. 3), there are three encoders E_Δ, E_c, E_r and a decoder G. Given a same-category seen image pair $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$, we use E_Δ to extract paired features $\{E_\Delta(\boldsymbol{x}_1), E_\Delta(\boldsymbol{x}_2)\} \in \mathcal{R}^{W \times H \times C}$, where $W \times H$ denotes the feature map size and C denotes the channel number. Then, we calculate the difference between $E_\Delta(\boldsymbol{x}_2)$ and $E_\Delta(\boldsymbol{x}_1)$, which is fed into E_r to obtain real delta $\boldsymbol{\Delta}^r_{x_1} \in \mathcal{R}^{W \times H \times C}$:

$$\boldsymbol{\Delta}^r_{x_1} = E_r(E_\Delta(\boldsymbol{x}_2) - E_\Delta(\boldsymbol{x}_1)), \tag{1}$$

where $\boldsymbol{\Delta}^r_{x_1}$ contains the additional information needed to reconstruct $\boldsymbol{x}_2$ from $\boldsymbol{x}_1$. We do not restrict our delta features to be linear offsets, which enables the delta features to learn more complex transformations. Then, $\boldsymbol{\Delta}^r_{x_1}$ is concatenated with $E_c(\boldsymbol{x}_1) \in \mathcal{R}^{W \times H \times C}$ and fed into G to obtain the reconstructed image $\hat{\boldsymbol{x}}_2$:

$$\hat{\boldsymbol{x}}_2 = G(\boldsymbol{\Delta}^r_{x_1}, E_c(\boldsymbol{x}_1)). \tag{2}$$

We employ a reconstruction loss $\mathcal{L}_1$ to ensure that $\hat{\boldsymbol{x}}_2$ is close to $\boldsymbol{x_2}$:

$$\mathcal{L}_1 = ||\hat{\boldsymbol{x}}_2 - \boldsymbol{x}_2||_1. \tag{3}$$

Considering the instability issue of early training stage, we use a feature matching loss [3] by matching the discriminative feature of $\hat{\boldsymbol{x}}_2$ with that of $\boldsymbol{x}_2$. In detail, we use a feature extractor $\hat{D}_I$ to extract the discriminative features of $\hat{\boldsymbol{x}}_2$ and $\boldsymbol{x}_2$ in each layer to calculate the feature matching loss:

$$\mathcal{L}_{fm} = \frac{1}{L} \sum_{l=1}^{L} ||\hat{D}^l_I(\boldsymbol{x}_2) - \hat{D}^l_I(\hat{\boldsymbol{x}}_2)||_1, \tag{4}$$

where L is the layer number of $\hat{D}_I$.

To support stochastic sampling for generation, we design another generation subnetwork in parallel with the reconstruction subnetwork (see Fig. 3). Two subnetworks share two encoders E_Δ, E_c and the decoder G. Besides, a new encoder E_f is introduced to obtain fake sample-specific delta. In our generation subnetwork, we concatenate a random vector $\boldsymbol{z}$ sampled from unit Gaussian distribution and the feature of conditional image $E_\Delta(\boldsymbol{x}_1) \in \mathcal{R}^{W \times H \times C}$, which is fed into E_f to obtain sample-specific delta $\boldsymbol{\Delta}^f_{x_1} \in \mathcal{R}^{W \times H \times C}$:

$$\boldsymbol{\Delta}^f_{x_1} = E_f(\boldsymbol{z}, E_\Delta(\boldsymbol{x}_1)), \tag{5}$$

where $\boldsymbol{\Delta}^f_{x_1}$ contains the additional information needed to transform conditional image $\boldsymbol{x}_1$ to another possible image within the same category. Then, analogous to the reconstruction subnetwork, $\boldsymbol{\Delta}^f_{x_1}$ is concatenated with $E_c(\boldsymbol{x}_1)$ and fed into G to produce a new image $\tilde{\boldsymbol{x}}_2$ belonging to the category of $\boldsymbol{x}_1$:

$$\tilde{\boldsymbol{x}}_2 = G(\boldsymbol{\Delta}^f_{x_1}, E_c(\boldsymbol{x}_1)), \tag{6}$$

in which $\tilde{\boldsymbol{x}}_2$ is the transformed result after applying delta $\boldsymbol{\Delta}^f_{x_1}$ to $\boldsymbol{x}_1$.

3.2 Generation Subnetwork

Adversarial loss: To make the generated image $\tilde{\boldsymbol{x}}_2$ close to real images, we employ a standard adversarial loss using the discriminator D_I. D_I contains the feature extractor $\hat{D}_I$ mentioned in Section 3.1 and a fully-connected (FC) layer. We adopt the hinge adversarial loss proposed in [47]:

$$\begin{aligned}\mathcal{L}^I_{adv,D} &= \mathbb{E}_{\boldsymbol{x}_2}[\max(0, 1-\mathrm{D}_I(\boldsymbol{x}_2))] + \mathbb{E}_{\tilde{\boldsymbol{x}}_2}[\max(0, 1+\mathrm{D}_I(\tilde{\boldsymbol{x}}_2))],\\ \mathcal{L}^I_{adv,G} &= -\mathbb{E}_{\tilde{\boldsymbol{x}}_2}[\mathrm{D}_I(\tilde{\boldsymbol{x}}_2)].\end{aligned} \tag{7}$$

The discriminator D_I tends to distinguish fake images from real images by minimizing $\mathcal{L}^I_{adv,D}$, while the generator tends to generate realistic images to fool the discriminator by minimizing $\mathcal{L}^I_{adv,G}$.

Classification loss: To ensure that $\tilde{\boldsymbol{x}}_2$ belongs to the expected category, we construct a classifier by replacing the last FC layer of D_I with another FC layer (the number of outputs is the number of seen categories). Then, the images from different categories can be distinguished by a cross-entropy classification loss:

$$\mathcal{L}_c = -\log p(c(\boldsymbol{x})|\boldsymbol{x}), \tag{8}$$

where $c(\boldsymbol{x})$ is the category label of $\boldsymbol{x}$. We train the classifier by minimizing $\mathcal{L}_{c,D} = -\log p(c(\boldsymbol{x}_2)|\boldsymbol{x}_2)$ of the target image $\boldsymbol{x}_2$. We also expect the generated image $\tilde{\boldsymbol{x}}_2$ to be classified as the same category of target image $\boldsymbol{x}_2$. Thus, we minimize $\mathcal{L}_{c,G} = -\log p(c(\boldsymbol{x}_2)|\tilde{\boldsymbol{x}}_2)$ when updating the generator.

Adversarial delta matching loss: To ensure that the generated sample-specific deltas function similarly to real deltas and encode the intra-category transformation, we design a novel adversarial delta matching loss to bridge the gap between real deltas and fake deltas. This goal is accomplished by a delta matching discriminator D_M, which takes a triplet (conditional image, output image, the delta between them) as input as shown in Fig. 3. Our delta matching discriminator D_M is constructed by feature extractor $\hat{\mathrm{D}}_I$ and four FC layers following global average pooling. In delta matching discriminator D_M, we extract the features of paired images $\{\hat{\mathrm{D}}_I(\boldsymbol{x}_1), \hat{\mathrm{D}}_I(\boldsymbol{x}_2)\}$ (*resp.*, $\{\hat{\mathrm{D}}_I(\boldsymbol{x}_1), \hat{\mathrm{D}}_I(\tilde{\boldsymbol{x}}_2)\}$), which are concatenated with sample-specific delta $\boldsymbol{\Delta}^r_{x_1}$ (*resp.*, $\boldsymbol{\Delta}^f_{x_1}$) to form a real (*resp.*, fake) triplet. Then, the real triplet and fake triplet are fed into the four FC layers to judge whether this conditional-output image pair matches the corresponding delta, in other words, whether the delta is the additional information required to transform the conditional image to the output image. In adversarial learning, the discriminator D_M needs to distinguish the real triplet $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\Delta}^r_{x_1}\}$ from the fake triplet $\{\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_2, \boldsymbol{\Delta}^f_{x_1}\}$, while the generator aims to synthesize realistic fake triplet to fool the discriminator. The delta matching adversarial loss is also in the form of hinge adversarial loss [47], which can be written as

$$\begin{aligned}\mathcal{L}^M_{adv,D} =& \mathbb{E}_{\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{\Delta}^r_{x_1}}[\max(0, 1-\mathrm{D}_M(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\Delta}^r_{x_1}))] \\ &+\mathbb{E}_{\boldsymbol{x}_1,\tilde{\boldsymbol{x}}_2,\boldsymbol{\Delta}^f_{x_1}}[\max(0, 1+\mathrm{D}_M(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_2, \boldsymbol{\Delta}^f_{x_1}))], \\ \mathcal{L}^M_{adv,G} =& -\mathbb{E}_{\boldsymbol{x}_1,\tilde{\boldsymbol{x}}_2,\boldsymbol{\Delta}^f_{x_1}}[\mathrm{D}_M(\boldsymbol{x}_1, \tilde{\boldsymbol{x}}_2, \boldsymbol{\Delta}^f_{x_1})],\end{aligned} \tag{9}$$

where $\mathcal{L}^M_{adv,D}$ (*resp.*, $\mathcal{L}^M_{adv,G}$) is optimized for updating $\{\hat{\mathrm{D}}_I, \mathrm{D}_M\}$ (*resp.*, the generator).

Mode seeking loss: We observe that by varying random vector $\boldsymbol{z}$, the generated images may collapse into a few modes, which is referred to as mode collapse [44]. Therefore, we use a variant of mode seeking loss [44] to seek for more modes to enhance the diversity of generated images. Different from [44], we apply mode seeking loss to multi-layer features extracted by $\hat{D}_I$. In particular, we minimize the ratio of the distance between $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ over the distance between $\hat{D}^l_I(\tilde{\boldsymbol{x}}^1_2)$ and $\hat{D}^l_I(\tilde{\boldsymbol{x}}^2_2)$ at the l-th layer of $\hat{D}_I$:

$$\mathcal{L}_{ms} = \frac{1}{L}\sum_{l=1}^{L}\frac{||\boldsymbol{z}_1 - \boldsymbol{z}_2||_1}{||\hat{D}^l_I(\tilde{\boldsymbol{x}}^1_2) - \hat{D}^l_I(\tilde{\boldsymbol{x}}^2_2)||_1}. \tag{10}$$

Intuitively, when $||\boldsymbol{z}_1 - \boldsymbol{z}_2||_1$ is large, we expect $\hat{D}^l_I(\tilde{\boldsymbol{x}}^1_2)$ and $\hat{D}^l_I(\tilde{\boldsymbol{x}}^2_2)$ to be considerably different, which can push the generator to search more modes to produce diverse images. In our experiments (see Section 4.3), we find that mode seeking loss is critical for diversity. However, without the guidance of reconstruction subnetwork and adversarial delta matching loss, solely using mode seeking loss cannot generate meaningful deltas, with both diversity and realism significantly downgraded.

3.3 Optimization

We use θ_G to denote the model parameters of $\{E_\Delta, E_r, E_c, E_f, G\}$, while θ_D is used to denote the model parameters of $\{D_I, D_M\}$. The total loss function of our method can be written as

$$\mathcal{L} = \mathcal{L}^I_{adv} + \mathcal{L}^M_{adv} + \lambda_1\mathcal{L}_1 + \mathcal{L}_c + \lambda_{fm}\mathcal{L}_{fm} + \lambda_{ms}\mathcal{L}_{ms}, \tag{11}$$

in which λ_1, λ_{fm}, and λ_{ms} are trade-off parameters. $\mathcal{L}^I_{adv}$ represents $\mathcal{L}^I_{adv,G}$ (*resp.*, $\mathcal{L}^I_{adv,D}$) when updating the model parameters θ_G (*resp.*, θ_D). Similarly, $\mathcal{L}^M_{adv}$ represents $\mathcal{L}^M_{adv,G}$ (*resp.*, $\mathcal{L}^M_{adv,D}$) when updating the model parameters θ_G (*resp.*, θ_D).

θ_G and θ_D are optimized using related loss terms in an alternating fashion. In particular, θ_D is optimized by minimizing $\mathcal{L}^I_{adv,D} + \mathcal{L}^M_{adv,D} + \mathcal{L}_{c,D}$. θ_G is optimized by minimizing $\mathcal{L}^I_{adv,G} + \mathcal{L}^M_{adv,G} + \lambda_1\mathcal{L}_1 + \mathcal{L}_{c,G} + \lambda_{fm}\mathcal{L}_{fm} + \lambda_{ms}\mathcal{L}_{ms}$, in which $\mathcal{L}_{c,D}$ and $\mathcal{L}_{c,G}$ are defined below Eqn. 8.

Table 1. FID (↓) and LPIPS (↑) of images generated by different methods for unseen categories on four datasets in 1/3-shot setting

Method	Shot	VGGFace		Flowers		Animal Faces		NABirds	
		FID	LPIPS	FID	LPIPS	FID	LPIPS	FID	LPIPS
FIGR [13]	3	139.83	0.0834	190.12	0.0634	211.54	0.0756	210.75	0.0918
DAWSON [38]	3	137.82	0.0769	188.96	0.0583	208.68	0.0642	181.97	0.1105
GMN [4]	3	136.21	0.0902	200.11	0.0743	220.45	0.0868	208.74	0.0923
DAGAN [2]	3	128.34	0.0913	151.21	0.0812	155.29	0.0892	159.69	0.1405
DAGAN [2]	1	*134.28*	*0.0608*	*179.59*	*0.0496*	*185.54*	*0.0687*	*183.57*	*0.0967*
MatchingGAN[29]	3	118.62	0.1695	143.35	0.1627	148.52	0.1514	142.52	0.1915
F2GAN [30]	3	109.16	0.2125	120.48	0.2172	117.74	0.1831	126.15	0.2015
LoFGAN [23]	3	106.24	0.2096	112.55	0.2687	116.45	0.1756	124.56	0.2041
DeltaGAN	3	**78.35**	**0.3487**	**104.62**	**0.4281**	**87.0**4	**0.4642**	**95.97**	**0.5136**
DeltaGAN	1	***80.12***	***0.3146***	***109.78***	***0.3912***	***89.81***	***0.4418***	***96.79***	***0.5069***

4 Experiments

We conduct experiments on six few-shot image datasets: EMNIST [14], VGGFace [8], Flowers [49], Animal Faces [16], NABirds [63], and Foods [34]. Following MatchingGAN and FUNIT, we split all categories into seen categories and unseen categories. After having a few trials, we set $\lambda_1 = 10$, $\lambda_{fm} = 0.1$, and $\lambda_{ms} = 10$ by observing the quality of generated images during training. We adopt Adam optimizer with learning rate of $1e{-}4$. The batch size is set to 16 and our model is trained for 200 epochs. The details of datasets and implementation are reported in Supplementary.

4.1 Evaluation of Generated Images

To evaluate the quality of images generated by different methods, we calculate Fréchet Inception Distance (FID) [26] and Learned Perceptual Image Patch Similarity (LPIPS) [69] on four datasets. FID is used to measure the distance between the extracted features of generated unseen images and those of real unseen images. LPIPS is used to measure the diversity of generated unseen images. For each unseen category, the average of pairwise distances among generated images is calculated, and then the average of all unseen categories is calculated as the final LPIPS score. Since the number of conditional images in fusion-based methods GMN [4], MatchingGAN [29], F2GAN [30], and LoFGAN [23]) is a tunable hyper-parameter, we use 3 conditional images in each training and testing episode. In the testing stage, if K images are provided for each unseen category, we refer to this setting as K-shot setting. We report the 3-shot results for all methods and 1-shot results for the methods which only require one conditional image.

In either setting, following [23,30], we use each method to generate 128 images for each unseen category, which are used to calculate FID and LPIPS. For DeltaGAN and DAGAN which are applicable to both 1-shot and 3-shot settings, we generate 128 images based on one conditional image in 1-shot setting

Fig. 4. Images generated by our DeltaGAN in 1-shot setting on four datasets (from top to bottom:EMNIST, VGGFace, Flowers, Animal Faces, NABirds, and Foods). The conditional images are in the leftmost column

and generate 128 images by randomly sampling one conditional image each time in 3-shot setting. The results are summarized in Table 1, we can observe that our method achieves the lowest FID and highest LPIPS in the 3-shot setting, which demonstrates that our method could generate more diverse and realistic images compared with baseline methods. Besides, our method in 1-shot setting also achieves competitive results, which are even better than other baselines in 3-shot setting. We also compare our DeltaGAN with other few-shot image generation method [50] in Supplementary.

We show some example images generated by our DeltaGAN on six datasets in Fig. 4. We exhibit 12 generated images based on one conditional unseen image by sampling different random vectors. On EMNIST dataset, we can see that generated images maintain the concepts of conditional images and have remarkable diversity. On natural datasets VGGFace, Flowers, Animal Faces, NABirds, and Foods, our DeltaGAN can generate diverse images with high fidelity.

For comparison, we also show some example images generated by DAGAN and F2GAN in Fig. 5. For DAGAN, we arrange the results according to the conditional image. It can be seen that the structures of images produced by DAGAN are almost the same as the conditional image. For F2GAN, the generated images are still close to one of the conditional images and may have unreasonable

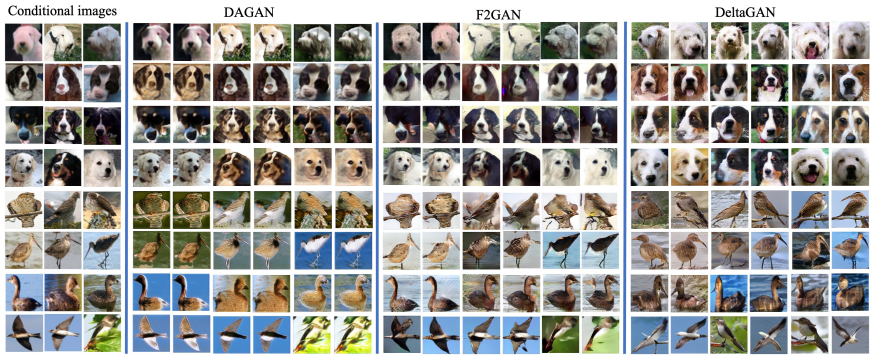

Fig. 5. Images generated by DAGAN, F2GAN, and our DeltaGAN in 3-shot setting on two datasets (from top to bottom: Animal Faces and NABirds). The conditional images are in the left three columns

shapes when fusing conditional images. Apparently, our DeltaGAN can produce images of higher quality and more diversity.

4.2 Few-shot Classification

In this section, we demonstrate that the new images generated by our DeltaGAN can greatly benefit few-shot classification. The experiments for low-data classification and comparison with traditional data augmentation methods can be found in Supplementary. Following the N-way C-shot setting in few-shot classification [20], in which evaluation episodes are created and the averaged accuracy over multiple evaluation episodes is calculated for evaluation. In each evaluation episode, N categories from unseen categories are randomly selected and C images from each of N categories are randomly selected. These selected $N \times C$ images are used as training set while the remaining unseen images from N unseen categories are used as test set. We pretrain ResNet18 [25] on the seen images and remove the last FC layer as the feature extractor, which is used to extract features for unseen images. In each evaluation episode in N-way C-shot setting, our DeltaGAN generates 512 new images to augment each of N categories. Based on the extracted features, we train a linear classifier with $N \times (C + 512)$ training images, which is then applied to the test set. we train a linear classifier to evaluate the few-shot generation ability of our DeltaGAN. Besides $N \times C$ training images, our generator can generate 512 images to augment each of N categories in the training set.

We compare our DeltaGAN with existing few-shot classification methods, including the representative methods MatchingNets [64], RelationNets [59], MAML [20] as well as the state-of-the-art methods MTL [58], MatchingNet-LFT [61], DPGN [67], DeepEMD [68], and GCNET [41]. For these baselines, no augmented images are added to the training set in each evaluation episode. Instead, the im-

Table 2. Accuracy(%) of different methods on three datasets in few-shot classification setting (10-way 1/5-shot). Note that fusion-based methods MatchingGAN, F2GAN, and LoFGAN are not applicable in 1-shot setting

Method	VGGFace		Flowers		Animal Faces	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MatchingNets [64]	33.68	48.67	40.96	56.12	36.54	50.12
MAML [20]	32.16	47.89	42.95	58.01	35.98	49.89
RelationNets [59]	39.95	54.12	48.18	61.03	45.32	58.12
MTL [58]	51.45	68.95	54.34	73.24	52.54	70.91
MatchingNet-LFT [61]	54.34	69.92	58.41	74.32	56.83	71.62
DPGN [67]	54.83	70.27	58.95	74.56	57.18	72.02
DeepEMD [68]	54.15	70.35	59.12	73.97	58.01	72.71
GCNET [41]	53.73	71.68	57.61	72.47	56.64	71.53
Delta-encoder [57]	53.19	67.57	56.05	72.84	56.38	71.29
MatchingGAN [29]	-	70.94	-	74.09	-	70.89
F2GAN [30]	-	72.31	-	75.02	-	73.19
LoFGAN [23]	-	73.01	-	75.86	-	73.43
DeltaGAN	**56.8**5	**75.71**	**61.23**	**77.09**	**60.31**	**74.59**

ages from seen categories are used to train those few-shot classifiers by strictly following their original training procedure. We also compare our DeltaGAN with few-shot image generation methods MatchingGAN and F2GAN as well as few-shot feature generation method Delta-encoder. We adopt the same augmentation strategy as our DeltaGAN in each evaluation episode. Besides, we compare our DeltaGAN with few-shot image translation method FUNIT [42] in Supplementary. By taking 10-way 1-shot/5-shot as examples, we report the averaged accuracy over 10 episodes on three datasets in Table 2. Our method achieves the best performance on all datasets compared with few-shot classification and few-shot generation baselines, which demonstrates the high quality of generated images by our DeltaGAN.

4.3 Ablation Studies

We analyze the impact of each loss and alternative network designs on Animal Faces dataset in 1-shot setting. For each ablated method, FID, LPIPS, and the accuracy of 10-way 1-shot classification augmented with generated images are reported in Table 3.

Loss terms: In our method, we employ a reconstruction loss $\mathcal{L}_1$, a mode seeking loss $\mathcal{L}_{ms}$, a feature matching loss $\mathcal{L}_{fm}$, a classification loss $\mathcal{L}_c$, and an adversarial loss $\mathcal{L}^I_{adv}$. To investigate the impact of each loss term, we conduct ablation studies on Animal Faces dataset by removing each loss term from the final objective in Eqn. 11 separately. The results are summarized in Table 3, which shows that the diversity and fidelity of generated images are compromised when removing $\mathcal{L}_1$. By removing mode seeking loss $\mathcal{L}_{ms}$, we can see that all metrics become much worse, which implies the mode collapse issue after removing $\mathcal{L}_{ms}$. Another observation is that ablating $\mathcal{L}_{fm}$ leads to slight degradation of generated images. Removing $\mathcal{L}_c$ results in severe degradation of generated images, since the

Table 3. Ablation studies of our loss terms and alternative network designs on Animal Faces dataset

Setting	Accuracy(%) ↑	FID ↓	LPIPS ↑
w/o $\mathcal{L}_1$	58.68	100.21	0.4191
w/o $\mathcal{L}_{ms}$	50.08	121.74	0.2976
w/o $\mathcal{L}_{fm}$	59.17	95.82	0.4324
w/o $\mathcal{L}_c$	42.21	196.18	0.4119
w/o $\mathcal{L}^I_{adv}$	52.18	139.46	0.3912
w/o $\mathcal{L}^M_{adv}$	57.12	115.11	0.4153
w/o real delta	53.03	128.69	0.3838
Global delta	58.96	94.51	0.4311
SC delta	56.11	101.05	0.4162
DC delta	55.29	105.91	0.4021
Simple D_1	54.53	129.17	0.3012
Simple D_2	58.01	109.54	0.4401
Simple D_3	59.51	94.12	0.4392
Linear delta	53.89	122.71	0.4091
DeltaGAN	**60.31**	**89.81**	**0.4418**

generated images may not belong to the category of conditional image. When $\mathcal{L}^I_{adv}$ is removed from the final objective, the worse quality of generated images indicates that typical adversarial loss can ensure the fidelity of generated images. To investigate the impact of our adversarial delta matching loss $\mathcal{L}^M_{adv}$ in Eqn. 9, We remove $\mathcal{L}^M_{adv}$ from the final objective in Eqn. 11, which is referred to as "w/o $\mathcal{L}^M_{adv}$" in Table 3. We can see that the diversity and fidelity of generated images are compromised without $\mathcal{L}^M_{adv}$, because $\mathcal{L}^M_{adv}$ can bridge the gap between real delta and fake delta.

Without real delta: To investigate the necessity of enforcing generated fake deltas to be close to real deltas, we cut off the links between real delta and fake delta by removing the reconstruction subnetwork and adversarial delta matching loss (*i.e.*, removing $\{\mathcal{L}^M_{adv}, \mathcal{L}_1, \mathcal{L}_{fm}\}$), which is referred to as "w/o real delta" in Table 3. Compared with DeltaGAN, both diversity and realism are significantly degraded, because generation subnetwork fails to generate meaningful deltas without the guidance of reconstruction subnetwork and adversarial delta matching loss. Thus, we conclude that mode seeking loss needs to cooperate with our framework to produce realistic and diverse images. Another observation is that "w/o $\mathcal{L}^M_{adv}$" is better than "w/o real delta", which can be explained as follows. Even without using adversarial delta matching loss, since the reconstruction subnetwork and the generation subnetwork share the same E_c and G, generated fake deltas have been implicitly pulled close to real deltas.

Sample-specific delta: To corroborate the superiority of sample-specific delta, we directly use random vectors to generate deltas, which is referred to as "Global delta" in Table 3. It can be seen that our design of sample-specific deltas can benefit the quality of generated images. Besides, with our trained DeltaGAN model, we exchange sample-specific deltas within images from the same category (*resp.,* across different categories) to generate new images, which is referred to as “SC delta” (*resp.,* “DC delta”) in Table 3. Compared with “SC delta” and “DC delta”, our DeltaGAN achieves the best performance on all metrics, which verifies our assumption that delta is sample-specific and exchangeable use of deltas may lead to performance drop. We also visualize some examples generated by “SC delta” (*resp.,* “DC delta”) in Supplementary.

Delta matching discriminator: In Section 3.2, we use conditional image, sample-specific delta, and output image as input triplet $\{\hat{D}_I(\boldsymbol{x}_1), \boldsymbol{\Delta}_{x_1}, \hat{D}_I(\boldsymbol{x}_2)\}$ for our delta matching discriminator D_M, which judges whether the conditional-output image pair matches the corresponding sample-specific delta. To evaluate the effectiveness and necessity of this input format, we explore different types of inputs for delta matching discriminator. As shown in Table 3, we use $\{\boldsymbol{\Delta}_{x_1}\}$ (*resp.*, $\{\hat{D}_I(\boldsymbol{x}_1), \boldsymbol{\Delta}_{x_1}\}$, $\{\hat{D}_I(\boldsymbol{x}_2), \boldsymbol{\Delta}_{x_1}\}$) as inputs of D_M, which is referred to as “Simple D_1” (*resp.*,“Simple D_2”,“Simple D_3”). We can see that “Simple D_1” is the worst, which demonstrates that only employing adversarial loss on delta does not work well. Besides, both “Simple D_2” and “Simple D_3” are worse than our DeltaGAN, which demonstrates the effectiveness of matching conditional-output image pair with the corresponding sample-specific delta.

Linear offset delta: To evaluate the effect of the learned non-linear “delta”, we replace the non-linear “delta” with linear “delta”, which is referred to as “Linear delta” in Table 3. In the reconstruction subnetwork, we set $\boldsymbol{\Delta}^r_{x_1} = E_\Delta(\boldsymbol{x}_2) - E_\Delta(\boldsymbol{x}_1)$, and $\hat{\boldsymbol{x}}_2 = G(\boldsymbol{\Delta}^r_{x_1} + E_c(\boldsymbol{x}_1))$, which means that we simply learn linear offset “delta” from same-class pairs of training data. In the generation subnetwork, we apply the generated fake “delta” $\boldsymbol{\Delta}^f_{x_1}$ to conditional image $\boldsymbol{x}_1$ to generate new image $\tilde{\boldsymbol{x}}_2 = G(\boldsymbol{\Delta}^f_{x_1} + E_c(\boldsymbol{x}_1))$. Based on Table 3, the FID gap between “Linear delta” and “DeltaGAN” indicates that complex transformations of intra-category pairs cannot be well captured by linear offset.

5 Conclusion

In this paper, we have explored applying sample-specific deltas to a conditional image to generate new images. Specifically, we have proposed a novel few-shot generation method DeltaGAN composed of a reconstruction subnetwork and a generation subnetwork, which are bridged by an adversarial delta matching loss. The experimental results on six datasets have shown that our DeltaGAN can substantially improve the quality and diversity of generated images compared with existing few-shot image generation methods.

Acknowledgement

The work is supported by Shanghai Municipal Science and Technology Key Project (Grant No. 20511100300), Shanghai Municipal Science and Technology Major Project, China (2021SHZDZX0102), and National Science Foundation of China (Grant No. 61902247).

References

1. Almahairi, A., Rajeswar, S., Sordoni, A., Bachman, P., Courville, A.C.: Augmented cyclegan: Learning many-to-many mappings from unpaired data. In: ICML (2018)
2. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. arXiv preprint arXiv:1711.04340 (2017)
3. Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: CVAE-GAN: fine-grained image generation through asymmetric training. In: ICCV (2017)
4. Bartunov, S., Vetrov, D.: Few-shot generative modelling with generative matching networks. In: AISTATS (2018)
5. Binkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. In: ICLR (2018)
6. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. In: ICLR (2018)
7. Cao, B., Wang, N., Li, J., Gao, X.: Data augmentation-based joint learning for heterogeneous face recognition. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) (2019)
8. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: FG (2018)
9. Chen, M., Fang, Y., Wang, X., Luo, H., Geng, Y., Zhang, X., Huang, C., Liu, W., Wang, B.: Diversity transfer network for few-shot learning. In: AAAI (2020)
10. Chen, T., Zhai, X., Ritter, M., Lucic, M., Houlsby, N.: Self-supervised gans via auxiliary rotation loss. In: CVPR (2019)
11. Chen, Z., Fu, Y., Wang, Y.X., Ma, L., Liu, W., Hebert, M.: Image deformation meta-networks for one-shot learning. In: CVPR (2019)
12. Chen, Z., Fu, Y., Zhang, Y., Jiang, Y.G., Xue, X., Sigal, L.: Multi-level semantic feature augmentation for one-shot learning. TIP **28**(9), 4594–4605 (2019)
13. Clouâtre, L., Demers, M.: Figr: Few-shot image generation with reptile. arXiv preprint arXiv:1901.02199 (2019)
14. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: an extension of MNIST to handwritten letters. In: IJCNN (2017)
15. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V.K., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: CVPR (2019)
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
17. Dixit, M., Kwitt, R., Niethammer, M., Vasconcelos, N.: AGA: attribute-guided augmentation. In: CVPR (2017)
18. Donahue, J., Simonyan, K.: Large scale adversarial representation learning. In: Advances in Neural Information Processing Systems (2019)
19. Feng, R., Gu, J., Qiao, Y., Dong, C.: Suppressing model overfitting for image super-resolution networks. In: CVPR (2019)
20. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017)
21. Gao, H., Shou, Z., Zareian, A., Zhang, H., Chang, S.: Low-shot learning via covariance-preserving adversarial augmentation networks. In: NeurIPS (2018)
22. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014)
23. Gu, Z., Li, W., Huo, J., Wang, L., Gao, Y.: Lofgan: Fusing local representations for few-shot image generation. In: ICCV (2021)

24. Hariharan, B., Girshick, R.B.: Low-shot visual recognition by shrinking and hallucinating features. In: ICCV (2017)
25. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
26. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS (2017)
27. Ho, D., Liang, E., Chen, X., Stoica, I., Abbeel, P.: Population based augmentation: Efficient learning of augmentation policy schedules. In: ICML (2019)
28. Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. In: ICML (2018)
29. Hong, Y., Niu, L., Zhang, J., Zhang, L.: Matchinggan: Matching-based few-shot image generation. In: ICME (2020)
30. Hong, Y., Niu, L., Zhang, J., Zhao, W., Fu, C., Zhang, L.: F2gan: Fusing-and-filling gan for few-shot image generation. In: ACM MM (2020)
31. Jo, H.J., Min, C.H., Song, J.B.: Bin picking system using object recognition based on automated synthetic dataset generation. In: 2018 15th International Conference on Ubiquitous Robots (UR) (2018)
32. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: CVPR (2019)
33. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. arXiv preprint arXiv:1912.04958 (2019)
34. Kawano, Y., Yanai, K.: Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In: ECCV (2014)
35. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
36. Lake, B., Salakhutdinov, R., Gross, J., Tenenbaum, J.: One shot learning of simple visual concepts. In: CogSci (2011)
37. Li, Y., Zhang, R., Lu, J., Shechtman, E.: Few-shot image generation with elastic weight consolidation. In: NeurIPS (2020)
38. Liang, W., Liu, Z., Liu, C.: Dawson: A domain adaptive few shot generation framework. arXiv preprint arXiv:2001.00576 (2020)
39. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. In: NeurIPS (2019)
40. Liu, J., Sun, Y., Han, C., Dou, Z., Li, W.: Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In: CVPR (2020)
41. Liu, L., Wang, B., Kuang, Z., Xue, J.H., Chen, Y., Yang, W., Liao, Q., Zhang, W.: Gendet: Meta learning to generate detectors from few shots. TNNLS (2021)
42. Liu, M., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. In: ICCV (2019)
43. Makhzani, A., Frey, B.J.: Pixelgan autoencoders. In: Advances in Neural Information Processing Systems (2017)
44. Mao, Q., Lee, H.Y., Tseng, H.Y., Ma, S., Yang, M.H.: Mode seeking generative adversarial networks for diverse image synthesis. In: CVPR (2019)
45. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: ICCV (2017)
46. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: Proceedings of 6th International Conference on Learning Representations (ICLR) (2018)
47. Miyato, T., Koyama, M.: cgans with projection discriminator. In: ICLR (2018)
48. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)

49. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: CVGIP (2008)
50. Ojha, U., Li, Y., Lu, J., Efros, A.A., Lee, Y.J., Shechtman, E., Zhang, R.: Few-shot image generation via cross-domain correspondence. In: CVPR (2021)
51. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. In: CVPR (2017)
52. Pu, Y., Zhe, G., Henao, R., Xin, Y., Carin, L.: Variational autoencoder for deep learning of images, labels and captions. In: NeurIPS (2016)
53. Ratner, A.J., Ehrenberg, H., Hussain, Z., Dunnmon, J., Ré, C.: Learning to compose domain-specific transformations for data augmentation. In: NeurIPS (2017)
54. Reed, S., Chen, Y., Paine, T.L., Den Oord, A.V., Eslami, S.M.A., Rezende, D.J., Vinyals, O., De Freitas, N.: Few-shot autoregressive density estimation: towards learning to learn distributions. In: ICLR (2018)
55. Rezende, D.J., Mohamed, S., Danihelka, I., Gregor, K., Wierstra, D.: One-shot generalization in deep generative models. In: ICML (2016)
56. Robb, E., Chu, W.S., Kumar, A., Huang, J.B.: Few-shot adaptation of generative adversarial networks. arXiv preprint arXiv:2010.11943 (2020)
57. Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Kumar, A., Feris, R., Giryes, R., Bronstein, A.: Delta-encoder: an effective sample synthesis method for few-shot object recognition. In: NeurIPS (2018)
58. Sun, Q., Liu, Y., Chua, T.S., Schiele, B.: Meta-transfer learning for few-shot learning. In: CVPR (2019)
59. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR (2018)
60. Tian, K., Lin, C., Sun, M., Zhou, L., Yan, J., Ouyang, W.: Improving auto-augment via augmentation-wise weight sharing. NeurIPS (2020)
61. Tseng, H., Lee, H., Huang, J., Yang, M.: Cross-domain few-shot classification via learned feature-wise transformation. In: Proceedings of 8th International Conference on Learning Representations (ICLR) (2020)
62. Tsutsui, S., Fu, Y., Crandall, D.: Meta-reinforced synthetic data for one-shot fine-grained visual recognition. In: NeurIPS (2019)
63. Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: CVPR (2015)
64. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: NeurIPS (2016)
65. Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F.S., van de Weijer, J.: Minegan: Effective knowledge transfer from gans to target domains with few images. In: CVPR (2020)
66. Wang, Y., Girshick, R.B., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: CVPR (2018)
67. Yang, L., Li, L., Zhang, Z., Zhou, X., Zhou, E., Liu, Y.: Dpgn: Distribution propagation graph network for few-shot learning. In: CVPR (2020)
68. Zhang, C., Cai, Y., Lin, G., Shen, C.: Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In: CVPR (2020)
69. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
70. Zhang, X., Wang, Z., Liu, D., Ling, Q.: DADA: Deep adversarial data augmentation for extremely low data regime classification. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2019)

71. Zhao, M., Cong, Y., Carin, L.: On leveraging pretrained gans for generation with limited data. In: ICML (2020)