



# Domain Adaption in One-Shot Learning

Nanqing Dong<sup>1,2</sup>(✉) and Eric P. Xing<sup>2</sup>

<sup>1</sup> Cornell University, Ithaca, NY 14850, USA  
nd367@cornell.edu

<sup>2</sup> Petuum Inc., Pittsburgh, PA 15217, USA

**Abstract.** Recent advances in deep learning lead to breakthroughs in many machine learning tasks. Due to the data-driven nature of deep learning, the training procedure often requires large amounts of manually annotated data, which is often unavailable. One-shot learning aims to categorize the new classes unseen in the training set, given only one example of each new class. Can we transfer knowledge learned by one-shot learning from one domain to another? In this paper, we formulate the problem of domain adaption in one-shot image classification, where the training data and test data come from similar but different distributions. We propose a domain adaption framework based on adversarial networks. This framework is generalized for situations where the source and target domain have different labels. We use a policy network, inspired by human learning behaviors, to effectively select samples from the source domain in the training process. This sampling strategy can further improve the domain adaption performance. We investigate our approach in one-shot image classification tasks on different settings and achieve better results than previous methods. Code related to this paper is available at: <https://github.com/NanqingD/DAOSL>.

**Keywords:** One-shot learning · Domain adaption  
Adversarial networks · Reinforcement learning  
Distance metric learning · Cognitive science

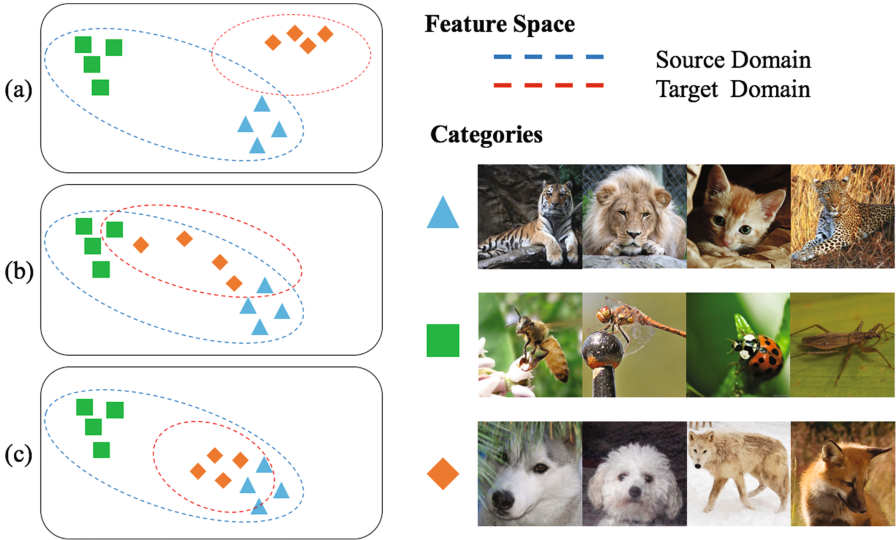
## 1 Introduction

Convolutional Neural Networks (CNNs) have led significant progress in the domain of computer vision such as image recognition [12], object detection [22] and semantic segmentation [19]. When modern visual recognition systems can benefit from large image datasets like ImageNet [6] and PASCAL VOC [8], deep learning methods still face the obstacle of requiring large amounts of manually annotated data. With the knowledge transfer, humans can tell the difference between up to 30,000 object categories [4]. Especially, children can recognize new objects quickly in their learning phase with proper guidance, even they only see the examples for few times. These motivate the study of one-shot learning, where one annotated example is available for each class to predict. One approach is based on Bayesian statistics. Li et al. [18] proposed a complex framework with

strong probabilistic hypothesis using generative object category model and variational Bayesian expectation maximization (VBEM). Another approach is *meta-learning* [27]. Santoro et al. [23] attacked the problem by learning to memorize unseen classes with a Memory Augmented Neural Network (MANN). Ravi and Larochelle [21] utilized a Long Short-Term Memory network (LSTM) [13] as a meta-learner to optimize the learner. There are two challenges in meta-learning approach. The gradient-based optimization usually requires large amounts of labeled data, and the random initiation can have unpredictable effects on the learner. In this work, we focus on a simpler but more efficient approach, the metric-based approach. The metric-based approach projects the raw images into a learned feature space and classifies the image based on a certain distance metric. Due to the simplicity and efficiency, the metric-based approach has been applied in the industry for tasks like face recognition and person re-identification.

The metric-based methods can achieve state-of-the-art performance in one-shot classification tasks, but the accuracy can be easily influenced when the test data comes from a different distribution [24, 29]. Domain adaption means learning a mapping from the source domain to the target domain with the presence of a *shift* between two data distributions, so a predictor trained on the source domain can be applied on the target domain [9, 32]. In our case, a good one-shot learning system can be applied to the target domain with classes unseen in the source domain, just like a student with only basic knowledge in English can differentiate Greek letters with just a glance. In previous domain adaption methods, examples in the source domain are assumed to have equal importance in the training process if there is no prior knowledge. Assume there is a learner wants to learn animals of Canidae family from an incomplete encyclopedia which only includes sections about Felidae and Insecta. Given only a few pictures about Canidae, the learner may find that dogs and cats share more features than bugs. After few trials, the learner should pay more attention to Felidae than Insecta, even though the learner may not have a clear definition of Canidae.

In this paper, we formulate the domain adaption problem in one-shot learning. Fused by recent advances in one-shot learning and domain adaption, we propose an adversarial framework for domain adaption in one-shot learning. We train the one-shot classifier and auxiliary domain discriminator simultaneously. Motivated by the behavior of human learners, we propose to use a policy gradient method [25, 26, 30] to select the samples from the source domain in the training phase, which is different from the traditional random sample selection. By incorporating the reinforced sample selection process in our adversarial framework, we further improve the domain adaption performance in one-shot learning. We also discuss the how the proposed sampling strategy is linked to *distance metric learning* (DML) [31] and *curriculum learning* [3]. The concept is illustrated in Fig. 1. This work focuses on a difficult situation where source domain and target domain do not have any overlap in categories. We investigate our approach in one-shot image classification tasks with different settings. To the best of our knowledge, there is no similar work in either one-shot learning or domain adaption.



**Fig. 1.** Illustration of the motivation. Examples are embedded to certain feature spaces under three situations. (a) No domain adaption. (b) Domain adaption with random sample selection. (c) Domain adaption with reinforced sample selection.

## 2 Related Work

Many works [16, 21, 23, 24, 29] have contributed to  $q$ -shot learning, here  $q > 0$  means the number of labeled examples for the new class unseen in the training set. One-shot learning is an extreme case when there is only one example for each new category. Compared with the Bayesian approach [18] and the meta-learning approach [21, 23] in one-shot learning, recently proposed metric-based methods [24, 29] achieve state-of-the-art performance with fewer parameters and simpler optimization settings. Given an *episode*, which consists of a query image and a support set of images, a metric-based method computes a certain similarity measure between the embedded query image and each of the embedded support image, and then uses the similarities as weights of a weighted nearest neighbor classifier to predict the label of the query image.

Domain adaption can also be accomplished through adversarial training after Goodfellow et al. first introduced adversarial networks in generative adversarial networks (GANs) [11]. A standard classifier can be decomposed into two parts, a feature extractor, and a label predictor. Domain-adversarial neural network (DANN) proposed a gradient reversal layer to connect an auxiliary domain discriminator with feature extractor for unsupervised domain adaption (UDA). One problem for DANN is that the domain discriminator converges quickly, which can cause the gradient to vanish [28]. Another unsupervised domain adaption method is adversarial discriminative domain adaption (ADDA) [28]. ADDA uses different feature extractors for each domain. The source feature extractor and

predictor (classifier) is trained on the source domain first. Then the source feature extractor is fixed and the predictor is replaced with a domain discriminator. The target feature extractor is trained with on the target domain in an adversarial fashion to align the representations of the target domain with the representations of the source domain. The problem with this method is that the performance on target domain is highly dependent on the predictor trained on the source domain [7]. With limited training examples, there is no guarantee of the quality of the predictor. In other words, the optimization objective for domain adaption and prediction on the source domain may not be aligned in one-shot setting. The most related recent work is few-shot adversarial domain adaption (FADA) [20], which focus on supervised domain adaption. FADA pairs examples from source domain with examples from target domain as input for domain classifier. Because target labels are used for pairing in the training process, FADA is a supervised domain adaption. For previous domain adaption methods, source domain and the target domain are required to have the same classes. But in one-shot learning, this constraint is relaxed.

### 3 Adversarial Domain Adaption with Reinforced Sample Selection

To address the problems listed in Sect. 2, we present our methodology for domain adaption in one-shot learning. Firstly, we formulate the domain adaption problem in metric-based one-shot learning. Secondly, we propose an adversarial domain adaption framework without stage-wise training scheme. Thirdly, we introduce the concept of overgeneralization in domain adaption. Finally, we propose reinforced sample selection as a solution to overgeneralization. The complete pipeline is illustrated in Fig. 2.

#### 3.1 Problem Definition

Given a source domain  $S$  as training data and a target domain  $T$  as test data, domain adaption learns a mapping between  $S$  and  $T$ . We denote

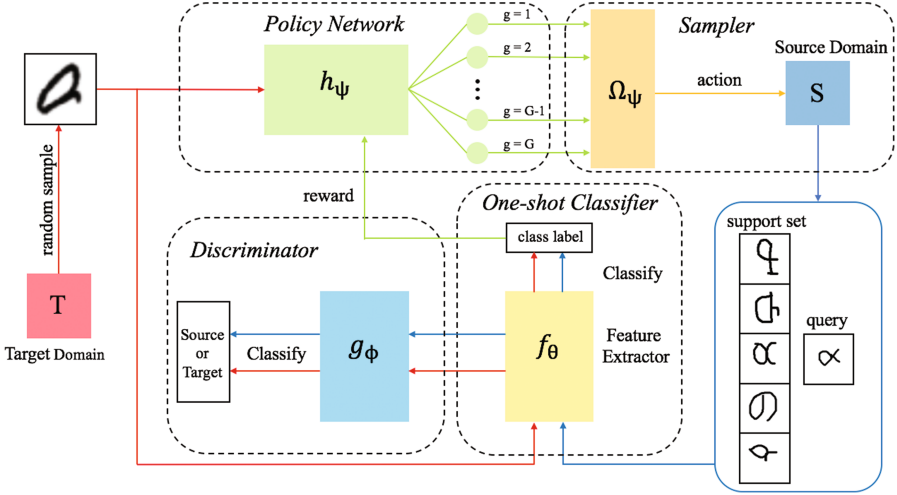
$$S = \{(x_1, y_1), \dots, (x_{N_S}, y_{N_S})\},$$

where  $x_i$  represents an example from  $S$  and  $y_i \in Y_S$  with  $Y_S = \{1, \dots, K_S\}$  is the corresponding label.  $x_i$  is multi-dimensional, for simplicity, we assume it can be represented as a  $D$ -dimension feature vector,  $x_i \in \mathbb{R}^D$ . We denote

$$T = \{ \{(\bar{x}_1, \bar{y}_1), \dots, (\bar{x}_t, \bar{y}_t)\}, \{\bar{x}_{t+1}, \dots, \bar{x}_{N_T}\} \} f,$$

where  $\bar{x}_j$  represents an example from  $D$  and  $\bar{y}_j \in Y_T$  with  $Y_T = \{K_S+1, \dots, K_S+K_T\}$ . In this paper, we assume  $K_S > K_T$  and  $N_T \gg t$ . We focus on  $Y_S \cap Y_T = \emptyset$ , which is the most difficult situation for the learner.

A  $K$ -way  $q$ -shot learning task is defined as: Given  $q$  labelled examples for each of  $K$  classes that have not been seen before as support set, classifying unlabelled query examples into one of  $K$  classes [16, 21, 23, 24, 29]. Let  $f_\theta$  denotes an



**Fig. 2.** Illustration of the model architecture. The figure depicts the data flow in the training phase. At the beginning of an episode, a random sample from the target domain goes through the feature extractor and discriminator for the first pass. Then policy network receives the sample and outputs a sampling policy to the sampler. The sampler selects support set and query image from the source domain based on the policy. The one-shot classifier uses the support set and query image to update the feature extractor. The target sample goes through the one-shot classifier with the support set again to calculate the reward. The reward is used to update the policy network. The details are described in Sects. 3.1, 3.2 and 3.4.

embedding function with parameters  $\theta$ .  $f_\theta$  embeds the input to a  $M$ -dimensional representation,  $f_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^M$ .  $d$  denotes a similarity measure function. For  $q = 1$  and  $k \in \{1, \dots, K\}$ , with support set  $\{(x_k, y_k)\}$  and query example  $\mathbf{x}$ , the probability of  $\mathbf{x}$  belongs to class  $k$  is defined as

$$p_\theta(y = k|\mathbf{x}) = \frac{\exp(d(f_\theta(\mathbf{x}), f_\theta(x_k)))}{\sum_{k'=1}^K \exp(d(f_\theta(\mathbf{x}), f_\theta(x_{k'})))}. \quad (1)$$

Under this definition, the metric-based one-shot learning problem can be formulated as a standard multiclass classification problem.

In a naive transfer learning setting, the classifier trained on  $S$  is finetuned on  $T$  to offset the shift between  $S$  and  $T$ , where  $t$  is expected to be much larger than  $K_T$  to produce a good result. However, we have  $t = K_T$  in one-shot learning. It is not practical to finetune the one-shot classifier with  $K_T$  labeled examples. We argue that, in one-shot learning, we can train  $f_\theta$  on  $S$  and use Eq. 1 to predict labels for  $\{\bar{x}_{t+1}, \dots, \bar{x}_{N_T}\}$  based on  $\{(\bar{x}_1, \bar{y}_1), \dots, (\bar{x}_t, \bar{y}_t)\}$ .  $f_\theta$  is a projection function and  $f_\theta(\bar{x}_j)$  represents the feature vector when  $\bar{x}_j$  is projected to some feature space. The objective of domain adaption in one-shot learning can be defined as follows: We want to find the optimal  $\theta$ , such that  $f_\theta(\bar{x}_j)$  has the most discriminative features for a classifier to correctly assign a label to it. The

loss for this objective is hard to be defined explicitly. To alleviate this problem, we use adversarial networks.

### 3.2 Adversarial Domain Adaption

The state-of-the-art methods for adversarial domain adaption (ADA) usually consist of multi-stage training paradigms [20,28]. We argue that, in one-shot learning, the training of the one-shot classifier and the discriminator should be optimized simultaneously. One critical issue for one-shot learning is overfitting [24], while stage-wise training can cause overfitting in each stage and the overfitting is intractable. The basic task of domain adaption is to make the original domain of representations  $f_\theta(x_i)$  and  $f_\theta(\bar{x}_j)$  indistinguishable [2]. As in [11], we introduce a discriminator which is a parametric function of  $g_\phi$ .  $g_\phi$  takes the embedded features as input and outputs a probability score for the input comes from source domain,  $g_\phi : \mathbb{R}^M \rightarrow \mathbb{R}$ . The discriminator is then a binary classifier,

$$p_\phi(y = 1|f_\theta(x_i)) = \frac{\exp(g_\phi(f_\theta(x_i)))}{1 + \exp(g_\phi(f_\theta(x_i)))}, \quad (2)$$

$$p_\phi(y = 0|f_\theta(\bar{x}_j)) = \frac{1}{1 + \exp(g_\phi(f_\theta(\bar{x}_j)))}. \quad (3)$$

The one-shot classifier and the domain discriminator are optimized alternatively.

Given fixed  $f_\theta$ ,  $g_\phi$  is optimized to maximize the probability of correctly differentiating  $f_\theta(\bar{x}_j)$  from  $f_\theta(x_i)$ . The binary cross entropy loss is defined as

$$J_\phi = -\frac{1}{B_S} \sum_i \log(p_\phi(y = 1|f_\theta(x_i))) - \frac{1}{B_T} \sum_j \log(1 - p_\phi(y = 0|f_\theta(\bar{x}_j))), \quad (4)$$

where  $B_S$  is the batch size of samples from  $S$  and  $B_T$  is the batch size of samples from  $T$  for the discriminator updating step.

Given fixed  $g_\phi$ ,  $f_\theta$  is optimized to achieve two goals at the same time. Firstly, we want to train a one-shot classifier on  $S$  which can assign the correct label for each query example. The multiclass cross entropy loss is defined as

$$J_{cls} = -\frac{1}{B_S} \sum_i \sum_k y_k \log(p_\theta(y = k|x_i)), \quad (5)$$

where  $B_S$  is the batch size of samples from  $S$  and  $y_k$  is binary, denoting whether class label  $k$  is the correct classification for  $x_i$ . Secondly, we want to use the embedding function to project examples from both  $S$  and  $T$  to a feature space that  $S$  and  $T$  have high similarity. Adversarial networks transform the original problem of how to maximize the similarity between  $S$  and  $T$  into how to make them indistinguishable. So  $f_\theta$  is also trained to make discriminator to assign the wrong labels to  $f_\theta(\bar{x}_j)$ , where the optimization goal is to maximize the loss that  $f_\theta(\bar{x}_j)$  is classified from  $T$ . Following the practice of [11], the optimization

---

**Algorithm 1.** Training an episode of adversarial domain adaption in a  $K$  way one-shot learning task, where  $K \leq K_S$  and  $K \leq K_T$ . `sample()` denotes a function that samples fixed number of elements from a set. `sampleSupport()` denotes a function that samples the support set of image and label pairs from  $S$  and `sampleQuery()` denotes a function that samples the query image and label from  $S$  based on support set, same as [29, 24]. All samples are sampled uniformly without replacement.

---

**Input** :  $S, T, \theta, \phi$   
**Output**:  $\theta, \phi$   
 $Support \leftarrow \{\}$   
 $Query \leftarrow \{\}$   
 $Target \leftarrow \text{sample}(\{\bar{x}_j\}, B_T)$   
**for**  $b \in \{1, \dots, B_S\}$  **do**  
   $support \leftarrow \text{sampleSupport}(S)$ ; Add  $support$  to  $Support$   
   $query \leftarrow \text{sampleQuery}(S, support)$ ; Add  $query$  to  $Query$   
**end**  
Calculate  $J_\phi$  with  $Support, Query$  and  $Target$  by (4)  
Update  $\phi$  by minimizing  $J_\phi$   
Calculate  $J_{cls}$  with  $Support$  and  $Query$  by (5)  
Calculate  $J_{adv}$  with  $target$  by (6)  
Update  $\theta$  by minimizing  $J_\theta$

---

problem can also be seen as minimization of loss that  $f_\theta(\bar{x}_j)$  is classified from  $S$ , so this adversarial loss can be defined as

$$J_{adv} = -\frac{1}{B_T} \sum_j \log(p_\phi(y = 1 | f_\theta(\bar{x}_j))), \quad (6)$$

where  $B_T$  is the batch size of samples from  $T$ . The total loss for classifier updating step is then

$$J_\theta = J_{cls} + \lambda_{adv} J_{adv}, \quad (7)$$

where  $\lambda_{adv}$  is a weight for adversarial loss. The training for adversarial domain adaption in one-shot learning is illustrated in Algorithm 1. Note,  $\{\bar{y}_j\}$  is not used in the optimization for either  $\theta$  or  $\phi$ , so adversarial domain adaption in one-shot learning is unsupervised.

### 3.3 Overgeneralization

Generalization is an important ability for humans and animals to acquire knowledge in one circumstance and apply the knowledge to new situations [10]. In contrast, discrimination is the ability to discriminate different stimuli. Humans can not memorize all the discriminative features with limited memory. Generalization can help humans to save memory in the learning process. Domain adaption in one-shot learning can be seen as a mixture of generalization and discrimination. In this study, we observe a phenomenon that the learner learns too much

in  $S$  and performs worse on  $T$ . We call this phenomenon overgeneralization for domain adaption in one-shot learning.

Overgeneralization can be caused by the misaligned optimization objectives. The learner’s goal is to accurately classify the examples from  $T$ , while ADA tries to minimize the distance between the distributions of  $S$  and  $T$  in a projected space [2, 11]. There is no supervision for  $T$ , thus the extracted features are dependent on  $S$ . With limited memory, the learner memorizes more generalized features from  $S$  but misses the features that are most discriminative for  $T$ , especially when  $K_S \gg K_T$ . Previous methods [20, 28] have shown that ADA performs well when  $S$  and  $T$  share same categories. One solution is then to find a subset of  $S$ , so the distance between the distributions of the subset and  $T$  is minimal. Note this subset selection problem is not convex or differentiable. We present our solution, reinforced sample selection.

### 3.4 Reinforced Sample Selection

Random sample selection has been widely used in many machine learning tasks to reduce variance and avoid overfitting. In supervised learning, more examples usually help the learner to grasp more discriminative features. However, the large sample size of  $S$  may not help domain adaption in one-shot learning because  $S$  and  $T$  can have totally different categories. The unsupervised domain adaption problem is intractable since there are no labels from  $T$ . The minimization of  $J_{cls}$  can be seen as a regularization of  $f_\theta$  to learn useful features for one-shot learning task on  $S$ . However, there is no guarantee for the performance of  $T$ .

We propose to train the learner to learn the sampling strategy through reinforcement learning, which is in contrast to typical random sample selection. In the domain adaption process, the learning system actively selects samples from  $S$  when it sees an image from  $T$ . To accomplish this, we introduce a policy network to select the categories from  $S$ . In each episode, the support set and query image will be sampled from this selected categories. Be more specific, given an image from  $T$ , the policy network will output a policy for the sampler, and the sampler will sample examples from a subset of  $S$ . The examples sampled from the subset of  $S$  are used to train the one-shot classifier and the domain discriminator. Given  $\bar{x}$  from  $T$ , assume there are  $(x_{sim}, y_{sim}) \in S$  and  $(x_{dis}, y_{dis}) \in S$ , where  $y_{sim} \neq y_{dis}$ . Here, *sim* means  $x_{sim}$  and  $\bar{x}$  are similar because they share some attributes in the semantic feature space, e.g. a cat and a dog both have four legs and fur. *dis* means  $x_{dis}$  and  $\bar{x}$  are not similar. Mathematically,  $f_\theta$  trained in this way should make  $f_\theta(\bar{x})$  close to  $f_\theta(x_{sim})$  and distant to  $f_\theta(x_{dis})$  in a projected feature space, even without the label information from  $T$ . The illustration is presented in Fig. 1(c). We call this sampling mechanism reinforced sample selection (RSS). Since we output one sampling policy at once, RSS is actually a single-step Markov Decision Process [25].

The policy network is parameterized with  $\psi$ , denoted as  $h_\psi$ . We have  $h_\psi: \mathbb{R}^D \rightarrow \mathbb{R}^G$ , where  $G$  is the number of disjoint subsets of  $S$ . An intuitive design is to make sampling decision for each category independently, which can be implemented by  $G$  independent Bernoulli distributions. The support set is



then sampled from the selected categories. However, there are two problems with this design: (1) The number of possible combinations of the selected categories is huge for large  $G$  (for  $G = 10$ , we have  $2^{10} > 10^3$ ); (2) For each combination, there is a large variety which is uncontrollable. Here, constrained by the computational power, we simplify the problem by making the subsets mutually exclusive. Ideally,  $G = K_S$ , but considering computational complexity when  $K_S \gg K_T$ , in practice, we can utilize the side information (e.g. superclass), or clustering to  $G$  groups through a preprocessing step [31]. For  $\bar{\mathbf{x}} \in \{\bar{x}_j\}$ ,  $h_\psi(\bar{\mathbf{x}})$  is a  $G$  elements vector. Let  $g \in \{1, \dots, G\}$ , we define

$$p_\psi(y = g|\bar{\mathbf{x}}) = \frac{\exp(h_\psi(\bar{\mathbf{x}})[g])}{\sum_{g'=1}^G \exp(h_\psi(\bar{\mathbf{x}})[g'])}, \quad (8)$$

where  $[n]$  represents the  $n$ th element of a vector. We will decide whether or not to sample from group  $g$  based on a multinomial distribution with probabilities  $\{p_\psi(y = g|\bar{\mathbf{x}}) \mid \forall g \in \{1, \dots, G\}\}$ , the sampling policy is denoted as  $\Omega_\psi(\bar{\mathbf{x}})$ .

Another key component in reinforcement learning is setting the proper reward. With Euclidean distance defined on  $f_\theta$ , the optimization objective of  $f_\theta$  can be formulated as

$$\min \|f_\theta(\bar{\mathbf{x}}) - f_\theta(x_{sim})\|^2, \max \|f_\theta(\bar{\mathbf{x}}) - f_\theta(x_{dis})\|^2. \quad (9)$$

This can be further generalized as a deep DML problem with proper constraints [31]. However, the set of *sim* and the set of *dis* are not defined in most situations, and we can not solve the problem directly. Alternatively, we utilize the one-shot classifier. In an episode of a  $K$ -way one-shot learning task, we select the subset of  $S$  according to  $\Omega_\psi(\bar{\mathbf{x}})$  before sampling the support set and query image. After  $\theta$  and  $\phi$  are updated as in Algorithm 1, if the one-shot classifier correctly predicts the class label for the query image, then we replace the query image with the target image. We perform a one-shot classification with the original support set and updated query image. Note, the label of the query image is still the original label since we do not have the label for the target image. We want to see if the target image can confuse the one-shot classifier. The one-shot classifier is based on nearest neighbor search. If the target query image can be correctly classified, the target image is “close” to the corresponding image in the projected feature space. The reward is defined as

$$R(\Omega_\psi(\bar{\mathbf{x}})) = \begin{cases} 1 & \text{if correct,} \\ -\gamma & \text{otherwise.} \end{cases} \quad (10)$$

where  $\gamma$  is a small positive number. Since  $K_S \gg K_T$ , the reward will be sparse. In practice, given a support set, we choose to accumulate the reward by repeating the sampling operation for all the possible classes of query images. In other words, after the support set is sampled, we sample the query images for all  $K$  classes and for each class, we replace the query image with the target image to perform a one-shot classification. The reward of each query class is added up to calculate the total reward for the sampling action.

---

**Algorithm 2.** Training an episode of adversarial domain adaption with reinforced sample selection for a  $K$  way one-shot learning task, where  $K \leq K_S$  and  $K \leq K_D$ . The settings and notations are the same as Algorithm 1.

---

**Input** :  $S, T, \theta, \phi, \psi$   
**Output:**  $\theta, \phi, \psi$   
 $Query \leftarrow \{\}$   
 $Target \leftarrow \text{sample}(\{\bar{x}_j\}, 1)$   
Sample  $Support$  from  $S$  according to  $\Omega_\psi(Target)$   
**for**  $q \in \{1, \dots, K\}$  **do**  
|  $query \leftarrow \text{sampleQuery}(S, support)$ ; Add  $query$  to  $Query$   
**end**  
Calculate  $J_\phi$  with  $Support, Query$  and  $Target$  by (4)  
Update  $\phi$  by minimizing  $J_\phi$   
Calculate  $J_{cls}$  with  $Support$  and  $Query$  by (5)  
Calculate  $J_{adv}$  with  $Target$  by (6)  
Update  $\theta$  by minimizing  $J_\theta$   
Calculate  $J_{pn}$  with  $Support$  and  $Target$  by (11)  
Update  $\psi$  by minimizing  $J_{pn}$

---

The policy network is trained to maximize the expected reward  $\mathbb{E}_{\Omega_\psi}[R]$ . We define the loss for policy network as the negative expected reward

$$J_{pn} = -\mathbb{E}_{\Omega_\psi}[R(\Omega_\psi(\bar{\mathbf{x}}))]. \tag{11}$$

The  $J_{pn}$  or expected reward can be optimized by policy gradient, based on the REINFORCE rule [30]. The expected gradient is

$$\frac{\partial}{\partial \psi} J_{pn} = -\mathbb{E}_{\Omega_\psi}[R(\Omega_\psi(\bar{\mathbf{x}})) \frac{\partial}{\partial \psi} \log(p(\Omega_\psi(\bar{\mathbf{x}})))] \tag{12}$$

where  $\log(p(\Omega_\psi(\bar{\mathbf{x}})))$  means the log probability of sampled policy  $\Omega_\psi$  when the target image is  $\bar{\mathbf{x}}$ .  $\Omega_\psi$  is a multinomial distributions with  $G$  possible events, the probability mass function thus can be written as

$$p(\Omega_\psi(\bar{\mathbf{x}})) = \prod_{g=1}^G p_\psi(y = g|\bar{\mathbf{x}})^{\mathbf{1}_g}, \tag{13}$$

$\mathbf{1}_g$  is an indicator function indicates whether group  $g$  is selected by  $\Omega_\psi(\bar{\mathbf{x}})$ ,  $\sum_g \mathbf{1}_g = 1$ . RSS can be incorporated into Algorithm 1 with moderate modification. The updated algorithm is illustrated in Algorithm 2.

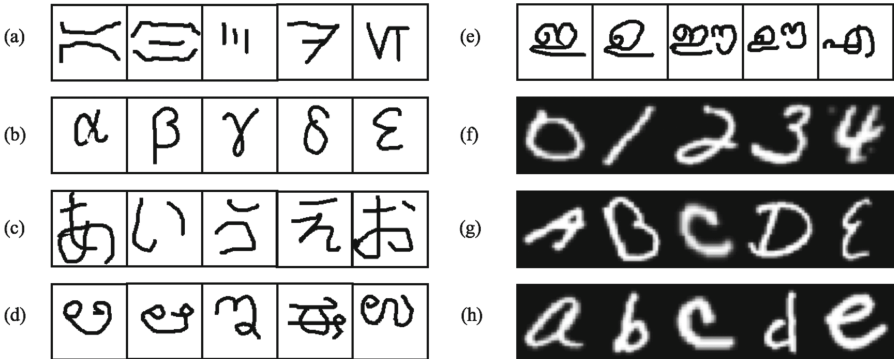
It is worth noting that RSS can be linked to curriculum learning. Similar to a curriculum, the entry-level courses can give a student general information about the field of study, which is easy to learn. The advanced courses have narrower topics but provide more details, and they are difficult to learn. When  $h_\psi$  is randomly initialized, the sampling strategy, similar to random selection, can help  $f_\theta$  learn more general information. As the learning proceeds, the sampling

strategy learned by  $h_\psi$  can focus on certain category groups and extract more domain-specific features, thus achieving better domain adaption performance. Similar to trial and error of human learners, RSS updates  $\psi$  and adjusts the output policy iteratively. By focusing on more relevant data and neglecting noise, RSS can also be interpreted as a weighted sampling method. A large probability for certain category means there is a higher chance that the category is sampled. The policy network learns the *attention* to category.

## 4 Experiments

### 4.1 Basic Settings

**Dataset.** Hand-written character recognition has been used to evaluate the machine learning algorithms in many works [16, 21, 23, 24, 29]. We use Omniglot [17] as the source domain and EMNIST [5] as the target domain (Fig. 3). Omniglot contains 1623 different characters from 50 different languages. Each character is written by 20 different people. Each image has a resolution of  $105 \times 105$ . Because the characters of *Latin* is identical to the characters of *English*, we remove *Latin*. The modified Omniglot has 1597 classes. EMNIST consists of 10 digits, 26 English letter with both uppercase and lowercase. There are 62 classes in total. Each image has a resolution of  $28 \times 28$ . We randomly select 20, 50 and 100 examples for each class to make a balanced subset of EMNIST. All the images are resized to  $28 \times 28$ , same as [21, 24, 29].



**Fig. 3.** Examples for hand-written character. Omniglot: (a) Aurebesh (an invented *Star Wars* language); (b) Greek; (c) Japanese (Hiragana); (d) Kannada; (e) Malayalam. EMNIST: (f) digits; (g) English (upper-case); (h) English (lower-case).

**Implementation.** There is no previous work for domain adaption in one-shot learning. A naive baseline model is training one-shot classifier on the source domain and applying it directly to the target domain, which is the standard

transfer learning. We choose Matching Networks (MN) [29] as the backbone architecture. We use Adam optimizer [15] for all experiments. The learning rate is fixed to  $10^{-4}$  for the one-shot classifier, the domain discriminator and the policy network. We choose  $B_S = 1$  and  $B_T = 1$ . We stop the training for all experiments train the one-shot classifier for 100 epochs which each epoch consists of 2000 episodes. The experiments are implemented in TensorFlow framework [1] on a GTX Titan X GPU.

**Evaluation.** The evaluation follows the protocol of [29]. The evaluation metric is the standard mean accuracy. In the evaluation phase, the support set and the query image are randomly sampled 10000 times. It is worth mentioning that [24, 29] only report the accuracy as a single number. With the fixed checkpoints of the model, we repeat the evaluation 100 times to produce the mean and the standard deviation of the accuracy.

## 4.2 Adversarial Domain Adaption

$f_\theta$  is a CNN feature extractor with four identical modules. Same as MN, each module is a sequential operations of two  $3 \times 3$  convolutions with batch normalization [14] and ReLU, and one  $2 \times 2$  max-pooling. The number of filters for the four modules are all 64.  $f_\theta$  is followed by a metric-based non-parametric classifier defined in Eq. 1. Here  $d(a, b) = \cos(a, b)$ .  $g_\phi$  consists of 3 fully-connected layers with number of outputs 64, 64 and 2. For this experiment, the domains differ in the content (language and writing styles) and image quality (color and resolution). In Table 1, we present the results of transfer learning (TL) and adversarial domain adaption (ADA) with different values  $\lambda_{adv}$  and different number of examples for each category in the target domain for 5-way 1-shot learning. ADA consistently outperforms TL with a large margin while  $\lambda_{adv}$  is sensitive to the target data. We fix  $\lambda_{adv} = 10^{-3}$  for the following experiments. In addition to  $d(a, b) = \cos(a, b)$ , we try a different the distance functions  $d(a, b) = -\|a - b\|^2$  [24] and the results are presented in Table 2. The results imply that *euclidean* distance is more suitable than *cosine* distance not only in one-shot learning [24], but also in domain adaption. Table 3 shows the results of  $k$ -way 1-shot learning tasks. Not surprisingly, ADA still consistently outperforms TL.

## 4.3 Reinforced Sample Selection

Without losing the generality of the proposed method, we simulate the experiments for RSS in the task of hand-written character recognition. The simulated experiments can be easily extended to general object recognition.  $h_\psi$  has similar architecture as  $f_\theta$ , where the four identical modules are followed by one fully-connected layer with number of outputs  $G$ . In this experiment, we simulate the *ideal* situation, where the source domain can be split into two disjoint subset, i.e.  $G = 2$ . There are one *sim* set and one *dis* set, as discussed in Sect. 3.4. Considering the huge computational cost for large  $K_S$ , we shrink both

**Table 1.** Sensitivity of  $\lambda_{adv}$  of ADA in 5-Way 1-Shot Learning.  $n_T$  means the number of examples for each category in the target domain.

Model	$n_T = 20$	$n_T = 50$	$n_T = 100$
Random Guess	20	20	20
TL	42.35 $\pm$ 0.51	47.48 $\pm$ 0.53	47.35 $\pm$ 0.47
ADA ( $\lambda_{adv} = 1$ )	46.23 $\pm$ 1.58	49.80 $\pm$ 1.68	52.70 $\pm$ 1.51
ADA ( $\lambda_{adv} = 10^{-1}$ )	51.30 $\pm$ 1.77	47.55 $\pm$ 1.73	51.06 $\pm$ 1.56
ADA ( $\lambda_{adv} = 10^{-2}$ )	52.58 $\pm$ 1.55	47.54 $\pm$ 1.65	48.58 $\pm$ 1.62
ADA ( $\lambda_{adv} = 10^{-3}$ )	53.73 $\pm$ 1.59	48.46 $\pm$ 1.54	48.32 $\pm$ 1.34
ADA ( $\lambda_{adv} = 10^{-4}$ )	49.98 $\pm$ 1.32	52.96 $\pm$ 1.43	48.67 $\pm$ 1.54

**Table 2.** Sensitivity of  $d(\cdot, \cdot)$  in ADA in 5-Way 1-Shot Learning.

Model	acc(%) $\pm$ std (%)
TL ( <i>cosine</i> )	42.35 $\pm$ 0.51
ADA ( <i>cosine</i> )	53.73 $\pm$ 1.59
TL ( <i>euclidean</i> )	50.07 $\pm$ 0.45
ADA ( <i>euclidean</i> )	55.87 $\pm$ 1.53

**Table 3.** ADA in  $k$ -Way 1-Shot Learning.

Model	5-way	10-way	20-way
Random Guess	20	10	5
TL	42.35 $\pm$ 0.51	31.09 $\pm$ 0.45	21.62 $\pm$ 0.43
ADA	53.73 $\pm$ 1.59	35.22 $\pm$ 1.62	23.30 $\pm$ 1.29

the source domain and target domain. For the target domain, we only use the capital English characters of EMNIST, i.e.  $K_T = 26$ . For the source domain, we only select 16 languages with 596 characters in total, i.e.  $K_S = 596$ . The set of *sim* contains 256 characters from *Anglo-Saxon* (Futhorc), *Armenian*, *Asomtavruli* (Georgian), *Cyrillic*, *Greek*, *Hebrew*, *Latin*, and *Mkhedruli* (Georgian). Because we do not involve the lower-case characters of EMNIST, *Latin* is added back to the source domain. The set of *dis* contains 340 characters from *Balinese*, *Bengali*, *Grantha*, *Gujarati*, *Gurmukhi*, *Kannada*, *Malayalam*, *Oriya*. See Fig. 3 for an intuitive illustration of the *sim* set, the *dis* set and the target domain. The results are presented in Table 4. Intuitively, training on more training data usually leads to better performance on the test data in the supervised setting, where the training data and the test data have the same categories. However, it may not be true in unsupervised setting, as discussed in Sect. 3.4. In the 5-way 1-shot learning task, we get an opposite result that training on the subset

which is similar to the test data produce better results, in both TL and ADA. As discussed in Sect. 3.3, this may be caused by overgeneralization. The same phenomenon can also be observed in the ADA of 10-way 1-shot learning task. RSS can even achieve better results than ADA trained on *sim*. RSS tries to utilize the source domain maximally and achieve a balance between the generalization and discrimination.

**Table 4.** RSS in  $k$ -Way 1-Shot Learning.

Model	5-way	10-way
TL	$41.17 \pm 0.51$	$36.19 \pm 1.42$
TL ( <i>dis</i> )	$45.78 \pm 1.68$	$32.23 \pm 1.30$
TL ( <i>sim</i> )	$46.34 \pm 1.34$	$30.50 \pm 1.41$
ADA	$45.02 \pm 1.43$	$33.99 \pm 1.36$
ADA ( <i>dis</i> )	$43.11 \pm 1.51$	$30.35 \pm 1.49$
ADA ( <i>sim</i> )	$51.20 \pm 1.64$	$37.31 \pm 1.51$
RSS	$52.31 \pm 1.63$	$38.41 \pm 1.48$

## 5 Conclusions

In this paper, we study the problem of domain adaption in one-shot learning. We review and compare the recent studies in one-shot learning and adversarial domain adaption. We formulate the problem of domain adaption in metric-based one-shot image classification. We propose an adversarial framework and investigate the limitations of adversarial training. Motivated by human learning, we introduce a new sampling strategy called reinforced sample selection to improve the domain adaption performance. We acknowledge that the improvements can be made to the reinforcement learning setting and optimization procedure. Domain adaption in one-shot learning and using reinforcement learning in domain adaption are both underdeveloped. In this work, we have the first trial in this area based on the cognitive science concepts and use experiments to validate the proposed framework. In the future, we will work more on the theoretical analysis of domain adaption in one-shot learning.

## References

1. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, pp. 265–283. USENIX Association (2016)
2. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in Neural Information Processing Systems, pp. 137–144 (2007)

3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: *International Conference on Machine Learning*, pp. 41–48. ACM (2009)
4. Biederman, I.: Recognition-by-components: a theory of human image understanding. *Psychol. Rev.* **94**(2), 115 (1987)
5. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: Emnist: an extension of mnist to handwritten letters. arXiv preprint [arXiv:1702.05373](https://arxiv.org/abs/1702.05373) (2017)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: Imagenet: a large-scale hierarchical image database. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255 (2009)
7. Dong, N., Kampffmeyer, M., Liang, X., Wang, Z., Dai, W., Xing, E.: Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio. In: Frangi, A.F., Schnabel, J.A., Davatzikos, C., Alberola-López, C., Fichtinger, G. (eds.) *MICCAI 2018*. LNCS, vol. 11071, pp. 544–552. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-00934-2\\_61](https://doi.org/10.1007/978-3-030-00934-2_61)
8. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **88**(2), 303–338 (2010)
9. Ganin, Y., et al.: Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**(1), 2030–2096 (2016)
10. Gluck, M.A., Mercado, E., Myers, C.E.: *Learning and memory: From brain to behavior*
11. Goodfellow, I., et al.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, pp. 2672–2680 (2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778 (2016)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
14. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *International Conference on Machine Learning*, pp. 448–456 (2015)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: *International Conference on Learning Representations* (2015)
16. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: *International Conference on Machine Learning Deep Learning Workshop* (2015)
17. Lake, B., Salakhutdinov, R., Gross, J., Tenenbaum, J.: One shot learning of simple visual concepts. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 33 (2011)
18. Li, F.F., Fergus, R., Perona, P.: One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(4), 594–611 (2006)
19. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: *International Conference on Learning Representations* (2015)
20. Motiian, S., Jones, Q., Iranmanesh, S., Doretto, G.: Few-shot adversarial domain adaptation. In: *Advances in Neural Information Processing Systems*, pp. 6673–6683 (2017)
21. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: *International Conference on Learning Representations* (2017)
22. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*, pp. 91–99 (2015)

23. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: International Conference on Machine Learning, pp. 1842–1850 (2016)
24. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, pp. 4077–4087 (2017)
25. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. *IEEE Trans. Neural Netw.* **9**(5), 1054–1054 (1998)
26. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing Systems, pp. 1057–1063 (2000)
27. Thrun, S.: Lifelong learning algorithms. In: Thrun, S., Pratt, L. (eds.) *Learning to Learn*, pp. 181–209. Springer, Boston (1998). [https://doi.org/10.1007/978-1-4615-5529-2\\_8](https://doi.org/10.1007/978-1-4615-5529-2_8)
28. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2962–2971. IEEE (2017)
29. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems, pp. 3630–3638 (2016)
30. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992)
31. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: Advances in Neural Information Processing Systems, pp. 521–528 (2003)
32. Zhang, K., Schölkopf, B., Muandet, K., Wang, Z.: Domain adaptation under target and conditional shift. In: International Conference on Machine Learning, pp. 819–827 (2013)