LoFGAN: Fusing Local Representations for Few-shot Image Generation

Zheng Gu[†1], Wenbin Li [†1], Jing Huo[*1], Lei Wang[2], and Yang Gao[1]

[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]School of Computing and Information Technology, University of Wollongong

Abstract

Given only a few available images for a novel unseen category, few-shot image generation aims to generate more data for this category. Previous works attempt to globally fuse these images by using adjustable weighted coefficients. However, there is a serious semantic misalignment between different images from a global perspective, making these works suffer from poor generation quality and diversity. To tackle this problem, we propose a novel Local-Fusion Generative Adversarial Network (LoFGAN) for few-shot image generation. Instead of using these available images as a whole, we first randomly divide them into a base image and several reference images. Next, LoFGAN matches local representations between the base and reference images based on semantic similarities, and replaces the local features with the closest related local features. In this way, LoFGAN can produce more realistic and diverse images at a more fine-grained level, and simultaneously enjoy the characteristic of semantic alignment. Furthermore, a local reconstruction loss is also proposed, which can provide better training stability and generation quality. We conduct extensive experiments on three datasets, which successfully demonstrates the effectiveness of our proposed method for few-shot image generation and downstream visual applications with limited data. Code is available at https://github.com/edward3862/LoFGAN-pytorch.

1. Introduction

As a representative deep generative model, generative adversarial networks (GANs) [7] have shown impressive results in various visual tasks in recent years. However, most GAN models still struggle with insufficient training data [25]. Although many GAN-based few-shot learning algorithms have been presented recently, most of these algorithms are specially designed for discriminative tasks like image classification [20] and segmentation [21], rather than the pure image generation in a data-limited regime.

To this end, few-shot image generation has aroused increasing attention. The goal of few-shot image generation is to generate diverse images for a novel category, when a few available images of this category are given. In particular, inspired by the episodic training mechanism [20], the generative model is generally trained on an auxiliary dataset with sufficient labeled training categories and images. After that, given a few images from a new unseen category, the learned generative model is expected to generate diverse images for this specific category. Considering the disjoint label spaces between the seen auxiliary dataset and the unseen test dataset, the generative model is hoped to obtain the generalization ability by learning from thousands of simulated few-shot image generation tasks.

Current few-shot generation approaches can be roughly divided into three types, *i.e.,* transformation-based [2], optimization-based [5, 13], and fusion-based [8, 9]. Transformation-based methods apply intra-category transformation on one conditional image while optimization-based methods introduces a meta-learning paradigm [6, 16] to learn an initialization strategy for unconditional image generation tasks, both applicable to simple generation tasks. Fusion-based methods (*inspired by metric-based few-shot learning*) define this problem as a conditional generation task. The generative model encodes several input images to a feature space and performs a fusion operation (*instead of the comparison operation in metric-based few-shot classification*). The fused feature is then decoded back to a realistic image of the same category.

The essence of fusion-based few-shot generation is to implement a label-consistent mapping from a few conditional inputs to diverse outputs while simultaneously maintaining the image quality and diversity. Technically, GMN [3] combines Matching Network [20] with VAE [11] by appending a decoder after the matching procedure. Due to the limited generation capacity of VAE, this method is only applicable to generate digits and simple visual pat-

[†]Equal contribution, [*]Corresponding author.

terns. To address this problem, MatchingGAN [8] replaces the VAE part with a generative adversarial network and achieves natural image generation for the first time, but still struggles with complex natural images. Recently, F2GAN [9] proposes a fuse-and-fill strategy in the fusion procedure to enhance the generation ability. However, the above methods still suffer from a limited and imprecise generation space which is formulated by a strictly linear combination as well as a weighted image-level reconstruction loss. In other words, the images in the same category are linearly fused with interpolation coefficients at the global feature map level. This may bring two problems. First, when the input images are not semantically aligned, the fused feature map will be misaligned too, and globally adding them will produce aliasing artifacts in the output image. Second, simple global combination will also hurt the generation diversity because the relative position of each local semantic area is strictly fixed during fusion.

To tackle the above problems, we propose a novel local-fusion approach to fusion-based few-shot image generation. Given a handful of images, we randomly choose one of them as a base image and the others as reference images. The base image defines a basis of the generation and the reference images act like a bank of many available local representations. We first select local positions randomly in the base image. Then, since the input images come from the same category, we can find semantically matched local representations for these selected positions in the bank. We fuse the matched local representations from different images in a more fine-grained level, and replace them back to the corresponding positions in the base image. The whole process is completed in a local fusion module at the feature level without additional parameters. Since the fusion operation is performed in local areas rather than the whole feature map, the generated images will contain fewer artifacts.

In addition, we propose a new local reconstruction loss to better cooperate with the proposed local fusion module at the training stage. In previous fusion-based few-shot image generation methods, a global reconstruction loss is used to enforce the generated images to contain the information of the input images, which is implemented by minimizing the pixel-level distance between the generated image and a weighted sum of input images. However, adding the input images at each pixel position as a reconstruction target can not ensure semantically alignment because each image is unique in content with different structures. To this end, we consistently stand in a 'local' view to tackle this problem. We enforce the generated images to be close to the input images in some local areas instead. We reproduce the above feature-level local fusion procedure at the image level to build a clearer image as the reconstruction target. We find the proposed local reconstruction loss can further improve the generation quality for few-shot image generation.

Our contributions can be summarized as follows:

- We propose *Local-Fusion Generative Adversarial Network (LoFGAN)* for few-shot image generation, which can flexibly match the semantically nearest local features to achieve better generation quality and diversity.
- We present a local fusion module along with a new local reconstruction loss to better train the network, which provides more refined guidance for generation.
- We conduct comprehensive experiments on three datasets where our method achieves the state-of-the-art performance in few-shot image generation, demonstrating the effectiveness of our proposed method.

2. Related Work

In this section, we introduce three kinds of related work in this paper: generative adversarial networks, few-shot generative adaptation and few-shot image generation.

2.1. Generative Adversarial Networks

Generative Adversarial Network (GAN) [7] is one category of generative models which is trained via adversarial learning. With the great ability of GANs to fit a data distribution, great improvement has been made in various tasks ranging from image generation [10], image editing [1] to image-to-image translation [26]. However, the impressive results are mainly attributed to the unlimited supply of training images. The discriminator may easily overfit in case of limited data, which makes the model hard to converge. Recently, some advanced data augmentation strategies [2, 25] have been proposed for training GANs with limited data, but these methods are mainly designed for unconditional generation, which is more like the vanilla GAN [7]. Different from the mainstream, in this paper, we try to solve this problem in a few-shot learning paradigm. We are interested in teaching a GAN to generate different images for a novel category given a few images of this category.

2.2. Few-shot Generative Adaptation

Estimation of a distribution from limited observations is biased and inaccurate, especially for GANs. Some methods try to mitigate the challenge of insufficient data via transfer learning [23, 22, 19]. With the help of auxiliary data (mainly for pre-training), these methods leverage a pre-trained GAN and adapt it to another image domain by adjusting the model parameters [12, 24]. The adapted model should be able to generate images within the target image domain with limited data. We classify these methods as few-shot generative adaptation, which assumes that the limited dataset and the auxiliary dataset are disjoint in the feature space (*i.e.,* in different image domains). These methods also assume that the model should be first pre-trained

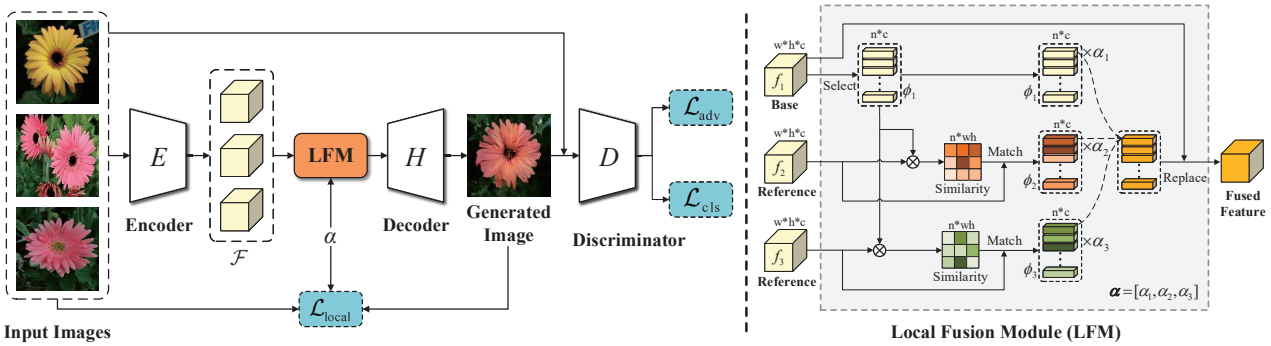

Figure 1: Our LoFGAN framework contains an encoder E, a local fusion module, a decoder H, and a discriminator D. The random coefficient vector $\boldsymbol{\alpha}$ is the input of both LFM module and local reconstruction loss. The local fusion module randomly chooses one of the encoded features as base and the rest features as references, and fuses them by local selection, local matching and local replacement.

on a large dataset and then carefully fine-tuned on the limited dataset. Different from the above methods, we assume the auxiliary dataset and limited dataset are disjoint in the label space (*i.e.,* different categories from the same image domain), and the model should be able to generate images for any unseen category directly without fine-tuning.

2.3. Few-shot Image Generation

Given a few images for an unseen category, the goal of few-shot image generation is to produce realistic and diverse images for this category, which is different from the goal of few-shot generative adaptation methods. Optimization-based methods FIGR [5] and DAWSON [13] combine adversarial learning with meta learning methods (*i.e.,* Reptile [16] and MAML [6]), but the generation quality is limited. For fusion-based methods, GMN [3] and MatchingGAN [8] generalize the matching network from few-shot classification task to few-shot image generation with VAE and GAN. F2GAN [9] improves MatchingGAN by adding a Non-local Attentional Fusion module to fuse and fill different level of features to generate images. These methods fuse the high level image features with a global coefficient, which will bring more aliasing artifacts and less diversity to the generated images. Besides, a global reconstruction loss is used to constrain the model to produce images that look like a weighted stack of the input images, which will further hurt the generation quality. Different from existing methods, we aim to fuse the deep features at a more fine-grained level by selecting, matching and replacing local representations, and use a local-based reconstruction loss to reduce aliasing artifacts.

3. Our Method

3.1. Overall Framework

Given k images sampled from a novel category, our goal is to generate new images for this category, which is called a k-shot image generation task. To achieve this goal, we can split an image dataset into two parts: seen categories $\mathbb{C}_s$ and unseen categories $\mathbb{C}_u$, where $\mathbb{C}_s \cap \mathbb{C}_u = \varnothing$. In the training stage, we sample hundreds of k-shot image generation tasks from $\mathbb{C}_s$ and feed them into the model, encouraging it to learn transferable generation ability to generate new images for unseen categories. In the test stage, the model can take images from one category in $\mathbb{C}_u$ to generate a new image.

Figure 1 shows the overall framework of our method. The generator G is a conditional one that contains an encoder E, a decoder H, and a local fusion module LFM. The input images $X = \{x_1, \ldots, x_k\}$ are first fed into the encoder E to extract deep features $\mathcal{F} = E(X)$. Then, the LFM module takes $\mathcal{F}$ and a random coefficient vector $\boldsymbol{\alpha}$ as inputs and produces a semantically aligned fused feature $\hat{\mathcal{F}} = \mathrm{LFM}(\mathcal{F}, \boldsymbol{\alpha})$. After that, the decoder H decodes the feature back to the image and obtains the generated image $\hat{x} = H(\hat{\mathcal{F}})$. The real images X and generated image $\hat{x}$ are fed into the discriminator D for adversarial training.

3.2. Local Fusion Module

Figure 1 shows a detailed illustration of the proposed LFM module under the 3-shot image generation setting. Given a set of encoded feature maps $\mathcal{F} = E(X) \in \mathbb{R}^{k\times w\times h\times c}$. Each $w \times h \times c$ tensor in $\mathcal{F}$ can be viewed as a set of $h \times w$ $c-$dimensional local representations. Our idea is to randomly assign one feature map from $\mathcal{F}$ as a base feature f_{base}, and denote the rest $k-1$ features maps as reference features $\mathbb{F}_{ref}$. The local fusion module will take the select f_{base} as a basis and the rest $\mathbb{F}_{ref}$ as a bank of local features to produce a fused feature. The whole fusion process can be divided into three steps, including local selection, local matching and local replacement.

Local Selection. Once the f_{base} is determined, the first step is to select which local representations in f_{base} should be replaced. Here we randomly select local representations from the $h \times w$ local positions in f_{base}. More specifically,

we select a number of $n = \eta \times w \times h$ local representations, where $\eta \in (0, 1]$ is a selection ratio that decides how many local representations should be fused. After feature selection, we obtain a set of n $c-$dimensional local representations ϕ_{base} from the base feature f_{base}.

Local Matching. The next step is to find semantically matched local representations in $\mathbb{F}_{ref}$ that can be used to replace ϕ_{base}. For each reference feature f_{ref} in $\mathbb{F}_{ref}$, we calculate the similarity between every two positions in ϕ_{base} and f_{ref} to build a similarity map M as below,

$$M^{(i,j)} = g(\phi_{base}^{(i)}, f_{ref}^{(j)}), \tag{1}$$

where $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, h \times w\}$ and g is a similarity metric. According to the similarity map, we can find the most similar local representation for each position in ϕ_{base}, and use them to replace the origin local representations in f_{base} in the next step. We denote the set of the best matched local representations from the $k-1$ reference feature maps as $\Phi_{ref} \in \mathbb{R}^{(k-1)\times n \times c}$. Note that we also record the position information for every local representation in ϕ_{base} and Φ_{ref}, which we use to calculate the local reconstruction loss in the next section.

Local Replacement. For each $c-$dimensional local representation in ϕ_{base}, we now have $k-1$ candidate local representations. For example, $\phi_{ref}^{(1)} \in \mathbb{R}^{(k-1)\times c}$ contains the most similar local representations with the first local representation $\phi_{base}^{(1)} \in \mathbb{R}^{c}$ that we can find in every f_{ref} (see the dotted lines in the LFM module in Figure 1). We fuse all of these local representations together and replace them to the corresponding positions in f_{base}. We use a random coefficient vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_k]$ to fuse the features for all the positions selected,

$$\phi_{fuse}^{(t)} = \alpha_{base} \cdot \phi_{base}^{(t)} + \sum_{i=1,\ldots,k, i\neq base} \alpha_i \cdot \phi_{ref}^{(i)}(t), \tag{2}$$

where $\sum_{i=1}^{k} \alpha_i = 1, \alpha_i \geq 0$ and $t = 1, \ldots, n$. We retain original local representation with a ratio α_{base}. Then we replace all the n fused local representations ϕ_{fuse} back to the corresponding positions in f_{base}. This produces a fused feature map $\hat{\mathcal{F}}$ as the output of the LFM module.

3.3. Local Reconstruction Loss

Given a set of input images $X = \{x_1, \ldots, x_k\}$ and a random coefficient vector $\boldsymbol{\alpha}$, previous methods adopt a weighted image-level reconstruction loss to constrain the generated image $\hat{x}$, which can be formulated as follows,

$$\mathcal{L}_{\text{global}} = ||\hat{x} - \sum_{i=1}^{k} \alpha_i \cdot x_i||_1, \tag{3}$$

where $\sum_{i=1}^{k} \alpha_i = 1, \alpha_i \geq 0$. Eq.3 means the generated image $\hat{x}$ should look more like x_i if α_i is given high, which is equal to minimizing the difference between the generated image with a weighted stack of the input images. This may lead to unsuitable supervision because the weighed stack of images will have aliasing artifacts when the input images are not aligned. To this end, we introduce a local reconstruction loss to tackle this problem. The idea is to reproduce the feature-level local fusion procedure at the image-level. Specifically, we record the positions for every selected base and reference local representations in ϕ_{base} and Φ_{ref}, and map each position of the selected feature back to the original image size to get a roughly fused image $\text{LFM}(X, \alpha)$. After that, we constrain $\hat{x}$ by the following loss,

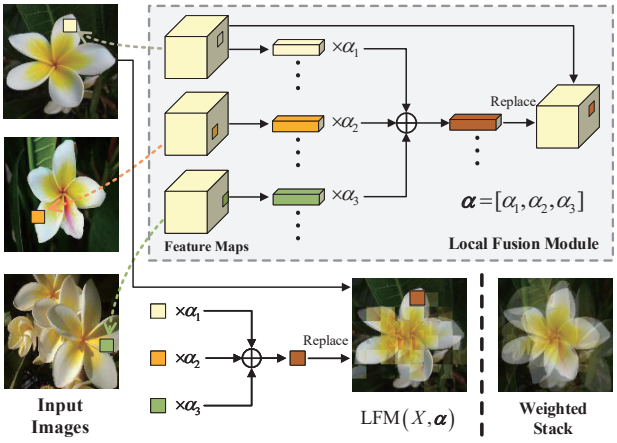

Figure 2: Calculation of the target image for local reconstruction loss. The whole process can be viewed as a reproduction of local replacement procedure. Comparing with the global reconstruction loss, our proposed local reconstruction loss produces more clear reconstruction target to train the model. Better view in color.

$$\mathcal{L}_{\text{local}} = ||\hat{x} - \text{LFM}(X, \boldsymbol{\alpha})||_1. \tag{4}$$

As seen in Figure 2, each position in the feature map corresponds to an image patch in the input image. Since we have got the position information of local representations during the local selection and local matching stages, the position of the corresponding image patches for each local representation can be easily found. We reproduce the local fusion procedure at the image level. Specifically, for the base image, we fuse the selected patches with similar patches from the reference images and replace them with the original image patches. Comparing with the target image of global reconstruction loss at the bottom right corner, our proposed local reconstruction loss presents fewer aliasing artifacts, which will help to improve the quality of generated images.

3.4. Objective Function

Let X denote the input images, $\hat{x} = G(X, \boldsymbol{\alpha})$ denotes the generated image, $c(X)$ denotes the label for X (only

available for seen categories). The generator G and discriminator D are optimized alternatively using the following loss functions in addition to the proposed $\mathcal{L}_{\text{local}}$.

Adversarial Loss. We use the hinge version GAN loss [18] to constrain the generator to generate realistic images that the discriminator cannot figure out:

$$\begin{aligned}\mathcal{L}_{\text{adv}}^{D} &= \max(0, 1 - D(X)) + \max(0, 1 + D(\hat{x})).\\ \mathcal{L}_{\text{adv}}^{G} &= -D(\hat{x}).\end{aligned} \tag{5}$$

Classification Loss. The classification loss follows ACGAN [18], where an auxiliary classifier is applied to classify the input images into the corresponding category. Specifically, the discriminator should correctly classify the real images and the generator is required to produce images while maintaining the same label with input images:

$$\begin{aligned}\mathcal{L}_{\text{cls}}^{D} &= -logP(c(X)|X).\\ \mathcal{L}_{\text{cls}}^{G} &= -logP(c(X)|\hat{x}).\end{aligned} \tag{6}$$

Therefore, the whole network is optimized end-to-end using the following objective function:

$$\begin{aligned}\mathcal{L}_{G} &= \mathcal{L}_{\text{adv}}^{G} + \lambda_{\text{cls}}^{G}\mathcal{L}_{\text{cls}}^{G} + \lambda_{\text{local}}\mathcal{L}_{\text{local}}^{G}.\\ \mathcal{L}_{D} &= \mathcal{L}_{\text{adv}}^{D} + \lambda_{\text{cls}}^{D}\mathcal{L}_{\text{cls}}^{D}.\end{aligned} \tag{7}$$

4. Experiments

4.1. Implementation

The encoder has one input convolutional block and four downsampling convolutional blocks. Each block has one convolutional layer followed by Leaky-ReLU activation and batch normalization. The decoder is symmetric to the structure of the encoder, which has four upsampling convolutional blocks and one output convolutional block. The feature size we use to perform LFM is 8×8. We use cosine similarity as the similarity function g in Eq.1. As for the discriminator, we adopt a similar network architecture in [14], which has four residual blocks as a feature extractor and two fully connected layers to evaluate realness and classification.

We use Adam optimizer to train the network $50,000$ iterations with a fixed learning rate of 0.0001 and another $50,000$ iterations with the learning rate linearly decayed to 0. In each iteration we randomly sample eight k-shot image generation tasks as one mini-batch to update the model. It takes about 36 hours to finish the training on one NVIDIA Tesla V100 GPU. We use real gradient penalty regularization [15] for training stability. The selection ratio η in LFM module is set to 0.5 by default. For hyper-parameters, we set $\lambda_{\text{cls}}^{G} = \lambda_{\text{cls}}^{D} = 1$ and $\lambda_{\text{local}} = 0.5$.

4.2. Evaluation Datasets

We use the following datasets for our experiments:

Flowers [17]. The Flowers dataset has 102 categories. We split it into 85 seen categories for training and 17 unseen categories for evaluation. Each category has a fixed number of 40 images.

Animal Faces [14]. The Animal Faces dataset contains 149 categories. We select 119 categories for training and 30 for evaluation with 100 images per category.

VGGFace [4]. For VGGFace dataset, we select 1802 categories for training and 552 for evaluation. The number of images for each category is also 100.

4.3. Baselines

We compare our method with several few-shot generation methods, including FIGR [5], GMN [3], DAWSON [13], DAGAN [2], MatchingGAN [8] and F2GAN [9]. To ensure a fair comparison, in our experiment, we implement a MatchingGAN model using the same network architecture, training strategy and hyper-parameters as the proposed method, which we denote as MatchingGAN†.

4.4. Quantitative Evaluation

The quantitative evaluation is conducted under a 3-way generation setting for both training and testing. Following MatchingGAN [8], we first train the model using images of the seen categories. Then we split the images of each unseen category into two parts, $\mathbb{S}_{in}$ and $\mathbb{S}_{real}$. We use the images in $\mathbb{S}_{in}$ to build a number of 128 3-shot image generation tasks, getting 128 generated images per category. The generated image set is denoted as $\mathbb{S}_{gen}$. We calculate the FID and LPIPS scores between $\mathbb{S}_{gen}$ and $\mathbb{S}_{real}$ to evaluate the generation. Furthermore, to evaluate the effectiveness of the LFM module, we replace the global fusion module in MatchingGAN† with our LFM module, which is recorded as MatchingGAN†+LFM. As shown in Table 1, introducing LFM brings a certain improvement to the baseline MatchingGAN† with lower FID and higher LPIPS, which means better quality and diversity can be achieved by using the LFM module. The proposed LoFGAN, introducing both LFM and $\mathcal{L}_{\text{local}}$, achieves the lowest FID and highest LPIPS on almost all of the three datasets, demonstrating the effectiveness of the proposed local reconstruction loss.

Figure 3 shows a comparison of generated images of our method with those from MatchingGAN† on all of the three datasets. In each row, we show six generated images for both of the methods. Since there is no base image in MatchingGAN†, we show two similar images for every input image (*i.e.,* the first two results of MatchingGAN† that look more like the first input image), and also show two images for each input images as the base image (*i.e.,* the first two results of LoFGAN are generated using the first image as the base image) for a clearer comparison. As can be seen, the outline of the images generated by MatchingGAN† is not clear enough, especially on the Flower dataset where

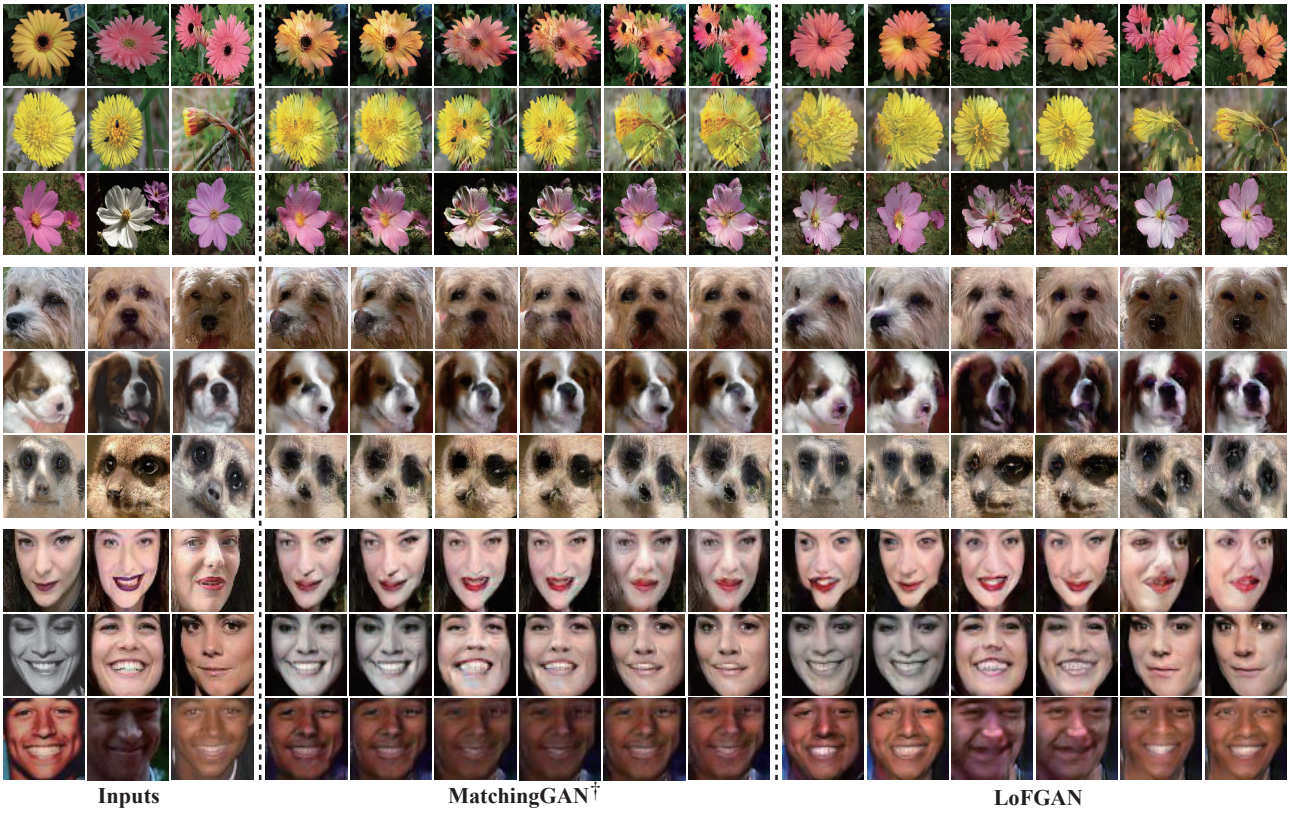

Figure 3: Images generated by MatchingGAN† and our proposed LoFGAN on Flowers, Animal Faces, and VGGFace. The first three columns are input images. We select two generated images for every base image.

Method	Type	Flowers		Animal Faces		VGGFace	
		FID(↓)	LPIPS(↑)	FID(↓)	LPIPS(↑)	FID(↓)	LPIPS(↑)
FIGR [5]	Optimization	190.12	0.0634	211.54	0.0756	139.83	0.0834
DAWSON [13]	Optimization	188.96	0.0583	208.68	0.0642	137.82	0.0769
DAGAN [2]	Transformation	151.21	0.0812	155.29	0.0892	128.34	0.0913
GMN [3]	Fusion	200.11	0.0743	220.45	0.0868	136.21	0.0902
MatchingGAN [8]	Fusion	143.35	0.1627	148.52	0.1514	118.62	0.1695
F2GAN [9]	Fusion	120.48	0.2172	117.74	0.1831	109.16	0.2125
MatchingGAN†	Fusion	139.90	0.3410	147.95	0.4695	27.93	0.2665
MatchingGAN†+LFM (ours)	Fusion	<u>86.59</u>	<u>0.3704</u>	<u>112.99</u>	**0.5024**	<u>22.99</u>	<u>0.2687</u>
LoFGAN (ours)	Fusion	**79.33**	**0.3862**	**112.81**	<u>0.4964</u>	**20.31**	**0.2869**

Table 1: Comparison of quantitative evaluation on FID and LPIPS. We quote the results of the first six methods from the F2GAN paper [9]. The best and second-best results are highlighted. † Results are re-implemented under the same setting for a fair comparison.

intro-class variances are relatively larger than human face. However, our method can produce much clearer images with fewer artifacts and various texture and color. And the local semantics are replaced by the proposed LoFGAN (*e.g.,* the mouth and the eyes are opened in the VGGFace dataset). Note that the color, texture, and background of the generated images are different in detail.

4.5. Visualization of the Learned Similarity

To verify whether the model correctly learns the semantic similarity in different images, we visualize the similarity maps between the base image and the reference image on unseen categories. After training the model on seen categories, we randomly select two images from one unseen cat-

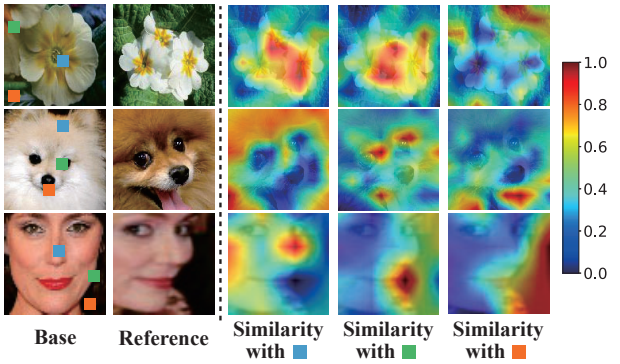

Figure 4: Visualization of the similarity maps. The first two columns are the base and reference images. The following columns show the similarity of the reference image with with red, orange and green points in the base image.

egory, taking one of them as the base image and the other as the reference image. Then we select some critical points on the base image and calculate their similarities with the reference image. For the flower image, we choose three different positions of the flower (*i.e.,* stamen, petal and background) in the base image. For the animal face image, we choose the forehead, eyes and mouth. For the human face image, we choose the nose, corner and background. Then we calculate the similarity between different positions in the base image and the whole reference image, checking whether the model can find the corresponding areas in the reference image. Figure 4 shows the visualization results. These results show that our method can find the most similar positions in the reference image. For example, the orange point in the first image represents a petal at the junction of two flowers, and the most relevant area is found in the middle area of the reference image. Through this way, our LoFGAN can make the fused feature semantically aligned.

4.6. Influence of the Selection Rate

The selection rate η is a hyper-parameter in our framework, which decides how many local representations should be replaced in the base feature. We visualize the generated images using different values of η in Figure 5. The first column shows three real images we select from one unseen category. We generate different images with η increasing from 0.1 to 1.0 using the same base image. Each row shows the output images from the same base image. It can be seen that the degree of the base image being modified gradually increases along with the growth of η. For example, when we choose the first image (a red flower with a black stamen) as the base image, the generated image is still red when η is low. However, it becomes a yellow flower with a green stamen when we increase the selection rate to 1.0. When we choose another flower as the base image, the similar result

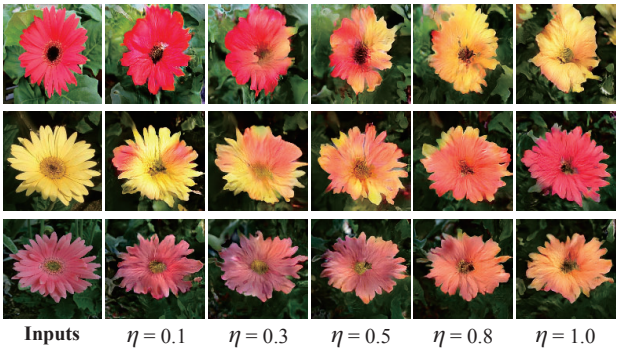

Figure 5: Images generated with different selection rates. The first column shows three input images. The following columns are generated images with η increasing from 0.1 to 1.0 when taking the first image in the same row as the base image.

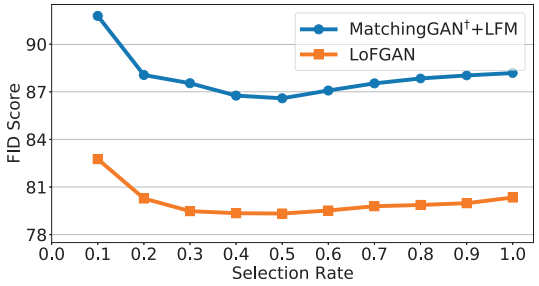

Figure 6: Comparison of FID score with selection rate changes from 0.1 to 1.0. The FID score decreases when η increases from 0.1 to 0.5, and rises slightly as η grows further to 1.0. A moderate selection rate can bring lower FID.

can be observed. Through this way, we can generate more diverse images by setting different values of η in our framework. Figure 6 shows the changes of FID score using different selection rates η. As can be seen, using either low or high selection rate will increase FID score. When η is low, the output images are almost the same as the base image, which means the diversity of generated images is not high enough. On the other hand, high selection rate may cause some unstable outputs, thus raising the FID score. We also compare the result with MatchingGAN†. It can be seen that the proposed LoFGAN outperforms MatchingGAN† with LFM in all of the settings, which further demonstrates the effectiveness of the proposed local reconstruction loss.

4.7. Augmentation for Classification

We also use the generated images to augment data for downstream image classification for the unseen categories. We split the unseen dataset into $\mathbb{D}_{train}$, $\mathbb{D}_{val}$ and $\mathbb{D}_{test}$, respectively. Following [8], a ResNet18 backbone is first initialized from the seen categories. We train a new classifier using the $\mathbb{D}_{train}$ without any augmentation, which is

Dataset	Flower	Animals	VGGFace
Standard	60.00	35.14	67.38
MatchingGAN†	60.39	35.90	65.17
MatchingGAN†+LFM	<u>62.75</u>	<u>36.10</u>	<u>68.20</u>
LoFGAN(ours)	**65.10**	**36.19**	**68.97**

Table 2: Comparison of top-1 accuracy on low-data image classification on Flower, Animal Faces and VGGFace. The best and second-best results are highlighted. The proposed LoFGAN achieves the most improvement over other methods.

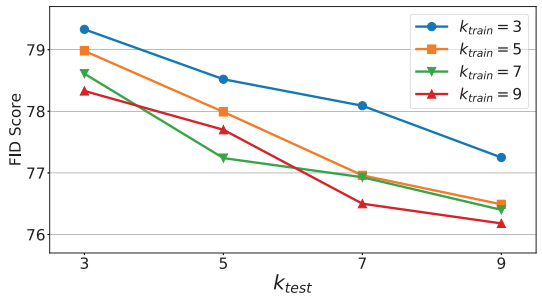

Figure 7: Comparison of few-shot image generation using different numbers of shots using the proposed LoFGAN. The FID score decreases with larger k_{train} and larger k_{test}.

referred as *Standard*. Then we use different few-shot image generation methods to augment the $\mathbb{D}_{train}$, and evaluate the resulted classification. For the Flower dataset, the data split for $\mathbb{D}_{train}$, $\mathbb{D}_{val}$ and $\mathbb{D}_{test}$ is 10:15:15 for each category. For Animal Face and VGGFace, we split the images into 30:35:35 for each category. We generate 30 images for Flower dataset and 50 images for Animal Face and VGGFace dataset for each unseen category for data augmentation.

The result is presented in Table 2. When the $\mathbb{D}_{train}$ only contains very few images, the few-shot generation models do help to improve the classification performance when compared with the result without augmentation. We achieve improvements of 2.75%, 0.96% and 0.82% by using MatchingGAN† with LFM, and 5.10%, 1.05% and 1.59% by using the proposed LoFGAN. The improvement corroborates the superiority of the proposed local fusion module and local reconstruction loss.

4.8. Comparison of Different Numbers of Shots

Although our model is trained under a 3-shot image generation setting for both training and test stages by default, it also supports input of different number of images. In this section, we evaluate the generation result of our model under different number of input images. Let k_{train} and k_{test} denote the number of input images for training and test stages, respectively. We train our LoFGAN with k_{train} in $\{3, 5, 7, 9\}$ on the Flower dataset, and then evaluate them using k_{test} in $\{3, 5, 7, 9\}$.

Figure 7 shows the FID score on using different combinations of k_{train} and k_{test}. As can be seen, increasing k_{test} brings lower FID scores under the same number of k_{train}. This may be because the increased number of k_{test} reduces the difficulty of generation tasks for the model. When we take more input images into the model, we can find more candidate local representations from a richer bank. More input images make it easier to find more matched positions, and to fuse more representations at the same time. Another interesting observation is that increasing k_{train} also helps to improve the generation. We calculate the average FID score for every k_{train}. With k_{train} increases from 3 to $5, 7$ and 9, the average FID changes from 78.30 to 77.85, 77.29, and 77.18, gaining an improvement of 0.45, 1.01 and 1.12 respectively. This result is intuitively consistent with relevant findings in few-shot image classification that more shots generally achieves better results.

5. Conclusion

In this paper, we propose *Local-Fusion Generative Adversarial Network (LoFGAN)*, a simple but effective way to generate more realistic and diverse images for few-shot image generation. Our contributions consist of a local fusion module which is based on local feature matching and replacing to produce semantically aligned deep features, and a local reconstruction loss which aligns corresponding semantic areas for the input images and better guides the model training. Experiments are conducted on three natural image datasets, showing that our LoFGAN has a better ability to generate realistic images with fewer aliasing artifacts and better diversity. Such improvement is achieved without introducing additional training parameters. Meanwhile, our approach still has limitations and there are problems to be tackled. For example, like other current fusion based methods, LoFGAN struggles with 1-shot image generation task, and the generated images will not be too different from the base images. Besides, there is still room for improvement on generation quality and diversity. We will explore these interesting issues in future work.

Acknowledgment

This work is supported by Science and Technology Innovation 2030 New Generation Artificial Intelligence Major Project No.2018AAA0100905, the National Natural Science Foundation of China No.61806092, Natural Science Foundation of Jiangsu Province No.BK20180326, the Fundamental Research Funds for the Central Universities No.02021438008, and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4432–4441, 2019.

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[3] Sergey Bartunov and Dmitry Vetrov. Few-shot generative modelling with generative matching networks. In *International Conference on Artificial Intelligence and Statistics*, pages 670–678, 2018.

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 67–74. IEEE, 2018.

[5] Louis Clouâtre and Marc Demers. Figr: Few-shot image generation with reptile. *arXiv preprint arXiv:1901.02199*, 2019.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 27:2672–2680, 2014.

[8] Yan Hong, Li Niu, Jianfu Zhang, and Liqing Zhang. Matchinggan: Matching-based few-shot image generation. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.

[9] Yan Hong, Li Niu, Jianfu Zhang, Weijie Zhao, Chen Fu, and Liqing Zhang. F2gan: Fusing-and-filling gan for few-shot image generation. In *ACM International Conference on Multimedia*, pages 2535–2543, 2020.

[10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[12] Yijun Li, Richard Zhang, Jingwan (Cynthia) Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 15897–15908. Curran Associates, Inc., 2020.

[13] Weixin Liang, Zixuan Liu, and Can Liu. Dawson: A domain adaptive few shot generation framework. *arXiv preprint arXiv:2001.00576*, 2020.

[14] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 10551–10560, 2019.

[15] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, pages 3481–3490. PMLR, 2018.

[16] Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

[17] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[18] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning (ICML)*, pages 2642–2651. PMLR, 2017.

[19] Esther Robb, Wen-Sheng Chu, Abhishek Kumar, and Jia-Bin Huang. Few-shot adaptation of generative adversarial networks. *arXiv preprint arXiv:2010.11943*, 2020.

[20] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 29:3630–3638, 2016.

[21] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9197–9206, 2019.

[22] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9332–9341, 2020.

[23] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *European Conference on Computer Vision (ECCV)*, pages 218–234, 2018.

[24] Miaoyun Zhao, Yulai Cong, and Lawrence Carin. On leveraging pretrained GANs for generation with limited data. In *37th International Conference on Machine Learning (ICML)*, volume 119, pages 11340–11351, 2020.

[25] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 7551–7562. Curran Associates, Inc., 2020.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE international conference on computer vision (ICCV)*, pages 2223–2232, 2017.