MetaGAN: An Adversarial Approach to Few-Shot Learning

Ruixiang Zhang[*,†]
MILA, Université de Montréal
sodabeta7@gmail.com

Tong Che[*]
MILA, Université de Montréal
tongcheprivate@gmail.com

Zoubin Grahahramani
University of Cambridge
zoubin@cam.ac.uk

Yoshua Bengio
MILA, Université de Montréal
yoshua.bengio@mila.quebec

Yangqiu Song
Hong Kong University of Science and Technology
yqsong@cse.ust.hk

Abstract

In this paper, we propose a conceptually simple and general framework called MetaGAN for few-shot learning problems. Most state-of-the-art few-shot classification models can be integrated with MetaGAN in a principled and straightforward way. By introducing an adversarial generator conditioned on tasks, we augment vanilla few-shot classification models with the ability to discriminate between real and fake data. We argue that this GAN-based approach can help few-shot classifiers to learn sharper decision boundary, which could generalize better. We show that with our MetaGAN framework, we can extend supervised few-shot learning models to naturally cope with unlabeled data. Different from previous work in semi-supervised few-shot learning, our algorithms can deal with semi-supervision at both sample-level and task-level. We give theoretical justifications of the strength of MetaGAN, and validate the effectiveness of MetaGAN on challenging few-shot image classification benchmarks.

1 INTRODUCTION

Deep neural networks have achieved great success in many artificial intelligence tasks. However, they tend to struggle when data is scarce or when they need to adapt to new tasks within a few numbers of steps. On the other hand, humans are able to learn new concepts quickly, given just a few examples. The reason for this performance gap between human and artificial learners is usually explained as that humans can effectively utilize prior experiences and knowledge when learning a new task, while artificial learners usually seriously overfit without the necessary prior knowledge.

Meta-learning [Thrun, 1998, Hochreiter et al., 2001] addresses this problem by training a particular adaptation strategy to a distribution of similar tasks, trying to extract transferable patterns useful for many tasks. Recently, many different meta-learning or few-shot learning algorithms have been proposed. These algorithms may take the forms of learning a shared metric [Sung et al., 2018, Snell et al., 2017], a shared initialization of network parameters [Finn et al., 2017], shared optimization algorithms [Ravi and Larochelle, 2017, Munkhdalai et al., 2017, Munkhdalai and Yu, 2017], or generic inference networks [Santoro et al., 2016, Mishra et al., 2018] . In the context of few-shot classification, these algorithms try to learn a good strategy to form a correct decision boundary between different classes from only a few samples of data in each class.

[*]Equal contribution.
[†]Work done at HKUST

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

In this work we present MetaGAN as a general and flexible framework for few-shot learning. Most state-of-the-art few-shot learning models can be integrated into MetaGAN seamlessly. While most few-shot learning models consider how to effectively utilize few labeled data in a supervised learning way, semi-supervised few-shot learning which is studied recently in [Ren et al., 2018] is proposed when unlabeled data are available. In this paper, we show that both supervised few-shot learning and semi-supervised few-shot learning can be unified naturally with our prpoposed MetaGAN framework. We can further extend the sample-level semi-supervised learning proposed in [Ren et al., 2018] to the task level. For sample-level semi-supervised few-shot learning, we allow some training samples to be unlabeled within a task. These training samples can either come from the same classes as the labeled samples, or come from different "distractor" classes. For task-level semi-supervised few-shot learning, we also allow purely unsupervised tasks, in which both support and query samples are all unlabeled. Task-level semi-supervised few-shot learning can be very natural in practice. For example, we can have robots with cameras collecting data in different places. It is safe to assume that the data collected by one robot in a short time range come from a specific distribution, so classifying these images can be viewed as one task. But these tasks are completely unlabeled, both in the support and in the query sets. The MetaGAN algorithm is able to learn to infer the shape and boundaries of data manifolds of the task-specific data distribution from both labeled and unlabeled examples.

We provide both intuitive and formal theoretical justifications on the key idea behind MetaGAN. The main difficulty in few-shot learning is how to form generalizable decision boundaries from a small number of training samples. We argue that adversarial training can help few-shot learning models by making it easier to learn better decision boundaries between different classes. Although training data is usually very limited for each task, we show that how fake data generated by a non-perfect generator in MetaGAN can help the classifier identify much tighter decision boundaries (real-fake decision boundaries) and thus can help boost the performance of few-shot learning.

We demonstrate the effectiveness of MetaGAN on popular few-shot image classification benchmarks in both supervised and semi-supervised settings. We choose two representative few-shot learning models, MAML[Finn et al., 2017] representing models that learn to adapt using gradients, and Relation Network[Sung et al., 2018] representing models that learn distance metrics, and combine them with MetaGAN. [3] We show that MetaGAN can consistently improve the performance of popular few-shot classifiers in all of these scenarios.

2 BACKGROUND

2.1 FEW-SHOT LEARNING

We formally define few-shot learning problems as following: Given a distribution of tasks $P(\mathcal{T})$, a sample task $\mathcal{T}$ from $P(\mathcal{T})$ is given by a joint distribution $P^{\mathcal{T}}_{X\times Y}(\mathbf{x}, y)$, where the task is to predict y given $\mathbf{x}$. We have a set of training sample tasks $\{\mathcal{T}_i\}_{i=1}^{N}$. Each training sample task $\mathcal{T}$ is a tuple $\mathcal{T} = (S_{\mathcal{T}}, Q_{\mathcal{T}})$, where the support set is denoted as $S_{\mathcal{T}} = S^s_{\mathcal{T}} \cup S^u_{\mathcal{T}}$, and the query set is denoted as $Q_{\mathcal{T}} = Q^s_{\mathcal{T}} \cup Q^u_{\mathcal{T}}$. The supervised support set $S^s_{\mathcal{T}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_{N\times K}, y_{N\times K})\}$ contains K labeled samples from each of the N classes (this is usually known as K-shot N-way classification). The optional unlabeled support set $S^u_{\mathcal{T}} = \{(\mathbf{x}^u_1, \mathbf{x}^u_2, \cdots \mathbf{x}^u_M\}$ contains unlabeled samples from the same set of N classes, which can also be empty in purely supervised cases. $Q^s_{\mathcal{T}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots (\mathbf{x}_T, y_T)\}$ is the supervised query dataset. $Q^u_{\mathcal{T}} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_P\}$ is the optional unlabeled query dataset. The objective of the model is to minimize the loss of its predictions on a query set, given the support set as input.

2.2 ADVERSARIAL TRAINING

The generative adversarial networks [Goodfellow et al., 2014] framework is one of the most popular approaches to generative modeling. It tries to adversarially train two neural networks, a generator and a discriminator. Adversarial training has seen a vast range of applications in recent years, such as semi-supervised learning [Dai et al., 2017, Salimans et al., 2016], unsupervised representation learning [Chen et al., 2016], imitation learning [Ho and Ermon, 2016] etc. However, few works have successfully combined adversarial training with few-shot learning. [Antoniou et al., 2018] proposed

[3] However, it is worth noticing that MetaGAN can also be easily combined with other models, such as prototypical networks or SNAIL.

to train a class conditioned GAN (DAGAN) to perform data augmentation. This is related to our proposal but is different in two aspects. 1) Their GAN model is trained separately from the classifier, only to provide additional data. 2) They treat generated data as real training data of the conditioned class. There are two drawbacks of this approach. First, GANs still have trouble in generating realistic samples in complex datasets such as ImageNet, so treating the generated images as real data in these datasets is questionable. Second, DAGAN can very easily run into mode collapsing. In many cases it is easy to collapse to an identity function — it just reconstruct the input image. Our approach does not require the generator to be perfect. Conversely, similar to the semi-supervised learning case [Dai et al., 2017], it can even benefit from an imperfect generator.

3 OUR APPROACH

MetaGAN is a conceptually simple and general framework for few-shot learning problems. Given a decent K-shot N-way classifier, similar to [Salimans et al., 2016] we introduce a conditional generative model with the objective to generate samples which are not distinguishable from true data sampled from a specific task. We increase the dimension of the classifier output from N to $N+1$, to model the probability that input data is fake. We train the discriminator (classifier) and generator in an adversarial setup.

The key idea behind MetaGAN is that imperfect generators in GAN models can provide fake data between the manifolds of different real data classes, thus providing additional training signals to the classifier as well as making the decision boundaries much sharper. We first describe our basic model formally in section 3.1, then introduce details of different instances of MetaGAN in following sections.

3.1 BASIC ALGORITHM

We first introduce the basic formulation of MetaGAN here. For a few-shot N-way classification problem $P(\mathcal{T})$ and dataset $\{\mathcal{T}_i\}_{i=1}^M$, assume we have one of the state-of-the-art few-shot classifiers $p_D(\mathbf{x};\mathcal{T}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \cdots p_N(\mathbf{x}))$. Note that D is conditioned on a specific task $\mathcal{T}$. In practice, this conditioning can be either via fast adaptation [Finn et al., 2017] or feeding the support set as input [Snell et al., 2017, Mishra et al., 2018, Sung et al., 2018]. We augment the classifier with an additional output, as done in semi-supervised learning with GANs [Salimans et al., 2016]: $p_D(\mathbf{x};\mathcal{T}) = (p_1(\mathbf{x}), p_2(\mathbf{x}), \cdots p_N(\mathbf{x}), p_{N+1}(\mathbf{x}))$. We also train a task-conditioned generator $G(\mathbf{z},\mathcal{T})$ with generating distribution $p_G^{\mathcal{T}}(\mathbf{x})$ that tries to generate data for the specific task $\mathcal{T}$. Then for the training episode of task $\mathcal{T}$ we maximize the following combination of the N-way classification objective and the real/fake classification objective for the discriminator:

$$\mathcal{L}_D^{\mathcal{T}} = \mathcal{L}_{\text{supervised}} + \mathcal{L}_{\text{unsupervised}}, \tag{1}$$

$$\mathcal{L}_{\text{supervised}} = \mathbb{E}_{\mathbf{x},y\sim Q_{\mathcal{T}}^s} \log p_D(y|\mathbf{x}, y \le N) \tag{2}$$

$$\mathcal{L}_{\text{unsupervised}} = \mathbb{E}_{\mathbf{x}\sim Q_{\mathcal{T}}^u} \log p_D(y \le N|\mathbf{x}) + \mathbb{E}_{\mathbf{x}\sim p_G^{\mathcal{T}}} \log p_D(N+1|\mathbf{x}) \tag{3}$$

For the generator, we minimize the non-saturating generator loss

$$L_G^{\mathcal{T}}(D) = -\mathbb{E}_{\mathbf{x}\sim p_G^{\mathcal{T}}}[\log(p_D(N+1|\mathbf{x}))]. \tag{4}$$

Then the overall objective for training MetaGAN is

$$\mathcal{L}_D = \max_D \mathbb{E}_{\mathcal{T}\sim P(\mathcal{T})} \mathcal{L}_D^{\mathcal{T}} \tag{5}$$

$$\mathcal{L}_G = \min_G \mathbb{E}_{\mathcal{T}\sim P(\mathcal{T})} \mathcal{L}_G^{\mathcal{T}}. \tag{6}$$

3.2 DISCRIMINATOR

MetaGAN generally doesn't impose restrictions on the design of discriminator. It can be adapted from almost any state-of-the-art few-shot learners. We adopt two popular choices of few-shot classification models as our disciminator, MAML[Finn et al., 2017] and Relation Networks [Sung et al., 2018], representing learning to fast fine-tune based models and learning shared embedding and metric based models respectively.

3.2.1 METAGAN WITH MAML

MAML trains a transferable initialization that is able to quickly adapt to any specific task with one step gradient descent. Formally the discriminator $D(\theta_d)$ is parametrized by parameters θ_d. For a specific task $\mathcal{T} \sim P(\mathcal{T})$, we update the parameters to $\theta'_d = \theta_d - \alpha \nabla_{\theta_d} \ell_D^{\mathcal{T}}$ according to the loss eq. 7

$$\ell_D^{\mathcal{T}} = -\mathbb{E}_{\mathbf{x}, y \sim S_{\mathcal{T}}^s} p_D(y|\mathbf{x}, y \leq N) - \mathbb{E}_{\mathbf{x} \sim S_{\mathcal{T}}^u} \log p_D(y \leq N|\mathbf{x}) - \mathbb{E}_{\mathbf{x} \sim p_G^{\mathcal{T}}} \log p_D(N+1|\mathbf{x}). \quad (7)$$

Then we minimize the expected loss on query set with adapted discriminator $D(\theta'_d)$ across tasks $\mathcal{T}$ to train the discriminator's initial parameters θ_d, and we train the generator using adapted discriminator $D(\theta'_d)$. Finally our whole model combinging MetaGAN with MAML can be trained using the loss introduced in eq. 5 and eq. 6, as shown below:

$$\mathcal{L}_D = \max_D \mathbb{E}_{\mathcal{T} \sim P(\mathcal{T})} \mathcal{L}_{D(\theta'_d)}^{\mathcal{T}} \quad (8)$$

$$\mathcal{L}_G = \min_G \mathbb{E}_{\mathcal{T} \sim P(\mathcal{T})} \mathcal{L}_G^{\mathcal{L}}(D(\theta'_d)). \quad (9)$$

We put the detailed algorithms for training MetaGAN with MAML model in the supplemental material.

3.2.2 METAGAN WITH RELATION NETWORK

The Relation Network (RN) is a few-shot learning model aiming to do classification via learning a deep distance metric between images. MetaGAN can integrate with RN in a principled and straightforward way.

For a specific task $\mathcal{T} \sim P(\mathcal{T})$, following [Sung et al., 2018] let $r_{i,j} = g_\psi(\mathcal{C}(f_\phi(\mathbf{x}_i), f_\phi(\mathbf{x}_j))), \mathbf{x}_i \in S_{\mathcal{T}}^s, \mathbf{x}_j \in Q_{\mathcal{T}}^s$ be the relevance score between query set image $\mathbf{x}_j$ and support set image $\mathbf{x}_i$, where g_ψ is the relation module, f_ϕ is the feature embedding network and $\mathcal{C}$ is the concatenation operator. Different from [Sung et al., 2018] we don't restrict $r_{i,j}$ to be in range of 0 to 1, we rather use $r_{i,j}$ as logits used in softmax classification

$$p_D(y = k|\mathbf{x}_j) = \frac{\exp(r_{k,j})}{1 + \sum_{i=1}^N \exp(r_{i,j})} \quad (10)$$

We adopt the simple trick proposed in [Salimans et al., 2016] by setting the logit of the fake class to 0, which is corresponding to the constant 1 appearing in denominator, to model $p_D(N+1|\mathbf{x})$ which is the probability that input data is fake. Thus we can train our model, MetaGAN with RN, directly using loss eq. 5 and eq. 6.

3.3 GENERATOR

We use a conditional generative model to generate fake data that is close to the real data manifold in one specific task $\mathcal{T}$. To do so, we first compress the information in the task's support dataset with a dataset encoder E into vector $h_{\mathcal{T}}$, which contains sufficient statistics for the data distribution of task $\mathcal{T}$. Then $h_{\mathcal{T}}$ is concatenated with random noise input z to be provided as input to the generator network. Inspired by the statistic network proposed in [Edwards and Storkey, 2017], our dataset encoder is composed of two modules:

Instance-Encoder Module The Instance-Encoder is a neural network that learns a feature representation for each individual data example in the dataset $S_{\mathcal{T}}^s$. It maps each data example $\mathbf{x}_i \in S_{\mathcal{T}}^s$ to feature space $e_i = \textit{Instance-Encoder}(\mathbf{x}_i)$.

Feature-Aggregation Module The Feature-Aggregation module takes each embedded feature vector e_i as input and produce the representation vector $h_{\mathcal{T}}$ for the whole task training set. Feasible aggregation methods include average pooling, max pooling and other element-wise aggregation operators. We use average pooling following [Edwards and Storkey, 2017] in our MetaGAN model.

By integrating an Instance-Encoder module and a Feature-Aggregation Module, the instance-encoder is encouraged to learn a representation such that averaging different samples in the learned feature space makes sense. Also, feature-aggregation makes it harder for the generator to simply reconstruct its inputs, which can lead to mode dropping [Che et al., 2017].

3.4 LEARNING SETTINGS

In this section we show that both supervised few-shot learning and semi-supervised few-shot learning can be unified in the MetaGAN framework.

Supervised Few-Shot Learning Supervised learning is the most common learning setting of few-shot classification models. For a task $\mathcal{T} \sim P(\mathcal{T})$, since an unlabeled set $S^u_\mathcal{T}$ and $Q^u_\mathcal{T}$ is not available, we use the labeled set $S^s_\mathcal{T}$ and $Q^s_\mathcal{T}$ to replace them respectively in loss eq. 1 and eq. 7.

Sample-Level Semi-Supervised Few-Shot Learning Sample-level semi-supervised learning follows the same setup as [Ren et al., 2018], where unlabeled data examples are available in each task. While our model is flexible enough to deal with different sets of unlabeled examples in the support set and the query set, for a task $\mathcal{T} \sim P(\mathcal{T})$ we only use a single unlabeled set of examples $U_\mathcal{T}$ to follow the same training scheme in [Ren et al., 2018], for a better comparison with our baseline models.

Specifically, for MetaGAN with MAML, we set $S^u_\mathcal{T} = S^s_\mathcal{T}$ and $Q^u_\mathcal{T} = U_\mathcal{T}$. For MetaGAN with RN, we set $S^u_\mathcal{T} = \emptyset$ and $Q^u_\mathcal{T} = U_\mathcal{T}$ in loss eq. 1 and eq. 7.

Task-Level Semi-Supervised Few-Shot Learning For Task-level semi-supervised learning, the training dataset $\{\mathcal{T}_i\}_{i=1}^M$ consisting of labeled tasks and unlabeled tasks. For labeled tasks we simply follow the supervised learning setting described above. For unlabeled tasks, we omit the supervised loss term by setting $Q^s_\mathcal{T} = \emptyset$ and $S^s_\mathcal{T} = \emptyset$ in loss eq. 1 and eq. 7.

As proposed in [Salimans et al., 2016] we adopt the "feature matching loss" as the generator loss $\mathcal{L}_G$ in both sample-level and task-level semi-supervised few-shot learning.

4 WHY DOES METAGAN WORK?

In this section, we introduce intuition as well as theoretical justifications of MetaGAN, which motivate various improvements we made on the model.

In a few-shot classification problem, the model tries to optimize a decision boundary for each task with just a few samples in each class. Obviously this problem is impossible if no information can be learned from other tasks, as there are so many possible decision boundaries to separate the few samples apart and most of them will not generalize. Meta-learning tries to learn a shared strategy across different tasks to form decision boundaries from few samples, in the hope that this strategy is able to generalize to new tasks.

Although this is reasonable, there can be some problems. For example, some objects look more similar than others. It may be easier to form a decision boundary between a cat and a car than between a cat and a dog. If the training data does not contain tasks that try to separate a cat and a dog, it may feels difficult to extract the correct features to separate these two classes of objects. However, on the other hand, the expectation to have all kinds of class combinations during training leads to the combinatorial explosion problem.

This is where our proposed MetaGAN formulation helps. Just as for the case of doing semi-supervised learning with GANs, we don't expect our generator to generate data that is exactly on the true data manifold. Instead, it is better that the generator is able to generate data a bit off the data manifold of each class, cf. fig. 1. This forces our discriminator to learn a much sharper decision boundary. Instead of only learning to separate cats and dogs, the discriminator of MetaGAN is forced to learn not only what are real cats or dogs, but also what are fake data generated from where is a bit off the cat and dog manifold. The discriminator thus has to extract features strong enough to decide the boundary of the real data manifold, which helps to separate different classes apart. Moreover, the separation between real/fake classes is independent of the class combinations selected during the few-shot learning process.

Following the ideas behind the theoretical justifications studied in the semi-supervised learning setting, we provide similar justifications in the few-shot learning problem. We include the formal statement of the assumptions in the supplemental material.

First, as in [Dai et al., 2017], for a specific task $\mathcal{T}$, we assume that the classifier relies on a feature extractor $f_\mathcal{T}$ to perform classification. We also make the assumption that $G(\cdot; \mathcal{T})$ is a "separating complement generator" (which we define in the supplemental material) for each task $\mathcal{T}$. Intuitively this means that the generator $G(z; \mathcal{T})$ satisfies two conditions: 1) the generator distribution $p^\mathcal{T}_G$ has a

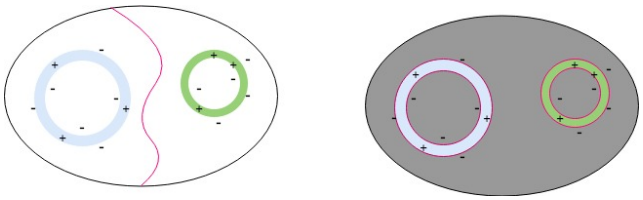

Figure 1: Left: decision boundary without metaGAN. Right: decision boundary with metaGAN. We use red curves to denote the decision boundary. Blue area in figure represents class A, green area represents class B, and gray area represents fake class. We use $+$ to denote real samples and $-$ to denote fake samples generated.

high density region that is disjoint with the data manifold of all classes; 2) This high density region of $p_G^{\mathcal{T}}$ can separate manifolds of different classes.

Then by following arguments similar to those in [Dai et al., 2017], we can prove the following:

Theorem 1 *Let $G_{\mathcal{T}}$ be a separating complement generator in each task $\mathcal{T}$ sampled from $P(\mathcal{T})$. Denote $S_{\mathcal{T}}$ the support set and $F_{\mathcal{T}}$ the generated fake dataset. We assume our learned meta-learner is able to learn a classifier $D_{\mathcal{T}}$ which obtains a strong correct decision boundary on the augmented support set$(S_{\mathcal{T}}, F_{\mathcal{T}})$. Then if $|F_{\mathcal{T}}| \to +\infty$, then $D_{\mathcal{T}}$ can almost surely correctly classify all real samples from the data distribution $p_{\mathcal{T}}(x)$ of the task.*

The theorem is saying that if we have a generator that is neither too good nor too bad, but can generate data around the the real class manifold and have a high density region that can help separating different classes apart, then the generated data together with a few real data can help us determine the correct decision boundary.

5 EXPERIMENTS

5.1 DATASETS

Omniglot is a dataset consisting of handwritten character images from 50 languages. There are 1623 classes of characters with 20 examples within each class. Following prior training and the evaluation protocol used in [Vinyals et al., 2016], we downsampled all images to 28×28 and randomly split the dataset into 1200 classes for traininig and 432 classes for testing. The same data augmentation techniques proposed by [Santoro et al., 2016] are utilized, randomly rotating each image by a multiple of 90 degrees to form new classes.

Mini-Imagenet is a modified subset of the well-known ILSVRC-12 dataset, consisting of 84×84 colored images from 100 classes with 600 random samples in each class. We follow the same class split as in [Ravi and Larochelle, 2017], that takes 64 classes for training, 16 classes for validation and 20 classes for testing.

5.2 SUPERVISED FEW-SHOT LEARNING

On the Omniglot dataset, MetaGAN with MAML shares the same discriminator network architecture and most model hyper-parameters setup with vanilla convolutional MAML[Finn et al., 2017]. We set the meta batch-size to 16 for 5-way classification and 8 for 20-way classification to fit the memory limit of the GPU. For MetaGAN with RN, we batch 15 query images for each class for both 1-shot 5-way and 5-shot 5-way classification, and we batch 5 query images for each class for 1-shot 20-way and 5-shot 20-way task. We set the meta batch-size of MetaGAN with RN model to 1 in our all experiments.

On Mini-Imagenet dataset, we train our MetaGAN with the MAML model using the first-order approximation method with 1 gradient step as proposed in [Finn et al., 2017], due to the consideration of computational cost.

For the conditional generator we adopt a ResNet-like architecture inspired by [Gulrajani et al., 2017] in both models; see more details of the architecture of the generator in supplemental material.

Model	5-way Acc.		20-way Acc.	
	1-shot	5-shot	1-shot	5-shot
Neural Statistician	98.1	99.5	93.2	98.1
Prototypical Nets	98.8	99.7	96.0	98.9
MAML	98.7 ± 0.4	$\mathbf{99.9 \pm 0.1}$	95.8 ± 0.3	98.9 ± 0.2
Ours: MetaGAN + MAML	99.1 ± 0.3	99.7 ± 0.21	96.4 ± 0.27	98.9 ± 0.18
Relation Net	99.6 ± 0.2	99.8 ± 0.1	97.6 ± 0.2	99.1 ± 0.1
Ours: MetaGAN + RN	$\mathbf{99.67 \pm 0.18}$	99.86 ± 0.11	$\mathbf{97.64 \pm 0.17}$	$\mathbf{99.21 \pm 0.1}$

Table 1: Few-shot classification results on Omniglot.

Model	5-way Acc.	
	1-shot	5-shot
Prototypical Nets	49.42 ± 0.78	68.20 ± 0.66
MAML(5 gradient steps)	48.70 ± 1.84	63.11 ± 0.92
MAML(5 gradient steps, first order)	48.07 ± 1.75	63.15 ± 0.91
MAML(1 gradient step, first order)	43.64 ± 1.91	58.72 ± 1.20
Ours: MetaGAN + MAML(1 step, first order)	46.13 ± 1.78	60.71 ± 0.89
Relation Net	50.44 ± 0.82	65.32 ± 0.7
Ours: MetaGAN + RN	$\mathbf{52.71 \pm 0.64}$	$\mathbf{68.63 \pm 0.67}$

Table 2: Few-shot classification results on Mini-Imagenet.

We use the Adam [Kingma and Ba, 2014] optimizer with initial learning rate as 0.001, $\beta_1 = 0.5$ and $\beta_2 = 0.9$ to train both generator and discriminator networks. For Omniglot we decay the learning rate starting from 10K batch updates, and cut it in half for every $10K$ following updates. For Mini-Imagenet we decay the learning rate starting from 30K batch updates, and cut it in half for every 10K updates.

We present our results of 5-way and 20-way few-shot classification for Omniglot dataset in table 1, and show results of Mini-Imagenet dataset in table 2. We see that our proposed MetaGAN consistently improves over baseline classifiers, and achieves comparable or outperforms state-of-the-art performance on the challenging Mini-Imagenet benchmark.

5.3 SAMPLE-LEVEL SEMI-SUPERVISED FEW-SHOT LEARNING

As introduced in section 3.4, we evaluate the effectiveness of our proposed MetaGAN in the sample-level semi-supervised few-shot learning setting, following a similar training and evaluation scheme without "distractors" to that proposed in [Ren et al., 2018] (We will point out the differences in the scheme later on). For the Omniglot dataset we sample 10% of the images of each class to form the labeled set, and take all remaining data as the unlabeled set. For Mini-Imagenet we sample 40% images of each class as the labeled set, and sample 5 images of each class for each training episode.

Note that our model only leverages unlabeled samples during the training phase, while the refining model proposed in [Ren et al., 2018] uses unlabeled samples in both training (5 samples for each class) and evaluation phases (20 samples for each class). This makes our model acquire strictly less information during evaluation, compared to [Ren et al., 2018]. The classifier trained with our proposed MetaGAN formulation is encouraged to form better decision boundaries by utilizing unlabeled and fake data, and is free from the demands of unlabeled samples during testing, different from the kmeans-based refining model [Ren et al., 2018] which strongly relies on the unlabeled data for testing.

Model	Omniglot 1-shot 5-way	Mini-Imagenet 1-shot 5-way	Mini-Imagenet 5-shot 5-way
Prototypical Nets(Supervised)	94.62 ± 0.09	43.61 ± 0.27	59.08 ± 0.22
Semi-Supervised Inference(PN)	97.45 ± 0.05	48.98 ± 0.34	63.77 ± 0.20
Soft k-Means	97.25 ± 0.10	50.09 ± 0.45	$\mathbf{64.59 \pm 0.28}$
Soft k-Means+Cluster	$\mathbf{97.68 \pm 0.07}$	49.03 ± 0.24	63.08 ± 0.18
Masked Soft k-Means	97.52 ± 0.07	$\mathbf{50.41 \pm 0.31}$	64.39 ± 0.24
Ours: Relation Nets(Supervised)	94.81 ± 0.08	44.24 ± 0.24	58.72 ± 0.31
Ours: MetaGAN + RN	97.58 ± 0.07	50.35 ± 0.23	64.43 ± 0.27

Table 3: sample-level Semi-Supervised Few-shot classification results on Omniglot and Mini-Imagenet.

Model	Omniglot 1-shot 5-way	Mini-Imagenet 1-shot 5-way
Prototypical Net(Supervised)	93.66 ± 0.09	42.28 ± 0.32
Relation Net(Supervised)	93.82 ± 0.07	43.87 ± 0.20
Ours: MetaGAN + RN	$\mathbf{97.12 \pm 0.08}$	$\mathbf{47.43 \pm 0.27}$

Table 4: Task-level Semi-Supervised 1-shot classification results on Omniglot and Mini-Imagenet.

We display the results of sample-level semi-supervised few-shot classification results on Omniglot and Mini-Imagenet in table 3. Though our model cannot be compared with the kmeans refining model directly as discussed above, we obtain comparable state-of-the-art results on both 1-shot and 5-shot tasks, while significantly improving the purely supervised baseline models.

5.4 TASK-LEVEL SEMI-SUPERVISED FEW-SHOT LEARNING

We proposed a new learning setting for the few-shot learning problem in section 3.4: task-level semi-supervised few-shot learning. In this learning setting, existing few-shot learning models[Ravi and Larochelle, 2017, Sung et al., 2018, Ren et al., 2018] are unable to effectively leverage purely unsupervised tasks, which consist of only unlabeled samples in both support set and query set.

To demonstrate that our proposed MetaGAN model can successfully learn from unsupervised tasks, we create new splits of Omniglot and Mini-Imagenet datasets. For the Omniglot dataset we randomly sample 10% of classes from the training set as a labeled set of classes, and the remaining 90% classes as an unlabeled set of classes. For Mini-Imagenet dataset we randomly sample 40% as labeled classes and the remaining 60% are unlabeled. The validation set and test set of each dataset remains unchanged, using all classes to evaluate the performance of models. During training time, we sample supervised tasks only from the labeled set of classes, and sample unsupervised tasks from the unlabeled set of classes. We alternate between sampled supervised tasks and sampled unsupervised tasks for training the MetaGAN model, while we only use sampled supervised tasks to train the baseline model.

We show the results of task-level semi-supervised few-shot classification results on Omniglot and Mini-Imagenet in table 4. By integrating the baseline model into the MetaGAN framework, the model effectively learned to utilize the unsupervised tasks for helping the classification task, showing that MetaGAN can learn transferable knowledge from totally unsupervised tasks.

6 CONCLUSION

We propose MetaGAN, a simple and generic framework to boost the performance of few-shot learning models. Our approach is based on the idea that fake samples produced by the generator can help classifiers learn a sharper decision boundary between different classes from a few samples.

We make an analogy between few-shot learning and semi-supervised learning- both of them have only a few labeled data and both can benefit from an imperfect generator. Then we modified the techniques used for semi-supervised learning with GANs to work in the few-shot learning scenario. We give intuitive as well as theoretical justifications of the proposed approach.

We demonstrated the strength of our algorithm on a series of few-shot learning and semi-supervised few-shot learning tasks. For future work, we plan to extend MetaGAN to the few-shot imitation learning setting.

ACKNOWLEDGEMENT

We thank Intel Corporation for supporting our deep learning related research.

References

Anthreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. 2018. URL https://openreview.net/forum?id=S1Auv-WRZ.

Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*, 2017.

Xi Chen, Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2172–2180. Curran Associates, Inc., 2016.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6510–6520. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7229-good-semi-supervised-learning-that-requires-a-bad-gan.pdf.

Harrison Edwards and Amos Storkey. *Towards a Neural Statistician*. 5th International Conference on Learning Representations (ICLR 2017), 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/finn17a.html.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4565–4573. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning.pdf.

Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to learn using gradient descent. In *Proceedings of the International Conference on Artificial Neural Networks*, ICANN '01, pages 87–94, London, UK, UK, 2001. Springer-Verlag. ISBN 3-540-42486-5. URL http://dl.acm.org/citation.cfm?id=646258.684281.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1DmUzWAW.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2554–2563, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/munkhdalai17a.html.

Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, Tong Wang, and Adam Trischler. Learning rapid-temporal adaptations. *CoRR*, abs/1712.09926, 2017.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *In International Conference on Learning Representations (ICLR)*, 2017.

Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJcSzz-CZ.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1842–1850, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/santoro16.html.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4077–4087. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6996-prototypical-networks-for-few-shot-learning.pdf.

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*, 2018.

Sebastian Thrun. Learning to learn. chapter Lifelong Learning Algorithms, pages 181–209. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0-7923-8047-9. URL http://dl.acm.org/citation.cfm?id=296635.296651.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6385-matching-networks-for-one-shot-learning.pdf.

Supplemental Material

1 THEORETICAL JUSTIFICATION

In this section we define the conditions and prove the theorem stated in the main text. Techniques are similar to [1].

Recall that we say a measurable set U is a support set of probability measure p iff $p(U) = 1$. A measurable set U is called p-dense iff for every point $z \in U$, for any neighborhood $U(z, \epsilon)$ of z, we have $p(U(z, \epsilon)) > 0$.

Fixing a task $\mathcal{T}$, our discriminator first compresses each data point $\mathbf{x}$ to a latent vector representation $\mathbf{z} = f(\mathbf{x})$, and then pass to a linear classifier, with weights $\mathbf{w}_i, i = 1, 2 \cdots K$. We further assume $\mathbf{w}_i, i = 1, 2 \cdots K$ are bounded by a uniform constant C.

we denote $p_{\mathcal{T}}(\mathbf{z})$ the distribution of latent representations of task $\mathcal{T}$. We assume that $p_{\mathcal{T}}$ has compact support B. Without loss of generality, we can also assume B is convex, otherwise we can take its convex closure. We also denote the probability distribution of latent distributions of class i as $p^i_{\mathcal{T}}(\mathbf{z}), i = 1 \dots K$. We define an open domain $U \subset \mathbb{R}^n$ is an ϵ-support of a probability measure p, if $p(U) > 1-\epsilon$. We assume that there exists some very small $\epsilon > 0$, we have a set of $U_i, i = 1, 2 \cdots, K$, such that U_i is an ϵ-support of $p^i_{\mathcal{T}}$ for all $i = 1, 2, \cdots K$. We also assume all U_i is disjoint from each other. For the adapted generator $G_{\mathcal{T}}(\mathbf{z})$, we denote its corresponding distribution in latent space as $p^G_{\mathcal{T}}(\mathbf{z})$. Assume $p^G_{\mathcal{T}}$ has $p^G_{\mathcal{T}}$-dense set $S_G \subset B$.

Now we can define what is a "complement separating generator".

Definition 1. *With the above assumptions and notations, we call a generator $G(z; \cdot)$ a complement separating generator if, for any task $\mathcal{T} \sim p_{\mathcal{T}}$, $G(z; \mathcal{T})$ satisfies the following two conditions:*

- *for all $i = 1, 2, \cdots K$, $U_i \cap S_G = \emptyset$.*
- *for all $i, j = 1, 2, \cdots K$, U_i and U_j are pathwise disconnected from each other in $B \setminus S_G$.*

Then we can formally state the main theorem as:

Theorem 1. *Let $G_{\mathcal{T}}$ be a separating complement generator. Denote $S_{\mathcal{T}}$ the support(training) set and $F_{\mathcal{T}}$ the generated fake dataset. We assume our learned meta-learner is able to learn a classifier $D_{\mathcal{T}}$ which obtains strong correct decision boundary on the augmented support set$(S_{\mathcal{T}}, F_{\mathcal{T}})$. More precisely, (1) for $\mathbf{x}, y \in S_{\mathcal{T}}$, $\mathbf{x} \cdot \mathbf{w}_y > \max\{0, \mathbf{x} \cdot \mathbf{w}_i\}$ for all $i \neq y$. (2) for $f(\mathbf{x}) \in F_{\mathcal{T}}$, $f(\mathbf{x}) \cdot \mathbf{w}_i < 0$ for all $i \leq K$.*

Then if $|F_{\mathcal{T}}| \to +\infty$, then $D_{\mathcal{T}}$ can almost surely correctly classify all real samples from the data distribution $p_{\mathcal{T}}(x)$ of the task.

Proof. We first need to prove when $|F_{\mathcal{T}}| \to +\infty$, for all $\mathbf{z} \in S_G$, we have almost surely $\max_{i \leq K} \mathbf{w}_i \cdot \mathbf{z} \leq 0$. The detailed proof is subtle. Here we only give a sketch. From the assumption that S_G is $p^G_{\mathcal{T}}$-dense, one can easily deduce that when $|F_{\mathcal{T}}| \to +\infty$, the points $F_{\mathcal{T}}$ become dense in S_G. More precisely, for any $\epsilon > 0$, any $\mathbf{z} \in S_G$, when $|F_{\mathcal{T}}| \to +\infty$, then almost surely there exits $\mathbf{z}' \in F_{\mathcal{T}}$, such that $|\mathbf{z} - \mathbf{z}'| < \epsilon$. From the assumption $\mathbf{w}_i, i = 1, 2 \cdots K$ are bounded by a uniform constant C, we can get almost surely $\max_{i \leq K} \mathbf{w}_i \cdot \mathbf{z} \leq 0$.

Then we prove by contradiction. If for any task $\mathcal{T}$, D successfully adapted to a support set $(S_{\mathcal{T}}, F_{\mathcal{T}})$, without loss of generality, we can assume $S_{\mathcal{T}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^K$ is one-shot. If there is a data

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

point $(\mathbf{x}, y)$ which is classifier incorrectly, namely there exists some $j \neq y$, such that $\mathbf{w}_j \cdot f(\mathbf{x}) > \mathbf{w}_y \cdot f(\mathbf{x}) > 0$. In the mean time $\mathbf{w}_j \cdot f(\mathbf{x}_j) > 0$. So for all $\alpha \in [0, 1]$, $\mathbf{w}_j \cdot [\alpha f(\mathbf{x}_j) + (1-\alpha) f(\mathbf{x})] > 0$. This contradicts with two facts: 1) U_y and U_j are pathwise disconnected from each other in $B \setminus S_G$; 2) almost surely $\max_{i \leq K} \mathbf{w}_i \cdot \mathbf{z} \leq 0$, for all $\mathbf{z} \in S_G$.

So the theorem is proved. □

2 ALGORITHMS FOR TRAINING METAGAN WITH MAML

We describe the detailed algorithm for training MetaGAN with MAML model as following:

Algorithm 1 MetaGAN with MAML

$G(\mathbf{z}, \mathcal{T})$**: Generator network parameterized by** θ_g.
$D(x)$**: Discriminator network. parameterized by** θ_d.

Initialize θ_g, θ_d randomly.
while not done **do**
  Sample a batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$. ▷ Discriminator Update
  for all $\mathcal{T}_i$ **do**
    Get K real samples $\mathcal{D}_r = \{\mathbf{x}^{(i)}, y^{(i)}\}$ from $\mathcal{T}_i$.
    Sample K generated samples $\mathcal{D}_f = \{\mathbf{x}^{(j)}\} = G(\mathbf{z}^{(j)}, \mathcal{T}_i)$ from $G(\mathbf{z}, \mathcal{T}_i)$.
    Evaluate discriminator loss $\ell_D^{\mathcal{T}_i}$ with D_r and D_f.
    Compute adapted discriminator parameters $\theta'_{d_i} = \theta_d - \alpha \nabla_{\theta_d} \ell_D^{\mathcal{T}_i}$.
  end for
  Update θ_d using loss $\mathcal{L}_D$
  Sample a batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$. ▷ Generator Update
  for all $\mathcal{T}_i$ **do**
    Sample K generated samples $\mathcal{D}_f = \{\mathbf{x}^{(j)} = G(\mathbf{z}^{(j)}, \mathcal{T}_i)\}$ from $G(\mathbf{z}, \mathcal{T}_i)$.
    Compute adapted discriminator parameters $\theta'_{d_i} = \theta_d - \alpha \nabla_{\theta_d} L_D$.
    Compute generator loss gradient $\nabla_{\theta_g} L_G^{\mathcal{T}_i}$ with the adapted discriminator.
  end for
  Update generator parameters θ_g with accumulated generator loss gradients.
end while

3 GENERATOR AND DISCRIMINATOR ARCHITECTURE

3.1 GENERATOR

We describe the generator architecture used in Omniglot models in table 3.2. The generator used in Mini-Imagenet models are similar. Please refer to provided code [1] for more details on the network architecture and training hyperparameters.

3.2 DISCRIMINATOR

For both model MetaGAN with MAML and MetaGAN with RN, we adopt the same neural network architecture as MAML and RN respectively.

[1] https://github.com/sodabeta7/MetaGAN

$2\times$ { *conv2d* 64 feature maps with 3×3 kernels and Leaky-Relu activations }
conv2d 64 feature maps with 3×3 kernels, stride 2 and Leaky-Relu activations
$2\times$ {*conv2d* 128 feature maps with 3×3 kernels and Leaky-Relu activations }
conv2d 128 feature maps with 3×3 kernels, stride 2 and Leaky-Relu activations
$2\times$ { *conv2d* 256 feature maps with 3×3 kernels and Leaky-Relu activations }
conv2d 256 feature maps with 3×3 kernels, stride 2 and Leaky-Relu activations
fully-connected layer with 256 units and Leaky-Relu activations
sample-dropout and *concatenation* with number of samples
average pooling within each dataset
concatenation embeded features with noise input z
upsample conv2d 512 feature maps with 3×3 kernels and Leaky-Relu activations with residual connection
upsample conv2d 256 feature maps with 3×3 kernels and Leaky-Relu activations with residual connection
upsample conv2d 128 feature maps with 3×3 kernels and Leaky-Relu activations with residual connection
upsample conv2d 1 feature maps with 3×3 kernels and Leaky-Relu activations with residual connection

Table 1: Omniglot Conditional Generator

References

[1] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6510–6520. Curran Associates, Inc., 2017.