Learning to Memorize Feature Hallucination for One-Shot Image Generation

Yu Xie[1],Yanwei Fu[1]*, Ying Tai[2],Yun Cao[2],Junwei Zhu[2],Chengjie Wang[2]
[1]School of Data Science, Fudan University, [2]Youtu Lab, Tencent

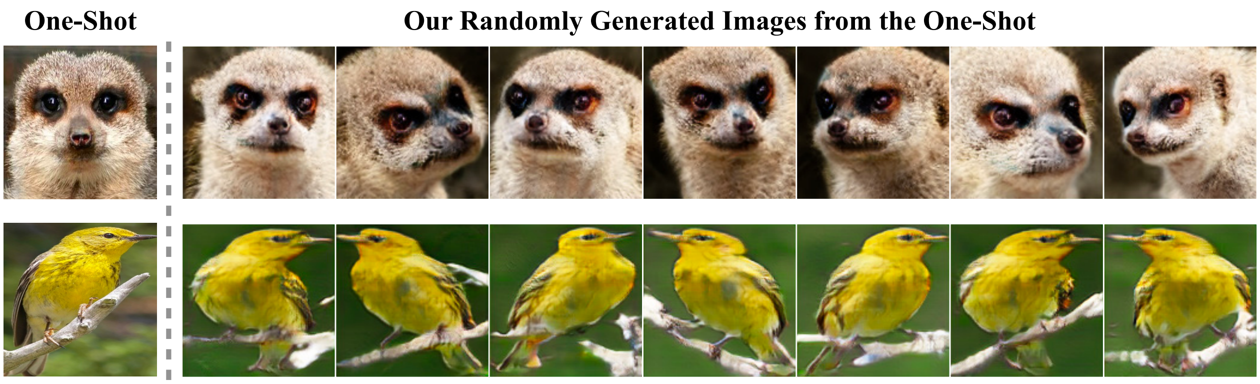

Figure 1. Task illustration. Given the category of only one available image, our model well synthesizes the images of that category.

Abstract

This paper studies the task of One-Shot image Generation (OSG), where generation network learned on base dataset should be generalizable to synthesize images of novel categories with only one available sample per novel category. Most existing methods for feature transfer in one-shot image generation only learn reusable features implicitly on pre-training tasks. Such methods would be likely to overfit pre-training tasks. In this paper, we propose a novel model to explicitly learn and memorize reusable features that can help hallucinate novel category images. To be specific, our algorithm learns to decompose image features into the Category-Related (CR) and Category-Independent(CI) features. Our model learning to memorize class-independent CI features which are further utilized by our feature hallucination component to generate target novel category images. We validate our model on several benchmarks. Extensive experiments demonstrate that our model effectively boosts the OSG performance and can generate compelling and diverse samples.

1. Introduction

As humans, our knowledge of concepts and the rich imagination ability may allow us to visualize or 'hallucinate' what the given image of the novel object would look like in other poses, viewpoints, or background, as shown in Fig. 1. Essentially, humans can robustly learn novel concepts with very little supervision, benefiting from the well-known ability of *learning to learn*. Inspired by such ability, previous works [6, 26, 28] study the recognition task in the low-data regime. In contrast, this paper addresses the task of One-Shot image Generation (OSG), which is defined as learning to synthesize images of a novel category with only one training example. Especially, the newly synthesized images should be visually similar to the given example. For example, given a new example of a novel target category in Fig. 1, the OSG task aims at generating new possible animal images by implicitly varying their key attributes, such as poses, viewpoints, and actions while crucially not changing the category of the example image.

Extensive efforts have been devoted to the one-shot image generation task. Specifically, some few-shot recognition models [31, 35] explore the generative models as data-augmentation methods, while these methods do not necessitate generating images of good visual quality. Then, to reduce the cost, researchers [16, 24] study training GANs using only a few images and produce high-quality images of good texture yet lacking semantic information. On the other hand, there are many transfer learning-based methods [14, 21, 33, 34] that transfer the pre-training model to the target task with only a few training samples. In these works, the models pre-trained on large datasets are adapted to some specific novel tasks or domains.

*Corresponding author. This work was supported in part by NSFC under Grant (No. 62076067), and SMSTM Project (2018SHZDZX01).

Figure 2. Given a single image of a panda, the category independent features pre-learned on the base dataset (prior knowledge) would be reused to hallucinate the new images. Thus the synthesized panda images have similar grass backgrounds or similar poses of open-mouth as some images in the base dataset.

Despite there are plenty of previous endeavors, our OSG task is still very difficult. The key challenges come from two folds. (1) There is insufficient training data as only one input image per class is available. (2) Pre-training (base) categories and target (novel) categories are dis-jointed, and the features learned on base are not necessarily generalizable for image synthesis of target categories.

To address these challenges, this paper proposes to explicitly explore features of hallucination. Our key insight is to learn features that are reusable and transferable from source categories to the target. For example in Fig. 2, given with only one example image of a panda, people can still imagine how a panda looks like in different backgrounds or poses. This is because people can maintain the prior knowledge about category-independent (class-agnostic) information, such as grass and open-mouth, and apply it to hallucinate the new panda images. This motivates us to exploit the Category-Independent (CI) and Category-Related (CR) features. Technically, it is inefficient to produce the labels to directly supervise the learning process of CI and CR features.

To this end, we present the model of learning to Memorize Feature Hallucination (MFH) that is capable of explicitly learning the CR and CI features via the image reconstruction process on the source/base dataset. The key component of our MFH is to introduce a memory module to learn and store the CI features. Specifically, our MFH is composed of two parts: Learning to Memorize (L2M), and Feature Hallucination (FeaHa). The L2M has the CI and CR encoders and the memory. The FeaHa is composed of a generator and discriminator.

More specifically, the CR encoder is to extract CR features with supervision from the category label, while the CI encoder projects CI features onto a memory from the given image. The memory serves as a dictionary of the CI features. To efficiently utilize the memory, a novel addresser network is presented in our work. Note that since there is no directly labeled supervision for the CI encoder, we introduce an implicit supervision strategy at the pairwise level. Particularly, given two different images from the same category, we assume these two images have the same CR, and yet different CI features. In the training stage, we randomly select two CI features from memory and combine them with the same CR feature; and we encourage the generator to synthesize the image differently. Simultaneously, we enforce the classifier to predict the label of the reconstructed image the same as the original category. We thus define such pairwise relationships as diversity loss to supervise our MFH, which is learned in an end-to-end manner. In the testing stage, we use the CR features from the input image and sample the CI features from the memory. Then we employ the generator to hallucinate the new images. Extensive experiments on two benchmarks validate the efficacy of our model.

Contributions We highlight several key contributions here: (i) We propose a novel method of learning to memorize feature hallucination for the task of OSG. (ii) Our MFH has the component of L2M and FeaHa. The L2M learns how to disentangle image features and repurpose the memory structure to preserve the CI features. By sampling from memory, our feature hallucination component can produce new images. (iii) To efficiently learn the class-independent CI features, we present a novel pairwise supervision strategy to help model explicitly learn features that can be reused in one-shot generation tasks. The learned CI features can consistently represent interpretable and meaningful concepts of various categories. (iv)Interestingly, we show that the newly synthesized images by our MFH can be directly employed as additional training instances, thus can boost the performance of one-shot classification.

2. Related Work

One-Shot Recognition It aims at rapidly generalizing to new recognition tasks containing an only one available sample. Methods of one-shot recognition can be roughly divided into these types: meta learning methods [26, 28] , metric learning based methods, optimization based methods [6] and so on. Beyond the recognition, this paper studies the one-shot image generation.

Image Generation There are many generative networks [5, 13, 38]. The basic problem to be solved is how to learn the data distribution and how to synthesize new pictures based on the learned distribution. The Generative Adversarial Networks(GANs) [7] is one of the most popular generative algorithms, with many well known unconditional models

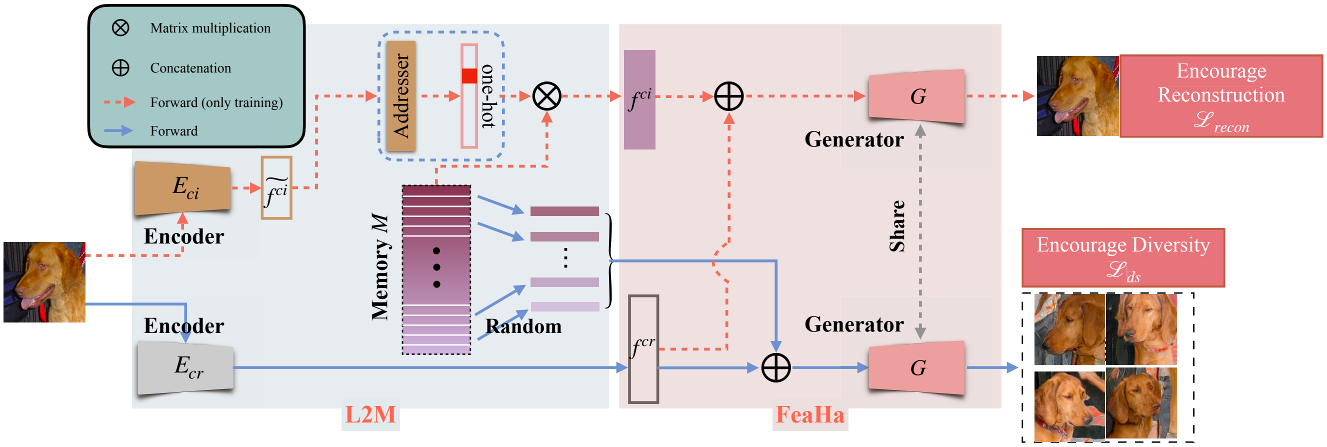

Figure 3. **Network Structure Diagram.** Our **M** reserves the CI features. In the inference phase, the generation network G uses randomly selected CI features in the **M** and image CR features f^{cr} in novel categories to generate diverse images.

including StyleGAN [30], BigGAN [3], and editing based methods such as GANs Inversion [1, 2]. Unfortunately, the vanilla GAN demands heavily rely on the training data, and typically is not ready to synthesize the categories of only one training sample. This inspires the exploration of few-shot GAN.

One-Shot Image Generation Recently, there has been some research on one-shot generation tasks [22, 39]. Unlike one-shot recognition tasks that usually introduce meta-learning, one-shot generation tasks are often based on transfer learning. Some methods [10, 24, 29] try to directly learn the image distribution information with one sample, where FastGAN [16] uses data-augmentation and self-supervised algorithms to avoid discriminator over-fitting under few-shot training samples and SinGAN [24] uses a multi-scale structure to learn the internal distribution information of image from a single sample. Another solution is based on transfer learning [14, 25, 34]. However, these types of methods often focus on the performance of the model in the novel domain rather than the novel category. Here we mainly introduce the methods applicable to the novel category. BAS [21] tries to solve the mode collapse problem that may occur when fine-tuning the network, it proposed to only update the batch normalization parameters. FinetuneGAN [31] extends BAS as a data-augmentation method to improve the performance of few-shot image recognition models. MineGAN [33] designs a miner network to mine the knowledge that is most beneficial to a specific dataset. Different from the above one-shot image generation methods, our model solves the one-shot image generation task from the perspective of disentangled learning and feature reuse. Our model does not need to be fine-tuned or retrained on the target category.

Memory Networks It [36] proposes to expand memory modules to maintain the long-term memory of networks. Neural Turing Machines [8] extend the capabilities of neural networks by coupling them to external memory modules. Such memory networks are widely used in many tasks, such visual question answering [12, 27, 37] and 3D point cloud segmentation [9], and open world recognition [18]. Different from these works, our framework is learned to memorize the features, which are reused for the hallucination task.

3. Method

Problem Definition The One-Shot image Generation (OSG) task assumes that we have the base/source dataset $D_{src} = \{\mathbf{x}_{src}, \mathbf{y}_{src}\}$ and a novel dataset $D_{nov} = \{\mathbf{x}_{nov}, \mathbf{y}_{nov}\}$. $\mathbf{x}_{src}$ and $\mathbf{x}_{nov}$ denote the train and test set respectively. The label sets are $\mathbf{y}_{src}$ and $\mathbf{y}_{nov}$. We denote the categories of source dataset and novel dataset as $\mathbf{C}_{src}$ and $\mathbf{C}_{nov}$, where $\mathbf{C}_{src} \cap \mathbf{C}_{nov} = \emptyset$. We take the general few-shot learning setting: there are plenty of labeled instances on D_{src}, and only one labeled instance per class on D_{nov}. Given one image $\mathbf{x}_i^{nov}, \in D_{nov}$, our MFH aims at generating more diverse images $\tilde{\mathbf{x}}^{nov}$, which should maintain the category unchanged. Notably, our task is different from the vanilla class conditioned GAN, as we only have one-shot image per class.

Overview We propose a novel network of learning to Memorize Feature Hallucination (**MFH**) for a one-shot image generation task, as summarized in Fig. 3. It has the novel components of Learning to Memorize (L2M), and Feature Hallucination (FeaHa). The key insight of our model is to map the image to the *Category-Related* and *Category-Independent* embedding spaces through two encoders $\mathbf{E}_{cr}$ and $\mathbf{E}_{ci}$. The L2M module enforces the pairwise supervision to learn CI features reusable among categories, which are further memorized and stored in the memory structure M. The FeaHa component samples from the memory, and hallucinate new images with additional CR features from the input exemplar. Our model is trained end-to-end and does not require fine-tuning during inference.

Figure 4. **Model generated images.** Here we show the performance of our model given an input image. Here we emphasize that none of the displayed species have appeared in the training set. The images synthesized by our MFH under the one-shot setting are more diverse than other competitors.

3.1. Learning to Memorize

The L2M component has the category-related $\mathbf{E}_{cr}$ and category-independent encoder $\mathbf{E}_{ci}$, which maps the input images to the CR & CI embedding spaces, respectively. The L2M further preserves the CI features into the memory module $\mathbf{M}$, which will be readable by the FeaHa module. To efficiently learn the memory, we present a novel addresser $\mathbf{R}$ network to read the information from $\mathbf{M}$ for reconstruction.

Encoder $\mathbf{E}_{cr}$ The $\mathbf{E}_{cr}$ calculates the mean of instances $\mathbf{x}_i$ from the same category. Particularly, given class $c \in C_{src} \cup C_{nov}$, we can encode the averaged features of its class,

$$f_c^{cr} = \frac{1}{K} \sum_{i=1}^{K} \mathbf{E}_{cr} \left(\mathbf{x}_i \cdot \mathbb{1}_c \left(y_i \right) \right), \tag{1}$$

Where f_c^{cr} represents the mean feature of class c in the prototypical-type embedding space; f_c^{cr} is CR feature. K represents number of samples; and we have $K = 1$ on the novel class. $\mathbb{1}_c : Y \to \{0, 1\}$ is an indicator function:

$$\mathbb{1}_c \left(y \right) = \begin{cases} 1, y = c \\ 0, y \neq c \end{cases} \tag{2}$$

Encoder $\mathbf{E}_{ci}$ Different from CR features from $\mathbf{E}_{cr}$, we take CI features from the memory module $\mathbf{M}$. Specifically, The CI encoder $\mathbf{E}_{ci}$ extracts feature $\widetilde{f_i^{ci}}$ from the input image $\mathbf{x}_i$, as $F^{ci} = \{\mathbf{E}_{ci} \left(\mathbf{x}_i \right)\}_{i=1}^{K}$. Here the encoded features F^{ci} are further utilized as the intermediate representations to construct the target CI features in the Memory $\mathbf{M}$.

Memory M and Addresser R The vanilla strategy of memory networks such as VQ-VAE [23] employs the nearest neighbor to read target information from Memory $\mathbf{M}$. However, it has some underlying disadvantages of very sensitivity to initialization and non-stationary to clustered neural activation in training our MFH framework. To this end, we present a novel Addresser $\mathbf{R}$ with the structure of a multi-layer perception. The input of Addresser $\mathbf{R}$ is $\widetilde{f^{ci}}$, and its output is a one-hot vector, representing the position of the target CI feature in Memory $\mathbf{M}$. To differentiablly learn the one-hot vectors in Memory $\mathbf{M}$, we employ the Gumbel-softmax [11] for optimization, it can be formulate as:

$$\pi_i = \frac{\exp\left(\left(R\left(\widetilde{f_i^{ci}} \right) + g_i \right) / \tau \right)}{\sum_{j=1}^{k} \exp\left(\left(R\left(\widetilde{f_i^{ci}} \right) + g_i \right) / \tau \right)} \tag{3}$$

where π_i is One-hot vector that refers to the position of the target CI feature in the Memory $\mathbf{M}$. g_i are independent and identically distributed (i.i.d) samples drawn from $\mathrm{Gumbel}\left(0, 1\right)$. The hyperparameter τ is the temperature coefficient in Gumbel-softmax.

Since π_i is a one-hot vector, we can easily use matrix multiplication to obtain the target CI feature from M. The final CI feature is:

$$f_i^{ci} = \pi_i \cdot \mathbf{M} \tag{4}$$

where $\pi_i \in \mathcal{R}^{1 \times n}$ and $\mathcal{M} \in \mathcal{R}^{n \times w}$, n represents the number of CI features stored in $\mathbf{M}$, and w is the dimension of CI feature.

3.2. Feature Hallucination

Feature hallucination contains two modules: generator and discriminator. The generator is subject to imagining new pictures according to the CI features in Memory M and CR features from novel categories while the discriminator is responsible for adversarial training.

Generation Network The generation network G is responsible for combining CR and CI features to generate the corresponding images. To facilitate such a purpose, we make a good design of the structure. Given a CR feature f^{cr} and a CI feature f_i^{ci} from memory $\mathbf{M}$, we first concatenate these two features as the input of the network. We learn to control Adaptive Instance Normalization (AdaIN) operations after each convolution layer of the synthesis network G. Note that different from the AdaIN in StyleGAN, we utilize different conditions for AdaIN to help the disentangled learning: before the resolution of the feature map reaches 32×32, we use the CI feature f_i^{ci} as a condition for AdaIN, and we utilize the CR feature f^{cr} for the rest of generator network.

The reason for our design is to consider that the generated image needs to maintain the same category as the input image, so we only employ the CR features to calculate the AdaIN parameter in the second half of the generated network.

$$\mathbf{x}_i^{gen} = G\left(f_c^{cr}, f_i^{ci}\right) \tag{5}$$

where $\mathbf{x}_i^{gen}$ indicates generated images, f_i^{ci} refers to CI features selected from Memory $\mathbf{M}$.

In the inference stage, we randomly sample the FeaHa components from Memory $\mathbf{M}$ and combine it with the CR feature of the One-shot image as the input of the Generator to imagine new images.

Discriminator D Considering that we have strict requirements on the category of the generated image, it requires to be consistent with the input image category. Thus we use the discriminator structure of cGAN [19, 20].

3.3. Loss Functions and Training Strategy

Here we give a detailed description of the loss functions and training strategy of the model. During the training process, we only use the images in the source dataset. For simple notation, assuming that in one forward process, we randomly sample one image $\mathbf{x}$ from one category $\mathbf{y}$. We train the OSG task by solving a minimax optimization,

$$\operatorname{minmax} \mathcal{L}_{GAN} + \lambda_R \mathcal{L}_R + \lambda_{ds}\mathcal{L}_{ds} + \lambda_{cb}\mathcal{L}_{cb} \tag{6}$$

where $\mathcal{L}_{GAN}$, $\mathcal{L}_R$, $\mathcal{L}_{ds}$, and $\mathcal{L}_{cls}$ are the GAN loss, the content image reconstruction loss, diversity loss and category balance loss individually.

The **GAN loss** is a conditional one given by

$$\begin{aligned}\mathcal{L}_{GAN}(G, D) =& \mathbf{E}_{\mathbf{x},\mathbf{y}}\left[-\log D(\mathbf{x}, \mathbf{y})\right] \\ &+ \mathbf{E}_{\mathbf{x},\mathbf{y}}\left[\log(1 - D(\mathbf{x}^{gen}, \mathbf{y}))\right]\end{aligned} \tag{7}$$

The loss is computed only using the corresponding binary prediction score of the class, the GAN loss here includes classification supervision.

The **Reconstruction Loss** L_R is to help the network better learn how to generate images. According to the input $\mathbf{x}$, we can obtain its category-independent feature $f_{\mathbf{x}}^{ci}$ and category-related features $f_{\mathbf{x}}^{cr}$ respectively. The loss $\mathcal{L}_R$ encourages the generator G to reconstruct the input image $\mathbf{x}$ based on $f_{\mathbf{x}}^{ci}$ and $f_{\mathbf{x}}^{cr}$. That is

$$\mathcal{L}_R = \mathbf{E}_{\mathbf{x}}\left[\left\|\mathbf{x} - G\left(f_{\mathbf{x}}^{cr}, f_{\mathbf{x}}^{ci}\right)\right\|_1\right] \tag{8}$$

Reconstruction loss is the key to ensure that the model can synthesize high-quality images.

Algorithm 1 PairWise Supervision

Require: images $\mathbf{x}_i$ with label $\mathbf{y}_i$

1: Sample f_a^{ci} and f_b^{ci} from $\mathcal{M} \in \mathcal{R}^{n\times w}$ ▷ Randomly sample two CI features from Memory $\mathbf{M}$

2: $f^{cr} = \mathbf{E}_{cr}(\mathbf{x}_i)$ ▷ Extract Category-Related feature from $\mathbf{x}_i$

3: $\mathbf{x}_a^{gen} = G(f^{cr}, f_a^{ci})$ ▷ Combine f_a^{ci} and f^{cr} to generate corresponding image

4: $\mathbf{x}_b^{gen} = G(f^{cr}, f_b^{ci})$ ▷ Combine f_b^{ci} and f^{cr} to generate corresponding image

5: Classify the generated images, $Cls(\mathbf{x}_a^{gen}) = Cls(\mathbf{x}_b^{gen}) = \mathbf{y}_i$ ▷ The categories of the two randomly generated images need to be consistent with the input images $\mathbf{x}_i$. The classifier is included in the discriminator.

6: Calculate $\alpha - \mathbf{E}_{\mathbf{x}}\left[\|G(f_{cr}, m_1) - G(f_{cr}, m_2)\|_1\right]$ ▷ Encourage different CI features to get different images

---

Here we introduce our pairwise **Diversity Loss** in detail, which is the key to supervising our MFH to explicitly extract reusable features. According to the previous introduction, we randomly sample two category-independent features f_a^{ci}, f_b^{ci} from memory $\mathbf{M}$, then combine them with $f_{\mathbf{x}}^{cr}$ as the input of the Generator to generate two images $\mathbf{x}_a^{gen}$, $\mathbf{x}_b^{gen}$. $\mathcal{L}_{ds}$ encourages the generated two images to be significantly different. $\mathcal{L}_{ds}$ can be formulated as:

$$\mathcal{L}_{ds} = \alpha - \mathbf{E}_{\mathbf{x}}\left[\left\|G\left(f_{\mathbf{x}}^{cr}, f_a^{ci}\right) - G\left(f_{\mathbf{x}}^{cr}, f_b^{ci}\right)\right\|_1\right] \tag{9}$$

where α is hyperparameter for controlling the diversity.

Finally, $\mathcal{L}_{cb}$ is used to make the distribution of each CI features as balanced as possible.

$$L_{cb} = \mathrm{KL}\left(\pi_i \parallel q(\pi)\right) \tag{10}$$

where KL is the Kullback-Leibler divergence, and $q(\pi)$ is assumed to be uniformly distributed.

The pseudo code of our proposed module is in Alg 1, which show just how easy it is to implement our pairwise diversity supervision.

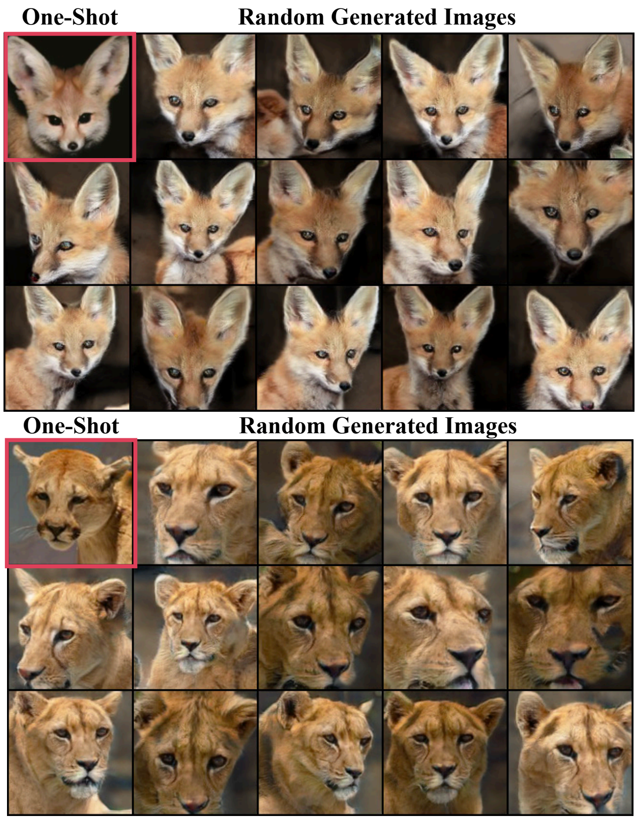

Figure 5. Visualization of different category images combined with the same CI features. One-Shot images are marked with a red box, and the rest are images synthesized by the model.

4. Experiments

AnimalFace [17]. This dataset is constructed by using images from ImageNet [4] dataset. The images come from 149 carnivorous animals of 119 source/seen classes and the 30 target/unseen classes. This dataset contains a total of 117574 images.

NABirds(NAB) [32]. A high-quality dataset containing 48,562 images of North American birds with 555 categories, part annotation, and bounding boxes [32]. We evaluate whether our model as a data-augmentation method improves the performance of the one-shot classification model on this dataset. We follow MetaIRNet [31] setting, and split NAB with a portion of train:test=3:1.

Implementation We use Adam with learning rate (lr)=0.0001, β_1 = 0.001 and β_2 = 0.999 for all methods. Spectral normalization is applied to the discriminator. The final generator is a historical average version of the intermediate generators where the update weight is 0.001. We train the model for 150,000 iterations in total. Each training batch consists of 64 content images, which are evenly distributed on a DGX machine with one 3090 GPU, each with 24GB RAM. The resolution of the images we generated and input is 128×128. For Memory $\mathbf{M}$ in the network, we set

Dataset	Method	FID
AnimalFace [17]	MineGAN [33]	94.25
	FastGAN [16]	80.23
	FinetuneGAN [31]	91.39
	BAS [21]	102.31
	ours	**75.28**
NABirds [32]	MineGAN [33]	79.28
	FastGAN [16]	59.64
	FinetuneGAN [31]	75.56
	BAS [21]	84.56
	Ours	**42.24**

Table 1. Comparison with other methods in the one-shot setting on AnimalFace and Nab datasets.

50 memory sizes for both datasets. The details of MFH are in the supplementary.

Evaluation Protocol Here we evaluate our model from two perspectives, which are the quality of images generated by the model and whether the generated images are helpful for one-shot classification tasks. For the quality of the generated image, we employ Frechet Inception Distance (FID) to measures the similarity between two sets in the embedding space. FID is widely used to measure both quality and diversity of the generated images. For each dataset, we let the model generate 50 images for each category and randomly sample 50 images from each test category to calculate the FID with the synthesized image. To evaluate whether our method is helpful for one-shot classification tasks, we follow the setting in MetaIRNet [31] and use our method as a data-augmentation strategy to expand support set. For fair comparison, we use ProtoNet [26] as the base classifier of other data augmentation baselines.

4.1. Main Results and Discussion

Quantitative Results We compare our method with other methods in the one-shot setting on AnimalFace and Nab datasets. FinetuneGAN [31], MineGAN [33] and BAS [21] are first trained on ImageNet [4] and then adapt model on a one samples in the target category by fine-tuning the weights of the model, FastGAN [16] use a self-supervised algorithm to ensure that the discriminator will not overfit even with few samples. Here our main comparison methods are to make the generative network generalize to the novel category, and some methods [14, 22] whose purpose is to generalize to the novel domain are not included in our comparison. From Tab. 1, our FID is much lower than other competitors.

From Tab. 2, we can see that the data-augmentation method for comparison includes both traditional image transformations, such as Gaussian noise and flip as well as generative networks FinetuneGAN, introduced by MetaIRNet [31], is based on the BAS model extension). When

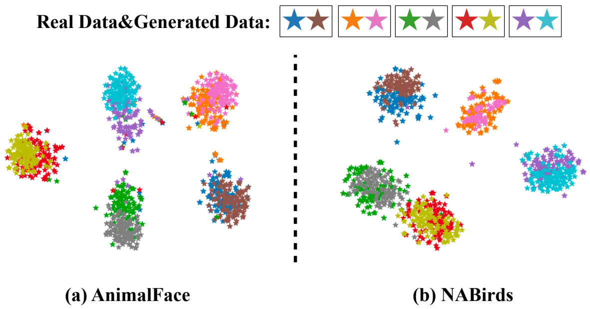

Figure 6. We visualize the tSNE plot of generated images and real images. It is clear that the images synthesized by our model have high diversity while keeping the category labels accurate.

Method	Data Augmentation	NABirds Acc.↑
ProtoNet	-	77.93±0.67
ProtoNet	FinetuneGAN	76.28±0.63
ProtoNet	Flip	78.72±0.64
ProtoNet	Gaussian	77.94±0.67
ProtoNet	Ours	79.02±0.61
MetaIRNet	FinetuneGAN	79.21±0.63
MetaIRNet	FinetuneGAN, Flip	79.52±0.62
MetaIRNet	Ours	**82.98±0.60**

Table 2. Results of on 5-way 1-shot tasks from NABirds with ImageNet pre-trained ResNet18.

using our model as a data-augmentation method, it can be improved by about 4 points compared to the basic protonet. Our model as a data-augmentation strategy is also significantly better than other data-augmentation methods. In order to better show why our model can improve the performance of the one-shot classification model, In Fig. 6 we use tSNE to visualize the distribution of our generated samples and realistic samples in the embedding space.

Qualitative Analysis As seen from Fig. 4 in the one-shot setting, our model can produce diverse and high-quality samples. When selecting different CI features from memory back and combining them with the CR feature of the one-shot image, our model generates more diverse images while keeping the same category of the generated image and the input image. This shows that our model disentangles “category-independent” and “category-related” features well. FinetuneGAN can only synthesize images similar to the image used for training, and the quality of the synthesized images is also very poor. FastGAN performs better than FinetuneGAN but the images it generated still lack diversity. Our model can generate more diverse images while keeping the object category unchanged. In Fig. 5, It can be seen from the experimental results that images synthesized by combining different “category-related features” with the same “category-independent features” will have the same mode (such as “looking to the left”) while retaining the same category features as the input image. This further reveals the insights of our model. We use a crowdsourcing platform to invite 50 users who are unknown to our project and make binary voting of the quality and diversity of the images synthesized by different methods. Each user randomly gives one synthesized image of each method. We summarize the results as shown in tab 3, our methods have obtained more user votes on both evaluation indicators. In Fig. 7, We can see that the model has learned how to change the background and posture of the object. In other words, the model of unsupervised learning has characterized the background and posture as two key features shared among categories. Such results are reasonable.

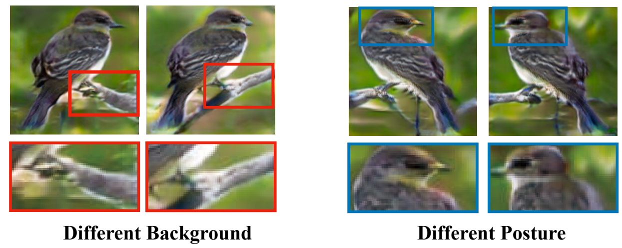

Figure 7. We show that in datasets with similar backgrounds, such. as NABirds, our model can learn not only features such as pose but also some background features that can be shared.

	Dataset	Methods		
		Ours	FinetuneGAN	FastGAN
Im.Q	AnimalFace	32	5	13
	NABirds	28	4	18
Im.D	AnimalFace	42	2	6
	NABirds	38	1	11

Table 3. User study. We invite 50 users to vote by the generated image quality(Im.Q) and generated image diversity(Im.D).

ds-loss	gumbel-softmax	AnimalFace	NABirds
	✓	90.54	71.36
✓		87.63	65.72
✓	✓	**75.28**	**42.24**

Table 4. Ablation of MFH. Here we mainly analyze the two most important components, diverse loss and gumbel-softmax.

5. Ablation Study

Here we mainly discuss the two modules of the model. One is L2M module. In the previous introduction, we explained why we choose Gumbel softmax for Addresser **R** instead of the K-means [15]. In ablation study, we will verify it through experiments. The other one is the design of the loss function, especially the influence of $\mathcal{L}_{ds}$ on model performance. Finally, we will also give the failure case of the network and analyze the reasons.

Effect of the Gumbel Softmax In this paper, we use a classification network to directly predict the CI features' ad-

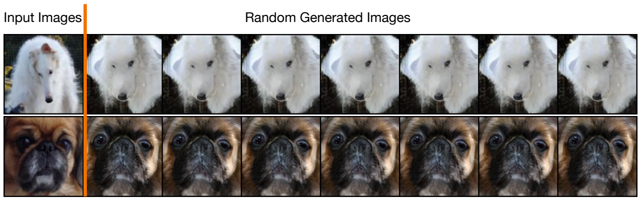

Figure 8. Results of using K-means [15] to replace Gumbel softmax. From left to right are the input images and the network randomly generated images.

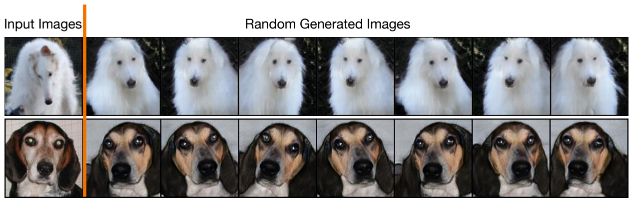

Figure 9. After removing the diverse loss $\mathcal{L}_{ds}$, the performance of the model. From left to right are the input images and the network randomly generated images.

dresses to which each sample belongs, and Gumbel-softmax as a differentiable argmax operation. Here we replace the Gumbel-softmax operation with the K-means to show why we choose Gumbel-softmax instead of K-means. In the training process, we calculate the distance between the feature of the source sample and the memory items. And through the stop gradient method in VQ-VAE to update each memory item.

As shown in Fig. 8, When Addresser **R** uses K-means instead of Gumbel Softmax, it is easy to cause multiple CI features to collapse into one CI feature, which causes the generation network G to be insensitive to the input from the memory bank. This is why no matter which CI feature we select, the output of the generated network is the same, and the generated images lack the diversity of content. After replacing gumble-softmax with k-means, in Tab. 4, the FID score has also risen sharply, which indicates that the effect and diversity of the model's image generation have deteriorated.

Effect of the Design of Loss Function In order to make the images generated by the network have diversity, and keep its category consistent to the input images. Here we remove the diversity loss to understand the impact of the two losses on the network generation performance. As shown in Fig. 9, When we remove $\mathcal{L}_{ds}$, although not all generated images are the same, the diversity of generated images is still significantly reduced, multiple CI features have overlapped. As shown in Tab. 4, When we remove the diverse loss, the FID score performance of the model has greatly increased on the two different data sets of AnimalFace and NABirds. This shows that the diversity of generated images is significantly reduced.

Interpolate between CI Features Although our network is trained to set as a hyperparameter the number of CI features in memory **M**. Such CI features are discrete variable. Here we show that we can generate more images by interpolating between the two CI features. Specifically, we randomly select CI features from the memory; and then we perform linear interpolation between them. As shown in Fig. 10, we can see that the intermediate CI features can generate meaningful results.

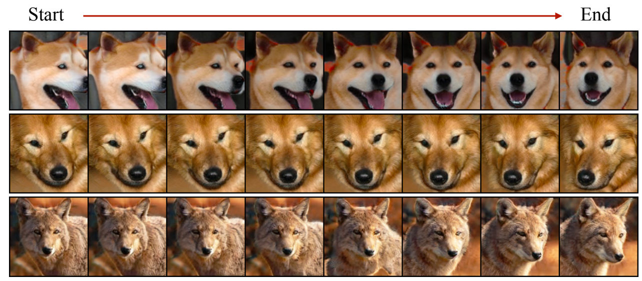

Figure 10. We randomly select two CI features and interpolate from one to another. Our model can generate meaningful intermediate results by using these interpolated CI features.

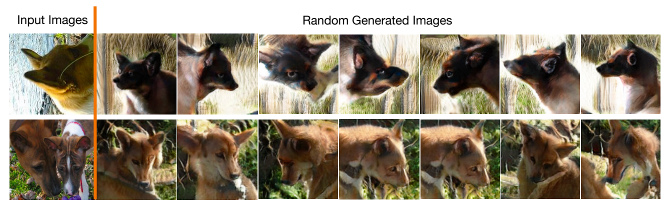

Figure 11. We show that some failure cases are caused by strange poses and multi-object occlusion.

Failure Case Analysis Fig. 11 shows several failure cases generated by our model.The reason for the failure case may be that there are cases in the image that have not been seen in the training, such as multiple animals and strange poses and so on.

6. Conclusion

In this paper, we introduce a novel framework to solve one-shot image generation problems. We propose a generative model to learn and memorize the category-independent features on the source, so as to generate more data based on this learned knowledge when given the one-shot example. Specially, we propose a pairwise diversity supervision strategy to help the model learn category-independent features explicitly. We show that while given only one example of a new category, our network can still generate plausible and diverse new images whose category is strictly consistent with the input sample. We validate our model on several benchmarks and achieve state-of-the-art generation performance.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019. 3

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8305, 2020. 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016. 2

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 1, 2

[7] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 2

[8] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 3

[9] Tong He, Dong Gong, Zhi Tian, and Chunhua Shen. Learning and memorizing representative prototypes for 3d point cloud semantic and instance segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 564–580. Springer, 2020. 3

[10] Tobias Hinz, Matthew Fisher, Oliver Wang, and Stefan Wermter. Improved techniques for training single-image gans. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1300–1309, 2021. 3

[11] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[12] Mahmoud Khademi. Multimodal neural graph memory networks for visual question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7177–7188, 2020. 3

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[14] Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. Few-shot image generation with elastic weight consolidation. *arXiv preprint arXiv:2012.02780*, 2020. 1, 3, 6

[15] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003. 7, 8

[16] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2020. 1, 3, 6

[17] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10551–10560, 2019. 6

[18] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 3

[19] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 5

[20] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 5

[21] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2750–2758, 2019. 1, 3, 6

[22] Utkarsh Ojha, Yijun Li, Jingwan Lu, Alexei A Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10743–10752, 2021. 3, 6

[23] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 4

[24] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4570–4580, 2019. 1, 3

[25] Mohamad Shahbazi, Zhiwu Huang, Danda Pani Paudel, Ajad Chhatkuli, and Luc Van Gool. Efficient conditional gan transfer with knowledge propagation across classes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12167–12176, 2021. 3

[26] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017. 1, 2, 6

[27] Zhou Su, Chen Zhu, Yinpeng Dong, Dongqi Cai, Yurong Chen, and Jianguo Li. Learning visual knowledge memory networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7736–7745, 2018. 3

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 1, 2

[29] Vadim Sushko, Jurgen Gall, and Anna Khoreva. One-shot gan: Learning to generate samples from single images and videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2596–2600, 2021. 3

[30] S. Laine T. Karras and T. Aila. A style-based generator architecture for generative adversarial networks. *CVPR*, 2019. 3

[31] Satoshi Tsutsui, Yanwei Fu, and David Crandall. Meta-reinforced synthetic data for one-shot fine-grained visual recognition. *arXiv preprint arXiv:1911.07164*, 2019. 1, 3, 6

[32] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015. 6

[33] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9332–9341, 2020. 1, 3, 6

[34] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 218–234, 2018. 1, 3

[35] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018. 1

[36] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 3

[37] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406. PMLR, 2016. 3

[38] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016. 2

[39] Miaoyun Zhao, Yulai Cong, and Lawrence Carin. On leveraging pretrained gans for generation with limited data. In *International Conference on Machine Learning*, pages 11340–11351. PMLR, 2020. 3