D^{2}GAN: A FEW-SHOT LEARNING APPROACH WITH DIVERSE AND DISCRIMINATIVE FEATURE SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

The rich and accessible labeled data fuel the revolutionary success of deep learning. Nonetheless, massive supervision remains a luxury for many real applications, boosting great interest in label-scarce techniques such as few-shot learning (FSL). An intuitively feasible approach to FSL is to conduct data augmentation via synthesizing additional training samples. The key to this approach is how to guarantee both discriminability and diversity of the synthesized samples. In this paper, we propose a novel FSL model, called D^{2}GAN, which synthesizes Diverse and Discriminative features based on Generative Adversarial Networks (GAN). D^{2}GAN secures discriminability of the synthesized features by constraining them to have high correlation with real features of the same classes while low correlation with those of different classes. Based on the observation that noise vectors that are closer in the latent code space are more likely to be collapsed into the same mode when mapped to feature space, D^{2}GAN incorporates a novel anti-collapse regularization term, which encourages feature diversity by penalizing the ratio of the logarithmic similarity of two synthesized features and the logarithmic similarity of the latent codes generating them. Experiments on three common benchmark datasets verify the effectiveness of D^{2}GAN by comparing with the state-of-the-art.

1 INTRODUCTION

One of the key triggers for the recent flourish of deep learning in various tasks is the rich and accessible labeled data. However, in many specific real applications, only limited labeled data are available. This incurs the investigation of few-shot learning (FSL), i.e., to learn from a few labeled data. To combat with deficiency of labeled data, some FSL methods resort to enhance the discriminability of the feature representations such that a simple linear classifier learned from few labeled samples can reach satisfactory classification results (Vinyals et al. (2016); Snell et al. (2017); Triantafillou et al. (2017)). Another category of methods investigate how to quickly and effectively update a deep neural network using a few labeled data (Finn et al. (2017); Li et al. (2017); Ravi & Larochelle (2017); Munkhdalai & Yu (2017)). Alternatively, a third group of methods aim to augment the labeled samples based on the provided ones (Chen et al. (2019a); Gao et al. (2018)).

This paper approaches FSL in the line of data augmentation. Whatever data augmentation technique is used, we usually need to use the augmented data to train a classifier. Therefore, whether the augmented samples can encapsulate the discriminative information of the classes becomes pivotal to secure the discriminability of the classifier. On the other hand, the decision boundary of a classifier can only be determined precisely when training samples exhibit sufficient intra-class variance. Thus, the diversity of the augmented samples is also of a crucial role. This is in fact the essential motivation of investigating data augmentation for FSL, as the few labeled samples can capture only limited intra-class variance.

We propose a novel conditional GAN based FSL model, called D^{2}GAN, which synthesizes new features conditioned on those of the few labeled samples. To secure discriminability of the synthesized features, D^{2}GAN incorporates a novel classification regularizer to encourage the generator to synthesize features of high correlation with features of real samples from the same class while of low correlation with those from different classes.

It is more complicated to ensure diversity of the synthesized features, as conditional GANs are notoriously susceptible for mode collapse problem. This is due to the use of usually high dimensional and structured data as the condition tends to make the generator likely ignore the latent code, which controls diversity. To avoid this problem, we propose a novel anti-collapse regularizer, which gives high penalty on the case where mode collapse more likely occurs. It is derived from the observation that noise vectors that are closer in the latent code space are more likely to be collapsed into the same mode when mapped to feature space. We therefore directly penalize the ratio of the logarithmic similarity of two synthesized feature vectors and the logarithmic similarity of the two noise vectors generating them.

With discriminative and diverse feature synthesized, we can get highly effective classifiers and accordingly appealing recognition results. In summary, the contributions of this paper are as follows:

- We propose a novel conditional GAN based FSL model which augments the labeled samples by synthesizing fake features conditioned on those of the few labeled samples.
- We propose two regularizers to train our novel GAN model. The first one secures feature discriminability by requiring the synthesized feature classifying well other samples of the same classes. The second ensures feature diversity by penalizing the ratio of the logarithmic similarity two synthesized feature vectors and the logarithmic similarity of the two noise vectors generating them.
- The proposed method reaches the state-of-the-art on three common benchmark datasets.

2 Related Work

We review the related work about general data augmentation, Generative Adversarial Networks (GAN) and Few-Shot Learning (FSL).

Data augmentation. Standard data augmentation techniques include flipping, rotating, adding noise and randomly cropping images, adding Gaussian perturbation, transforms, and rescaling of training images (Chen et al. (2019b)). However, adding noise or jittering on the original images is particularly suspect to visual similarity with the original images. This is undesirable for FSL as we only have a very limited number of images to be based on; synthesizing quality of any single image is vital for the final recognition results. Recently, some more sophisticated data augmentation techniques have been proposed to augment labeled samples for FSL, including hallucinating (Wang et al. (2018)), shrinking and hallucinating features (Hariharan & Girshick (2017)), composing synthesized representations (Yu & Kristen (2017); Chen et al. (2019b)), or using GANs (Gao et al. (2018); Zhang et al. (2018)).

GAN. GAN was originally proposed as a method for image synthesis and has achieved impressive results (Goodfellow et al. (2014)). However, GAN is also known for its instability in training and suffers from the mode collapse problem (Arjovsky et al. (2017); Gulrajani et al. (2017)). In order to mitigate these problems and improve the quality of synthetic samples, many methods have been proposed. WGAN (Arjovsky et al. (2017)) and WGAN-GP (Gulrajani et al. (2017)) propose to optimize GAN on an approximate Wasserstein distance by enforcing 1-Lipschitz smoothness. LSGAN (Mao et al. (2017)) offers a simple yet effective solution by replacing the cross-entropy loss of GAN with a least square loss that pushes the scores of real and fake samples to different decision boundaries so that the gradient never vanishes even when two distributions are totally disjoint.

FSL. Regarding to the perspective of approaching FSL, existing algorithms can generally be classified as three categories. The first category of methods aim to enhance the discriminability of feature representations of images. With this goal, a number of methods address it via deep metric learning and learn deep embedding models that output discriminative features for any given images (Vinyals et al. (2016); Snell et al. (2017); Triantafillou et al. (2017); Sung et al. (2018)). The difference lies in the loss functions used. A more common category of algorithms strive for enhancing the flexibility of the model such that it can be readily updated using few labeled samples. These methods either aim to optimize a meta-learned classifier to be easily fine-tuned using the small-scale support set provided (Ravi & Larochelle (2017); Finn et al. (2017); Li et al. (2017); Munkhdalai & Yu (2017); Nichol et al. (2018)), or resort to neural network generation which trains meta-learning networks to adaptively generate some components of another neural network using the support set (Qiao et al. (2018); Gidaris & Komodakis (2018); Rusu et al. (2019)). The last category of methods combat the

data deficiency problem directly, aiming to effectively augment labeled data. Some methods try to employ additional examples by some form of transfer learning from external data (Ren et al. (2018); Wang & Hebert (2016)); others aim to synthesize new data based on existing labeled ones by various data augmentation techniques tailored for FSL problem (Chen et al. (2019a); Schwartz et al. (2018); Hariharan & Girshick (2017); Yu & Kristen (2017); Gao et al. (2018); Zhang et al. (2018)).

This paper builds on top of the WGAN model and proposes a novel GAN architecture to augment labeled samples by synthesizing fake features conditioned on those of the real labeled ones. Our method is significantly different from the peer GAN based FSL algorithms (Zhang et al. (2018); Gao et al. (2018)). (Zhang et al. (2018)) serves as a plugin for existing FSL classifiers, aiming to enhance their capability by using GAN to generate fake data between the manifolds of different real data classes. It does not use the synthesized samples for classification at all. Our method instead is independent from existing FSL approaches and we use GAN to augment the labeled data set on which a standard (rather few-shot) classifier can be learned. (Gao et al. (2018)) has the same objective as us for synthesizing samples. However, it applies cycle-consistency constraint on the synthesized samples to guarantee discriminability, which only enforces a synthesized sample to reconstruct well the conditioned sample on which it is generated; it neglects to exploit the inter-class competing information, which shall be crucial for FSL. Our method instead requires synthesized features to have high correlation with real features of the same classes and low correlation with those of different classes. Class discrimination information is thus fully utilized, ensuring discriminative features synthesized. Further, (Gao et al. (2018)) secures diversity of synthesized samples for a novel class by a covariance constraint derived from its most similar base classes. We instead achieve this goal by an anti-collapse regularizer. Our model also shares some similarity with (Xian et al. (2018)), a recent GAN based zero-shot learning algorithm, which synthesizes features based on the semantic description of classes. Apart from the different sources conditioned to synthesis new features, we further propose the novel classification regularizer and anti-collapse regularier to secure us diverse and discriminative features based on the few labeled ones.

3 ALGORITHM

In this section, we will first introduce some preliminary knowledge about Wasserstein GAN. Then, we will elaborate the details how we build our D^{2}GAN model on top of it.

3.1 WASSERSTEIN GAN

Generative Adversarial Networks (GANs) are a powerful class of generative models that cast generative modeling as a game between two networks: a generator network produces synthetic data given some noise source and a discriminator network discriminates between the generators output and real data. GANs can produce very visually appealing samples, but are often hard to train. (Arjovsky et al. (2017)) provides an analysis of the convergence properties of the value function being optimized by GANs and proposes the Wasserstein GAN (WGAN), which leverages the Wasserstein distance to produce a value function which has better theoretical properties than the original. We adopt the improved WGAN model (Gulrajani et al. (2017)), which optimizes the following min-max problem,

$$\min_G \max_D \quad \mathop{\mathbb{E}}_{\tilde{\mathbf{x}}\sim\mathbb{P}_g}[D(\tilde{\mathbf{x}})] - \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathbb{P}_r}[D(\mathbf{x})] + \lambda \mathop{\mathbb{E}}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2], \tag{1}$$

where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the model distribution defined by $\tilde{\mathbf{x}} \sim G(\mathbf{z})$, with $\mathbf{z} \sim p(\mathbf{z})$ randomly sampled from some noise distribution p. $\mathbb{P}_{\hat{\mathbf{x}}}$ is defined by sampling uniformly along straight lines between pairs of points sampled from the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$, i.e., $\hat{\mathbf{x}} = \alpha\mathbf{x} + (1-\alpha)\tilde{\mathbf{x}}$ with $\alpha \sim U(0,1)$. The first two terms of Eq. (1) approximate the Wasserstein distance, while the third term penalizes on the gradient norm of $\hat{\mathbf{x}}$.

3.2 D^{2}GAN

Following the literature, we formally define the Few-Shot Learning (FSL) problem as following: Given a distribution of tasks $P(\mathcal{T})$, a sample task $\mathcal{T}\sim P(\mathcal{T})$ (also called an episode) is a tuple $\mathcal{T} = (S_\mathcal{T}, Q_\mathcal{T})$, where the support set $S_\mathcal{T} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_{N\times K}, y_{N\times K})\}$ contains K labeled samples from each of the N classes. This is usually known as K-shot N-way classification.

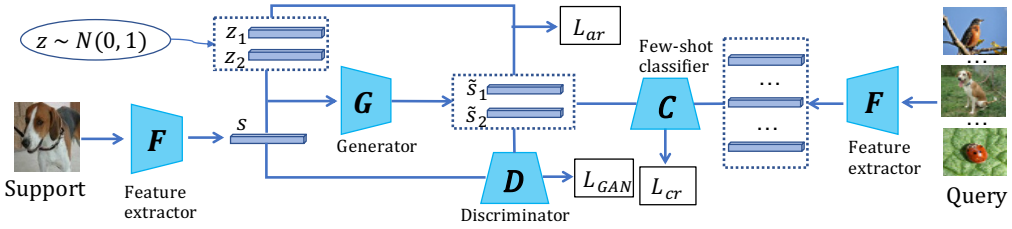


Figure 1: Framework of the proposed D^{2}GAN.

$Q_{\mathcal{T}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_T, y_T)\}$ is the query dataset. The objective is to minimize the classification prediction risk of $Q_{\mathcal{T}}$, according to $S_{\mathcal{T}}$.

Few-shot learning with conditional GAN. We approach FSL from the perspective of data augmentation via conditional GAN. The training pipeline is illustrated in Figure 1. The feature extraction network F takes as input each image $(\mathbf{x}, y) \in S_{\mathcal{T}}$, producing a feature representation vector

$$\mathbf{s} = F(\mathbf{x}). \tag{2}$$

When there are multiple samples for class y, i.e., $K > 1$, we simply average the feature vectors and take the averaged vector as the prototype of class y (Snell et al. (2017)). Conditioned on $\mathbf{s}$, we synthesize fake features for the class.

Unlike previous GAN models which sample a single random noise variable from some distribution, we sample two noise variables $\mathbf{z}_1$ and $\mathbf{z}_2 \sim N(0, 1)$. The generator G synthesizes fake feature $\tilde{\mathbf{s}}_1$ ($\tilde{\mathbf{s}}_2$) taking as input $\mathbf{z}_1$ ($\mathbf{z}_2$) and the class prototype $\mathbf{s}$,

$$\tilde{\mathbf{s}}_i = G(\mathbf{s}, \mathbf{z}_i), \quad \forall i = 1, 2. \tag{3}$$

Generator G aims to synthesize $\tilde{\mathbf{s}}_i$ to be as similar as possible to $\mathbf{s}$. The discriminator D, taking $\mathbf{z}_i$ and $\mathbf{s}$ as input, tries to discern $\tilde{\mathbf{s}}_i$ as fake. Within the WGAN framework, the loss function is as follows,

$$L_{GAN_i} = \mathbb{E}[D(\tilde{\mathbf{s}}_i, \mathbf{z}_i)] - \mathbb{E}[D(\mathbf{s}, \mathbf{z}_i)] + \lambda \mathbb{E}[(\|\nabla_{\hat{\mathbf{s}}_i} D(\hat{\mathbf{s}}_i)\|_2 - 1)^2], \quad \forall i = 1, 2. \tag{4}$$

Simply training the model with the above GAN loss does not guarantee the generated features are well suited for learning a discriminative classifier because it neglects the inter-class competing information among different classes. Moreover, since the conditioned feature vectors are of high dimension and structured, it is likely that the generator will neglect the noise vector and all synthesized features collapse to a single point in the feature space. To avoid these problems, we append the objective function with a classification regularization term and an anti-collapse regularization term, aiming to mitigate the mode collapse problem and get diverse and discriminative features.

Classification regularization. As our objective is to classify well samples in the query set $Q_{\mathcal{T}}$, given the support set $S_{\mathcal{T}}$. To encourage the synthesized features to serve well this purpose as the real features, inspired by (Snell et al. (2017)), we define a non-parametric FSL classifier, which calculates the possibility of a query image $\mathbf{q} = F(\mathbf{x}_q)$ of being the same class as synthesized feature $\tilde{\mathbf{s}}_i$ as

$$P(y|\mathbf{x}_q) = \frac{\exp(\cos(\tilde{\mathbf{s}}_i, \mathbf{q}))}{\sum_{j=1}^{N} \exp(\cos(\tilde{\mathbf{s}}_i^j, F(\mathbf{q})))}, \tag{5}$$

where $\cos(\mathbf{a}, \mathbf{b})$ is the Cosine similarity of two vectors, implemented as $\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}$. The adoption of Cosine similarity, rather than the widely-used dot product, is inspired by a recent FSL algorithm (Gidaris & Komodakis (2018)), which proved that replacing dot product with Cosine can bound and reduce the variance of the neurons and thus result in models of better generalization.

With the proposed FSL classifier, the classification regularization for in a typical FSL task is defined as follows:

$$L_{cr_i} = \mathop{\mathbb{E}}_{(\mathbf{x}_q, y_q) \sim Q_{\mathcal{T}}} \log[-P(y|\mathbf{x}_q)] \quad \forall i = 1, 2. \tag{6}$$

We can see that this regularizer explicitly encourages the synthesized features to have high correlation with features of the same classes, while low correlation with features of different classes. This thus ensures feature discriminability.

Anti-collapse regularization. GAN models are known for suffering from the notorious mode collapse problem, especially for conditional GANs, where the conditional contexts are generally high-dimensional and structured (e.g., images or sentences) as opposed to the noise vectors. As such, the generators are likely to focus on the contexts and ignore the noise vectors (latent codes), which account for diversity. Specifically to our case, our goal is to augment the few labeled samples in the feature space; when model collapse occurs, all synthesized features may collapse to a single point in the feature space, failing to diversify the training samples. Observing that noise vectors that are closer in the latent code space are more likely to be collapsed into the same mode when mapped to feature space, we directly penalize the ratio of the logarithmic similarity two synthesized feature vectors and the logarithmic similarity of the two noise vectors generating them.

Specifically, remember that we sample two random variables $\mathbf{z}_1$ and $\mathbf{z}_2$ and generate two fake feature vectors, i.e., $\tilde{\mathbf{s}}_1 = G(\mathbf{s}, \mathbf{z}_1)$ and $\tilde{\mathbf{s}}_2 = G(\mathbf{s}, \mathbf{z}_2)$. When $\mathbf{z}_1$ and $\mathbf{z}_2$ are closer, $\mathbf{s}_1$ and $\mathbf{s}_2$ are more likely to be collapsed into the same mode. To mitigate this, we define the anti-collapse regularization term

$$\mathcal{L}_{ar} = \frac{\log(\cos(\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2))}{\log(\cos(\mathbf{z}_1, \mathbf{z}_2))}. \tag{7}$$

We can observe this term is to penalize the similarity between the two fake feature vectors when the latent codes generating them are of high similarity. In this way, it encourages the generator to explore the feature space and enhances the chances for generating samples of minor modes. On the other hand, the discriminator is forced to pay attention to generate samples from minor modes. Therefore, the regularization term offers a virtuous circle for training our proposed D^{2}GAN model.

With the above two regularization terms, we reach our final training objective as

$$\min_G \max_D \sum_{i=1}^{2} L_{GAN_i} + \alpha \sum_{i=1}^{2} L_{cr_i} + \beta L_{ar}, \tag{8}$$

where α and β are two hyper-parameters.

3.3 Classification with Synthesized Samples

In the test stage, given a FSL task $\mathcal{T} = (S_\mathcal{T}, Q_\mathcal{T})$, we first augment the labeled support set $S_\mathcal{T}$ with the learned generator G. Then, we train a classifier with the augmented supported set. The classifier is used to classify samples from the query set $Q_\mathcal{T}$. Specifically, suppose after data augmentation, we get an enlarged support set $\hat{S}_\mathcal{R} = \{(\mathbf{s}_1, y_1), (\mathbf{s}_2, y_2), \cdots, (\mathbf{s}_{N\times K'}, y_{N\times K'})\}$ where K' is the number of samples synthesized for each class. With $\hat{S}_\mathcal{R}$, we train a standard Softmax classifier $\mathbf{f}_c$ as

$$\min_\theta \mathop{\mathbb{E}}_{(\mathbf{s},y)\sim\hat{S}_\mathcal{R}} \log[-P(y|\mathbf{s};\theta)], \tag{9}$$

where $\theta \in \mathbb{R}^{d_s \times N}$ is the parameter of $\mathbf{f}_c$. With $\mathbf{f}_c$, we classify samples from $Q_\mathcal{T}$.

4 Experiments

We evaluate D^{2}GAN on three common benchmark datasets, namely, *Mini-ImageNet* (Vinyals et al. (2016)), *CUB* (Wah et al. (2011)) and CIFAR100 (Krizhevsky & Hinton (2009)). The *Mini-ImageNet* dataset is a subset of ImageNet. It has 60,000 images from 100 classes, 600 images for each class. We follow previous methods and use the splits in (Ravi & Larochelle (2017)) for evaluation, i.e., 64, 16, 20 classes as training, validation, and testing sets, respectively. The *CUB* dataset is a fine-grained dataset of totally 11,788 images from 200 categories of birds. We use the split in (Ravi & Larochelle (2017)) and 100, 50, 50 classes for training, validation, and testing, respectively. The CIFAR-100 dataset contains 60,000 images from 100 categories. We use the same data split as in (Zhou et al. (2018)). In particular, 64, 16 and 20 classes are used for training, validation and testing, respectively. For all the three datasets, we resize images to 224×224.

Following previous methods, we evaluate 5-way 1-shot and 5-way 5-shot classification tasks where each task instance involves classifying test images from 5 sampled classes with 1 or 5 randomly sampled images for each class as the support set. In order to reduce variance, we repeat the evaluation task 600 times and report the mean of the accuracy with a 95% confidence interval.

cGAN	✗	✓	✓	✓
CR	✗	✗	✓	✓
AR	✗	✗	✗	✓
	$52.73^{\star}$	57.58	60.57	62.26

Table 1: Ablation study on the *Mini-ImageNet* dataset for 5-way 1-shot setting. "cGAN" refers to conditional GAN, "CR" indicates the classification regularizer, and "AR" represents the anti-collapse regularizer. $\star$ indicates the result is obtained by the baseline method, ResNet18+SVM, which applies the SVM classifier directly on ResNet18 features without data augmentation.

We compare with various FSL algorithms, including (1) metric learning based methods: Matching Net (Vinyals et al. (2016)), PROTO NET (Snell et al. (2017)), MM-Net (Cai et al. (2018)), MACO (Hilliard et al. (2018)), GNN (Garcia & Bruna (2017)), RELATION NET (Sung et al. (2018)), and TPN (Liu et al. (2019)); (2) meta-learning based methods: MAML (Finn et al. (2017)), META-LSTM (Ravi & Larochelle (2017)), META-SGD (Li et al. (2017)), SNAIL (Mishra et al. (2017)), DFSVL (Gidaris & Komodakis (2018)), PPA (Qiao et al. (2018)), and LEO (Rusu et al. (2019)); (3) data augmentation based methods: MetaGAN (Zhang et al. (2018)), Dual TriNet (Chen et al. (2019b)), Δ-encoder (Schwartz et al. (2018)), and IDeMe-Net (Chen et al. (2019a)).

4.1 Implementation Details

Following the peer data augmentation based methods (Schwartz et al. (2018); Chen et al. (2019b;a)), we use ResNet18 (He et al. (2016)) as our feature extraction network F. We implement the generator G as a two-layer MLP, with LeakyReLU activation for the first layer and ReLU activation for the second one. The dimension of the hidden layer is 1024. The discriminator is also a two-layer MLP, with LeakyReLU as the activation function for the first layer and Sigmoid for the second layer. The dimension of the hidden layer is also 1024. The noise vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ are drawn from a unit Gaussian with the same dimensionality as the feature embeddings.

Following the peer methods, we perform two step training procedures. In the first step, we only train the feature extraction network F as a multi-class classification task using only the training split. We use Adam optimizer with an initial learning rate 10^{-3} which decays to the half every 10 epochs. We train F with 100 epochs with batch size of 128. In the second training stage, we train the generator and discriminator alternatively, using features extracted by F and update G once every 5 updates of D. The Adam optimizer has an initial learning rate of 10^{-5} for both models and decays to half every 20 epochs. We train the whole network with 100 epochs where there are 600 randomly sampled FSL tasks in each epoch. For the hyper-parameters, we set $\lambda = 10$ as suggested by (Gulrajani et al. (2017)), and $\alpha = \beta = 1$ for all the three datasets. During the test stage, we synthesize 300 fake features for each class.

The code is developed based on PyTorch.

4.2 Ablation Study

Based on conditional GAN (cGAN), the proposed D^2GAN model synthesizes fake features conditioned on those of labeled samples. Besides, D^2GAN incorporates the novel Classification Regularier (CR) and Anti-collapse Regularizer (AR) to ensure discriminability and diversity of the synthesized features. To evaluate the effectiveness and impact of these components, we conduct ablation study on the *Mini-ImageNet* dataset in the 5-way 1-shot setting. The results are shown in Table 1.

cGAN. Compared with the baseline result (without data augmentation), we can observe that it raises the accuracy from 52.73 to 57.58 by augmenting the labeled samples using our cGAN model. We analyze the reason is that the synthesized features enhance intra-class variance, which facilitates to make the decision boundaries much sharper.

CR. This regularizer is to regularize the synthesized features to have desirable classification property such that we can train from them a discriminative classifier. We can observe that the accuracy is boosted by 3% with this regularization term added, thereby substantiating its effectiveness.

AR. The regularizer aims to enhance the diversity of the synthesized features and Table 1 shows that it further brings about 2% performance gains.

	Backbone	1-shot	5-shot
ResNet18 + SVM (baseline)	ResNet18	52.73	73.31
Matching Net (Vinyals et al. (2016))	4Conv	43.56	55.31
PROTO Net (Snell et al. (2017))	4Conv	49.42	68.20
MM-Net (Cai et al. (2018))	4Conv	53.37	66.97
GNN (Garcia & Bruna (2017))	4Conv	50.33	66.41
RELATION NET (Sung et al. (2018))	4Conv	57.02	71.07
TPN (Liu et al. (2019))	4Conv	55.51	69.86
MAML (Finn et al. (2017))	4Conv	48.70	63.11
META-LSTM (Ravi & Larochelle (2017))	4Conv	43.44	60.60
SNAIL (Mishra et al. (2017))	4Conv	55.71	68.88
MACO (Hilliard et al. (2018))	4Conv	41.09	58.32
DFSVL (Gidaris & Komodakis (2018))	4Conv	55.95	73.00
META-SGD (Li et al. (2017))	4Conv	$50.47^{\diamond}$	$64.03^{\diamond}$
PPA (Qiao et al. (2018))	WRN-28-10	59.60	73.74
LEO (Rusu et al. (2019))	WRN-28-10	61.76	77.59
MetaGAN (Zhang et al. (2018))	4Conv	52.71	68.63
Dual TriNet (Chen et al. (2019b))	ResNet18	$58.80^{\dagger}$	$76.71^{\dagger}$
Δ-encoder (Schwartz et al. (2018))	ResNet18	59.90	69.70
IDeMe-Net (Chen et al. (2019a))	ResNet18	59.14	74.63
D^2GAN	ResNet18	**62.26**	**78.07**

Table 2: Few-shot classification accuracy on *Mini-Imagenet*. $\diamond$: using a large external dataset. $\dagger$: using label embedding trained on large corpus or human annotated class attributes. We omit the standard deviations, which are less than 2% for all methods. The best results are in **bold**.

4.3 Comparative Results

Mini-Imagenet. This dataset is the most extensively evaluated dataset. From Table 2 we can observe that D^2GAN attains the new state-of-the-art, for both the 1-shot and 5-shot setting. Compared with the other four data augmentation based methods, D^2GAN reaches significant improvements: it beats Δ-encoder (Schwartz et al. (2018)) by more than 8% for the 5-shot setting and Dual TriNet (Chen et al. (2019b)) by more than 3% for the 1-shot setting. Compared with MetaGAN (Zhang et al. (2018)) which is also based on GAN, D^2GAN achieves about 10% improvements for both the 1-shot and 5-shot cases. Besides the remarkable advantages over the peer data augmentation based methods, D^2GAN also exhibits remarkable advantages over the other two categories of methods. It beats the best metric learning based method RELATION NET (Sung et al. (2018)) by about 5% and 7% for the 1-shot and 5-shot setting, respectively. It also performs better than the best meta-learning based algorithm. Compared with the baseline method, "ResNet18+SVM", D^2GAN reaches about 10% and 5% improvements for the 1-shot and 5-shot setting, respectively. This substantiates the effectiveness of our proposed data augmentation techniques.

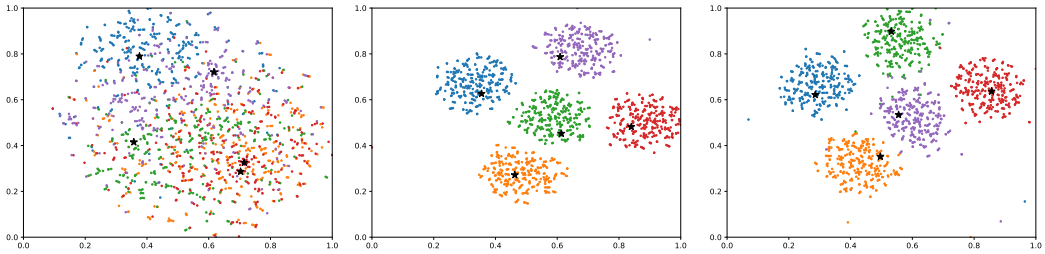
CUB & CIFAR100. From Table 3 we can see that D^2GAN reaches comparable results with the state-of-the-art on *CUB*, while attains the new state-of-the-art on *CIFAR100*. Specially on *CIFAR100*, D^2GAN beats Dual TriNet by 5% and 3% for 1-shot and 5-shot respectively. Compared with the best meta-learning based method, we get 7% and 4% improvements for the 1-shot and 5-shot respectively. One thing worthy to be noted is that compared with our baseline method, we have a moderate improvement in the 1-shot setting while reach only a marginal boost for the 5-shot setting for the *CUB* dataset. We speculate the reason is that this dataset is relatively small, less than 60 images per class on average; a large number of classes only have about only 30 images. Due to the small scale of this dataset, the intra-class variance is less significant than that of the other datasets, such that 5 labeled samples are already sufficient to capture most of the intra-class variance; performing data augmentation becomes less crucial than that for the other datasets.

4.4 Further Analysis

Impact of the number of synthesized features. Figure 3 shows the analysis on *Mini-ImageNet* about the recognition accuracy with respect to the number of synthesized features during test stage. We can observe that the classification accuracy keeps boosted with more features synthesized at the beginning, and remains stable with even more synthesized samples. This is reasonable because the class variance encapsulated by the few labeled samples has a limit; data augmentation based on these

	Backbone	*CUB*		*CIFAR100*	
		1-shot	5-shot	1-shot	5-shot
ResNet18 + SVM (baseline)	ResNet18	66.54	82.38	59.65	76.75
Matching Net (Vinyals et al. (2016))	4Conv	49.34	59.31	50.53	60.30
PROTO Net (Snell et al. (2017))	4Conv	45.27	56.35	-	-
MAML (Finn et al. (2017))	4Conv	38.43	59.15	49.28	58.30
META-LSTM (Ravi & Larochelle (2017))	4Conv	40.43	49.65	-	-
MACO (Hilliard et al. (2018))	4Conv	60.76	74.96	-	-
META-SGD (Li et al. (2017))	4Conv	66.90$^\diamond$	77.10$^\diamond$	61.60$^\diamond$	77.90$^\diamond$
Dual TriNet (Chen et al. (2019b))	ResNet18	69.61†	**84.10**†	63.41†	78.43†
Δ-encoder (Schwartz et al. (2018))	ResNet18	69.80	82.60	66.70	79.80
D^{2}GAN	ResNet18	**70.49**	83.92	**68.23**	**81.38**

Table 3: Few-shot classification accuracy on *CUB* and *CIFAR100*. Please refer Table 2 for details.



cGAN cGAN + CR cGAN + CR + AR

Figure 2: t-SNE (Maaten & Hinton (2008)) visualization of synthesized feature embeddings. From the left to right are the results of cGAN, cGAN with CR regularizer, and cGAN with both CR and AR regularizers. The real features are indicated by $\star$. Different colors represent different classes.

labeled samples can enlarge the variance to some extent, but it is still bounded by the few labeled samples themselves. When it reaches the peak, the performance reasonably turns stable.

Visualization of synthesized features. We showed quantitatively in the ablation study that owing to the CR and AR regularizers, we can generate diverse and discriminative features, which brings significant performance gains. Here we further study the effect of the two regularizers by showing the t-SNE visualization of the synthesized features. As shown in Figure 2, the synthesized features of different classes mix up together when using only cGAN for augmentation. As analyzed before, cGAN does not guarantee synthesizing semantically meaningful features. The problem is substantially resolved when we train cGAN with CR. The synthesized features exhibit clear clustering structure, which helps train a discriminative classifier. Further, with AR added, the synthesized features still exhibit favorable clustering structure. Taking a closer look of the visualization, we can find the features synthesized with AR added are more diverse than that without it: the clusterings are less compact, stretched to larger regions, and even contains some noise. This shows AR indeed helps diversify the synthesized features.

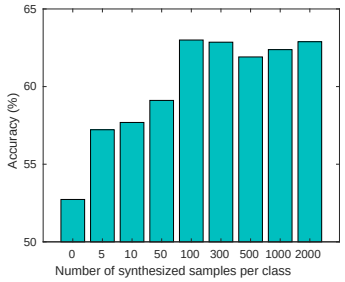


Figure 3: Impact of the number of synthesized samples on the *Mini-ImageNet* dataset.

5 Conclusions

We introduced in this paper D^{2}GAN, a new GAN model for few-shot learning. D^{2}GAN synthesizes feature embeddings for new classes conditioned on those of a few labeled samples of the classes. Two novel regularizers are proposed to train D^{2}GAN. The first one secures feature discriminability by requiring the synthesized features to be similar with other samples of the same classes. The second regularizer aims for enhancing the diversity of the synthesized features, by penalizing the ratio of the logarithmic similarity between two synthesized features and the logarithmic similarity of the two latent codes generating them. Experiments on three common benchmark datasets substantiate the superiority of D^{2}GAN to the state-of-the-art, and the two proposed regularizers indeed enhance feature discriminability and diversity.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

Qi Cai, Yingwei Pan, Ting Yao, Chenggang Yan, and Tao Mei. Memory matching networks for one-shot image recognition. In *CVPR*, 2018.

Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019a.

Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. *IEEE Transactions on Image Processing*, 2019b.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *NeuIPS*, 2018.

Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeuIPS*, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeuIPS*, 2017.

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *CVPR*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D Corley, and Nathan O Hodas. Few-shot learning with metric-agnostic conditional embeddings. *arXiv preprint arXiv:1802.04376*, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, 2017.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR, abs/1803.02999*, 2018.

Siyuan Qiao, Chenxi Liu, Wei Shen, and Alan L Yuille. Few-shot image recognition by predicting parameters from activations. In *CVPR*, 2018.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019.

Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *arXiv preprint arXiv:1806.04734*, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeuIPS*, 2017.

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 *CVPR*, 2018.

Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *NeuIPS*, 2017.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeuIPS*, 2016.

Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011.

Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018.

Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

Aron Yu and Grauman Kristen. Low-shot learning via covariance-preserving adversarial augmentation networks. In *ICCV*, 2017.

Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *NeuIPS*, 2018.

Fengwei Zhou, Bin Wu, and Zhenguo Li. Deep meta-learning: Learning to learn in the concept space. *arXiv preprint arXiv:1802.03596*, 2018.

APPENDIX A: NETWORK ARCHITECTURE

Feature extraction model: Standard ResNet18 (excluding the last FC layer).

Generator: FC (1024×1024)→ LeakyReLU → FC (1024×1024)→ ReLU.

Discriminator: FC (1024×1024)→ LeakyReLU → FC (1024×1024)→ Sigmoid.

APPENDIX B: PSEUDO CODE

Algorithm 1 outlines the main steps of training the proposed model.

Algorithm 1. Proposed FSL algorithm

Input: Training set $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{Y}_t\}$, hyper-paramters λ, α, and β.
Output: Feature extraction network F, parameterized by θ_f; generator G, parameterized by θ_g; discriminator D, paremterized by θ_d.

1. Train F as a standard multi-class classification task using $\mathcal{D}_t$
while not done **do**
  2. Randomly sample from $\mathcal{D}_t$ a batch of FSL tasks $\mathcal{T}_i^d \sim p(\mathcal{D}_t)$
  For each $\mathcal{T}_i^d$ **do**
    3. Sample a support set $\mathcal{S} = \{\{\mathbf{x}_{i,j}\}_{i=1}^N, y_j\}_{j=1}^M$ consisting of N samples for each of M classes.
    4. Sample query set $\mathcal{Q} = \{\{\mathbf{x}_{k,j}\}_{k=1}^Q, y_j\}_{j=1}^M$ consisting of Q samples for each of the M classes.
    5. Calculate prototype set of the M classes $\mathcal{P} = \{s_j\}_{j=1}^M$, where $\mathbf{s}_j = \frac{1}{N}\sum_{i=1}^N F(\mathbf{x}_{i,j})$.
    6. Sample M random variables $\mathcal{Z}_1 = \{\mathbf{z}_1^j\}_{j=1}^M$ and variables $\mathcal{Z}_2 = \{\mathbf{z}_2^j\}_{j=1}^M$
      where each $z_1^j \sim N(0,1)$ and each $z_2^j \sim N(0,1)$.
    7. Generate fake feature sets $\tilde{\mathcal{Z}}_1 = \{\tilde{\mathbf{z}}_1^j\}_{j=1}^M$ and $\tilde{\mathcal{Z}}_2 = \{\tilde{\mathbf{z}}_2^j\}_{j=1}^M$,
      where $\tilde{\mathbf{z}}_1^j = G(\mathbf{s}_j, \mathbf{z}_1^j)$ and $\tilde{\mathbf{z}}_2^j = G(\mathbf{s}_j, \mathbf{z}_2^j)$
    8. Calculate discriminator loss l_D according to Eq. (8) in the main text
    9. Update θ_d using l_D
  end For
  10. Randomly sample from $\mathcal{D}_t$ a batch of FSL tasks $\mathcal{T}_i^g \sim p(\mathcal{D}_t)$
  For each $\mathcal{T}_i^g$ **do**
    11. Sample a support set $\mathcal{S} = \{\{\mathbf{x}_{i,j}\}_{i=1}^N, y_j\}_{j=1}^M$ consisting of N samples for each of M classes.
    12. Sample query set $\mathcal{Q} = \{\{\mathbf{x}_{k,j}\}_{k=1}^Q, y_j\}_{j=1}^M$ consisting of Q samples for each of the M classes.
    13. Calculate prototype set of the M classes $\mathcal{P} = \{s_j\}_{j=1}^M$, where $\mathbf{s}_j = \frac{1}{N}\sum_{i=1}^N F(\mathbf{x}_{i,j})$.
    14. Sample M random variables $\mathcal{Z}_1 = \{\mathbf{z}_1^j\}_{j=1}^M$ and variables $\mathcal{Z}_2 = \{\mathbf{z}_2^j\}_{j=1}^M$
      where each $z_1^j \sim N(0,1)$ and each $z_2^j \sim N(0,1)$.
    15. Generate fake feature sets $\tilde{\mathcal{Z}}_1 = \{\tilde{\mathbf{z}}_1^j\}_{j=1}^M$ and $\tilde{\mathcal{Z}}_2 = \{\tilde{\mathbf{z}}_2^j\}_{j=1}^M$,
      where $\tilde{\mathbf{z}}_1^j = G(\mathbf{s}_j, \mathbf{z}_1^j)$ and $\tilde{\mathbf{z}}_2^j = G(\mathbf{s}_j, \mathbf{z}_2^j)$
    16. Calculate discriminator loss l_G according to Eq. (8) in the main text
    17. Update θ_g using l_G
  end For
end while

APPENDIX C: DETAILED EXPERIMENTAL RESULTS

In the main text, we omit the 95% confidence intervals over tasks to save space. We add the interval values and show them in Tables 4 - 6.

	1-shot	5-shot
ResNet18 + SVM (baseline)	52.73±1.44	73.31±0.81
Matching Net (Vinyals et al. (2016))	43.56±0.84	55.31±0.73
PROTO Net (Snell et al. (2017))	49.42±0.78	68.20±0.66
MM-Net (Cai et al. (2018))	53.37±0.48	66.97±0.35
GNN (Garcia & Bruna (2017))	50.33±0.36	66.41±0.63
RELATION NET (Sung et al. (2018))	57.02±0.92	71.07±0.69
TPN (Liu et al. (2019))	55.51±0.86	69.86±0.65
MAML (Finn et al. (2017))	48.70±1.84	63.11±0.92
META-LSTM (Ravi & Larochelle (2017))	43.44±0.77	60.60±0.71
SNAIL (Mishra et al. (2017))	55.71±0.99	68.88±0.92
MACO (Hilliard et al. (2018))	41.09±0.32	58.32±0.21
DFSVL (Gidaris & Komodakis (2018))	55.95±0.89	73.00±0.68
META-SGD (Li et al. (2017))	50.47±1.87	64.03±0.94
Qiao et al. (2018)	59.60±0.41	73.74±0.19
LEO (Rusu et al. (2019))	61.76±0.08	77.59±0.12
MetaGAN (Zhang et al. (2018))	52.71±0.64	68.63±0.67
Dual TriNet (Chen et al. (2019a))	58.80±1.37	76.71±0.69
Δ-encoder (Schwartz et al. (2018))	59.90	69.70
IDeMe-Net (Chen et al. (2019a))	59.14±0.86	74.63±0.74
D^2GAN	**62.26±0.83**	**78.07±0.65**

Table 4: Few-shot classification accuracy on *Mini-Imagenet* dataset. The ± indicates 95% confidence intervals over tasks. The best results are in **bold**.

	1-shot	5-shot
ResNet18 + SVM (baseline)	66.54±0.53	82.38±0.43
Matching Net (Vinyals et al. (2016))	49.34	59.31
PROTO Net (Snell et al. (2017))	45.27	56.35
MAML (Finn et al. (2017))	38.43	59.15
META-LSTM (Ravi & Larochelle (2017))	40.43	49.65
MACO (Hilliard et al. (2018))	60.76	74.96
META-SGD (Li et al. (2017))	66.90	77.10
Dual TriNet (Chen et al. (2019a))	69.61	**84.10**
Δ-encoder (Schwartz et al. (2018))	69.80±0.46	82.60±0.35
D^2GAN	**70.49**±1.02	83.92±0.71

Table 5: Few-shot classification accuracy on *CUB*. The ± indicates 95% confidence intervals over tasks. The best results are in **bold**.

	1-shot	5-shot
ResNet18 + SVM (baseline)	59.65±0.78	76.75±0.73
Matching Net (Vinyals et al. (2016))	50.53±0.87	60.30±0.82
MAML (Finn et al. (2017))	49.28±0.90	58.30±0.80
META-SGD (Li et al. (2017))	61.60±0.89	77.90±0.74
Dual TriNet (Chen et al. (2019a))	63.41±0.64	78.43±0.62
Δ-encoder (Schwartz et al. (2018))	66.70	79.80
D^2GAN	**68.23±1.04**	**81.38±0.78**

Table 6: Few-shot classification accuracy on *CIFAR100* datasets. The ± indicates 95% confidence intervals over tasks. The best results are in **bold**.