Domain Adaption in One-Shot Learning

Nanqing Dong[1,2] and Eric P. Xing[2]

[1] Cornell University, Ithaca, NY 14850, USA
nd367@cornell.edu
[2] Petuum, Inc., Pittsburgh, PA 15217, USA
eric.xing@petuum.com

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 distribution. 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.

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 [11], object detection [21] and semantic segmentation [18]. When modern visual recognition systems can benefit from large image datasets like ImageNet [5] and PASCAL VOC [6], 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 [3]. 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. [7] proposed a complex framework with strong probabilistic hypothesis using generative object category model and variational Bayesian expectation maximization (VBEM). Another approach is *meta-learning* [26]. Santoro et al. [22] attacked the problem by learning to memorize

unseen classes with a Memory Augmented Neural Network (MANN). Ravi and Larochelle [20] utilized a Long Short-Term Memory network (LSTM) [12] 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 [28, 23]. 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 [31, 8]. 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. We demonstrate that, in one-shot learning, the proposed method can achieve better results than previous domain adaption models. Motivated by the behavior of human learners, we propose to use a policy gradient method [29, 24, 25] 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 the proposed sampling strategy is linked to *distance metric learning* (DML) [30] and *curriculum learning* [2]. The concept is illustrated in Figure 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.

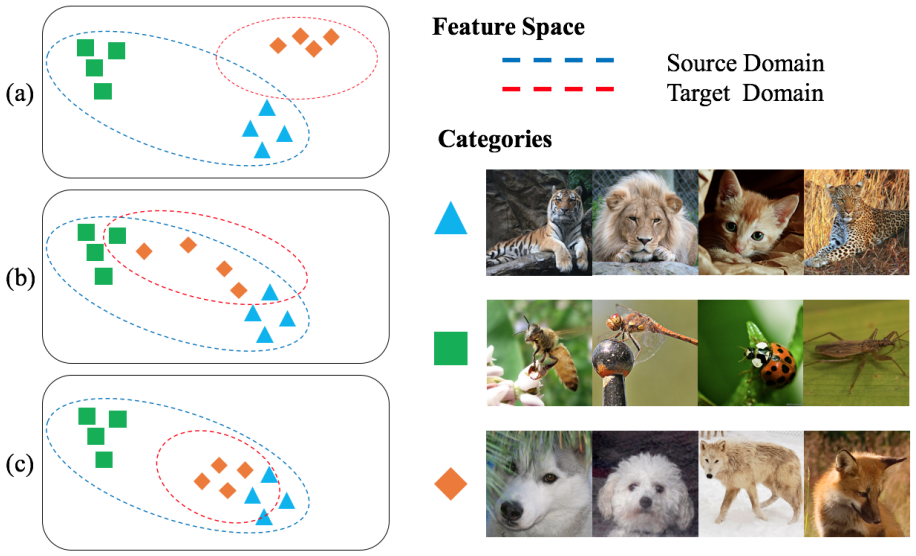

Fig. 1: Illustration of the motivation. Examples are embedded to certain feature spaces under three situations. (a) No domain adaption. (b) Domain adaption with random sample selection. (c) Domain adaption with reinforced sample selection.

2 Related Work

Many works [15, 22, 28, 20, 23] 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 [7] and the meta-learning approach [22, 20] in one-shot learning, recently proposed metric-based methods [28, 23] 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) [10]. 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. One problem for DANN is that the domain discriminator converges quickly, which can cause the gradient to vanish [27]. Another unsupervised domain adaption method is adversarial discriminative domain adaption (ADDA) [27]. ADDA ad-

dresses this problem by using independent feature extractors for each domain. The problem with this method is that the performance on target domain is highly dependent on the predictor trained on source domain. The stage-wise training means the training is split into different stages. In each stage, there is a different optimization objective. 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) [19], 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 Section 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 Figure 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

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

, where x_i represents an example from S and $y_i \in Y_S$ with $Y_S = \{1, ..., K_S\}$ 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

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

, where $\bar{x}_j$ represents an example from D and $\bar{y}_j \in Y_T$ with $Y_T = \{K_S + 1, ..., K_S + K_T\}$. 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 [15, 22, 28, 20, 23]. Let f_θ denotes an embedding function with parameters θ. f_θ embeds the input to a M-dimensional

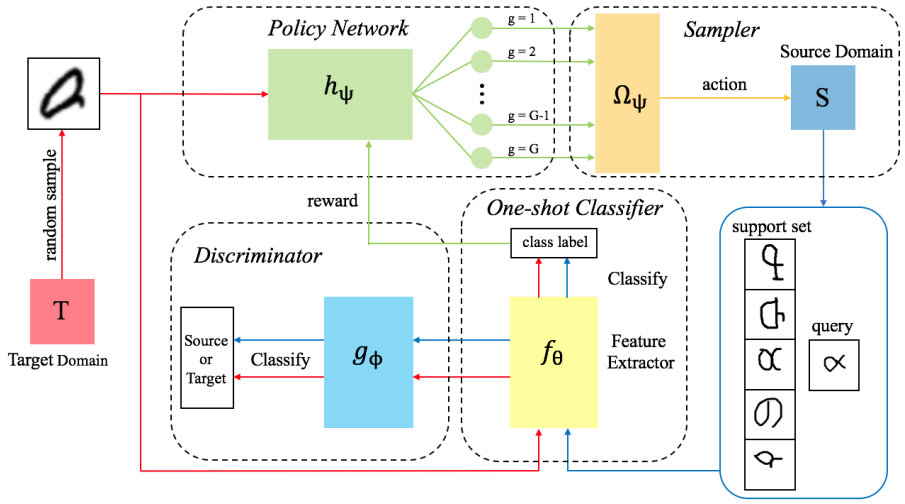

Fig. 2: Illustration of the model architecture. The figure depicts the data flow in the training phase. At the beginning of an episode, a random sample from the target domain goes through the feature extractor and discriminator for the first pass. Then policy network receives the sample and outputs a 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 Section 3.1, 3.2 and 3.4.

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

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

. 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_θ on S and use Equation 1 to predict labels for $\{\bar{x}_{t+1}, ..., \bar{x}_{N_T}\}$ based on $\{(\bar{x}_1, \bar{y}_1), ..., (\bar{x}_t, \bar{y}_t)\}$. f_θ 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 θ, 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 [27, 19]. 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 [23], 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 [1]. As in [10], we introduce a discriminator which is a parametric function of g_ϕ. g_ϕ takes the embedded features as input and outputs a probability score for the input comes from source domain, $g_\phi : \mathbb{R}^M \to \mathbb{R}$. The discriminator is then a binary classifier,

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

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

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

Given fixed f_θ, g_ϕ 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

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

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_ϕ, f_θ 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

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

, 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_θ 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 [10], 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 [28, 23]. All samples are sampled uniformly without replacement.

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

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

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

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

$$J_\theta = J_{cls} + \lambda_{adv} J_{adv} \tag{7}$$

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

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 [9]. 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 [1, 10]. 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 [27, 19] 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_θ 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 $\bar{\mathbf{x}}$ 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 $\bar{\mathbf{x}}$ 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 $\bar{\mathbf{x}}$ are not similar. Mathematically, f_θ trained in this way should make $f_\theta(\bar{\mathbf{x}})$ 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~~ The illustration is presented in Figure 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 [24].

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

possible combinations is huge for large G (for $G = 10$, we have $2^{10} > 10^3$) and the improvement of the performance is limited. Here, 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 [30]. More details about category grouping are explained in Section 4.3. For $\bar{\mathbf{x}} \in \{\bar{x}_j\}$, $h_\psi(\bar{\mathbf{x}})$ is a G elements vector. Let $g \in \{1, ..., G\}$, we define

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

, where $[n]$ represents the nth element of a vector. We will decide whether or not to sample from group g based on a multinomial distribution with probabilities $\{p_\psi(y = g|\bar{\mathbf{x}})| \ \forall \ g \in \{1, ..., G\}\}$, 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_θ, the optimization objective of f_θ can be formulated as

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

. This can be further generalized as a deep DML problem with proper constraints [30]. 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 θ and ϕ 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

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

, where γ 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.

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

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

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, θ, ϕ, ψ
Output: θ, ϕ, ψ
$Query \leftarrow \{\}$
$Target \leftarrow \text{sample}(\{\bar{x}_j\}, 1)$
Sample $Support$ from S accourding to $\Omega_\psi(Target)$
for $q \in \{1, ..., K\}$ **do**
  $query \leftarrow \text{sampleQuery}(S, support)$; Add $query$ to $Query$
end
Calculate J_ϕ with $Support$, $Query$ and $Target$ by (4)
Update ϕ by minimizing J_ϕ
Calculate J_{cls} with $Support$ and $Query$ by (5)
Calculate $J_{adv}$with $Target$ by (6)
Update θ by minimizing J_θ
Calculate J_{pn} with $Support$ and $Target$ by (11)
Update ψ by by minimizing J_{pn}

. The J_{pn} or expected reward can be optimized by policy gradient, based on the REINFORCE rule [29]. 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 Ω_ψ when the target image is $\bar{\mathbf{x}}$. Ω_ψ is a multinominal 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_ψ is randomly initialized, the sampling strategy, similar to random selection, can help f_θ learn more general information. As the learning proceeds, the sampling strategy learned by h_ψ 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 ψ 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

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. We use ADDA [27] as another baseline because FADA can not be adapted to one-shot classifier. We choose Matching Networks (MN) [28] and Prototypical Networks (PN) [23] as our backbone metric-based models. We use Adam optimizer [14] for all experiments. The learning rate is 0.0001 for the one-shot classifier and policy network, and is 0.00001 for the domain discriminator. We choose $B_S = 1$ and $B_T = 1$. Before we train the domain discriminator and policy network, we train the one-shot classifier to converge. The one-shot classification result is reported as a mean accuracy over 1000 episode on the target domain. All the experiments were implemented in TensorFlow framework on a GTX Titan X GPU.

Hand-written character recognition has been used to evaluate the machine learning algorithms in many works [15, 22, 28, 20, 23]. We use Omniglot [17] as the source domain and EMNIST [4] as the target domain (Figure 3). Omniglot contains 1623 different characters from 50 different languages. Each character is written by 20 different people. Each image has a resolution of 105×105. 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 28×28. The characters in Omniglot have fixed orientation, while the characters of EMNIST are randomly rotated. We randomly select 20 examples for each class to make a balanced subset of EMNIST. All the images are resized to 28×28, same as [28, 20, 23]. Omniglot and EMNIST are used as the main benchmark datasets in this paper.

4.2 Adversarial Domain Adaption

f_θ is a CNN feature extractor with four identical modules. Each module is a sequential operations of two 3×3 convolutions with batch normalization [13] and ReLU, and one 2×2 max-pooling. The number of filters for the four modules are 64, 128, 256 and 512. We use a deeper architecture to learn the language knowledge, more than just the strokes [28, 23]. f_θ is followed by a metric-based non-parametric classifier defined in Equation 1. Here $d(a, b) = cos(a, b)$ for MN and $d(a, b) = -||a - b||^2$ for PN. g_ϕ consists of 3 fully-connected layers with number of outputs 512, 512 and 2. h_ψ also has 4 modules while the number of filters are all 16. The modules are followed by one fully-connected layer.

For this experiment, the domains differ in the content (language and writing styles) and image quality (color). We present the results of 5-way one-shot learning and 20-way one-shot learning in Table 1. We also invert the color of images

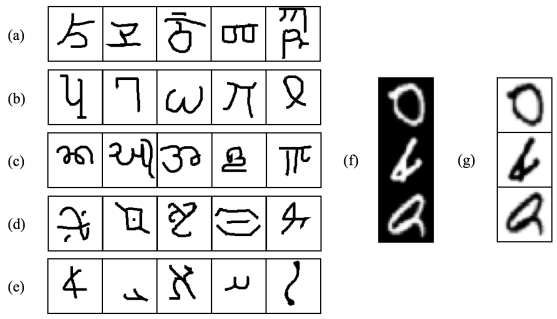

Fig. 3: Examples for hand-written character. Omniglot: (a) East Asian (b) European and (c) South and Southeast Asian (d) Fictitious and Magical (e) West Asian. EMNIST: (f) Original (g) Color Inverted.

model	5-way Acc	20-way Acc
MN	37.6%	16.7%
MN + ADDA*	46.2%	28.1%
MN + ADA (ours)	59.7%	36.4%
PN	46.8%	22.4%
PN + ADDA*	39.8%	26.4%
PN + ADA (ours)	56.5%	38.7%
MN + inv	37.2%	16.6%
MN + ADDA* + inv	44.1%	28.8%
MN + ADA (ours) + inv	61.0%	38.7%

Table 1: Comparison for Adversarial Domain Adaption in One-shot Learning. $*$ We reimplement ADDA [27] to have similar network architecture as ADA.

from EMNIST to isolate the effect from image quality (See Figure 3 (g)). It can be shown that our method consistently outperforms ADDA by a large margin. We do observe the overgeneralization in the training of ADDA described in Section 3.3. We conclude that ADDA suffers from serious overfitting in the source domain. Besides, ADDA nearly double the number of parameters which cost more memory space. We also note that the ADA is not as sensitive to the image quality as expected since the increase is minor when the color is inverted. The ADA does learn the knowledge behind the language. We decide to use the color-inverted EMNIST in the following experiments.

4.3 Reinforced Sample Selection

In this experiment, we want to examine both the effect and efficiency of RSS. The default h_ψ consists of 4 modules. Each module is a sequential operations of two 3×3 convolutions followed by batch normalization [13], ReLU and 2×2 max-pooling. The number of filters are all 32 for each modules. The modules are

model (G)	5-way Acc
MN + ADA (2)	57.5%
MN + ADA + RSS (2)	59.1%
MN + ADA (5)	46.1%
MN + ADA + RSS (5)	54.1%

Table 2: Results of RSS with different G (number of category groups).

model (η)	5-way Acc
MN + ADA + RSS (2.14%)	59.1%
MN + ADA + RSS (4.81%)	61.2%
MN + ADA + RSS (8.53%)	61.1%

Table 3: Results of RSS with different η (computational cost).

followed by one fully-connected layer with number of outputs G. Considering the huge computational cost for large K_S. We shrink both the source domain and target domain. For target domain, we only use the digits of EMNIST. For $G = 2$, we choose *Greek* and *Aurebesh* as 2 category groups. Aurebesh comes from the movie *Star Wars*. See the 4th image of Figure 3 (d) for an example. For $G = 5$, we choose *Greek*, *Aurebesh*, *Japanese*(*hiragana*), *Hebrew* and *Kannada* as 5 category groups. The results are presented in Table 2. With more irrelevant category groups, the performance of ADA decreases while RSS boosts the performance.

Because h_ψ brings more parameters, we also care about the computational cost for RSS. We define this cost as $\eta = \frac{|\psi|}{|\theta|+|\phi|}$, here $|\cdot|$ denotes the cardinality of the parameter set. For the default h_ψ and MN-based ADA model, we have $\eta = 2.14\%$. We also use more complex policy networks, where the number of filters are all 96 and 128. The η are 4.81% and 8.53%. The results are presented in Table 3 for $G = 2$. We conclude that with more parameters in h_ψ, there is a potential that the performance can be further improved. Because we use multinomial distribution instead of G independent Bernoulli distributions, we actually miss some possible combinations in the support set. To mitigate this problem, in the training phase, we alternatively use both Algorithm 1 and 2 decided by a Bernoulli sampler. The probability for the sampler to choose Algorithm 2 is the training accuracy for the source domain in last episode. The training accuracy for source domain reflects the generalization performance. When the generalization performance is good, the learner should focus more on domain-specific discrimination. We leave the G independent Bernoulli distributions design for future discussion.

4.4 Complex Settings

General object recognition is more challenging than the hand-written character recognition. The images took from real life usually have more background information. Besides, the objects in real life images are more complicated than hand-written characters, in both the number of categories and the complexity of features. We use CIFAR100 [16] as the source domain and ImageNet [5] as the target domain. CIFAR100 has 100 classes and 20 superclasses. Each image has a fixed resolution of 32×32. The original ImageNet has 1000 classes and more than 1 million images with different image size. We use the tiny version

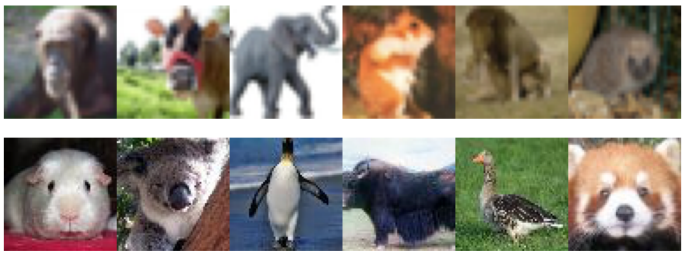

Fig. 4: Examples for animal classes. First Row: CIFAR100. Second Row: ImageNet.

ImageNet from Stanford University, where each image has a fixed resolution of 64×64. We select 20 animal classes (Figure 4) to simulate the target domain. For each class from CIFAR100 and ImageNet, we randomly select 100 images. All the images are resized to 32×32.

f_θ is a ResNet50 [11] without the last 1×1 convolutional layer. g_ϕ has the similar architecture as Section 4.2. The first two fully-connected layers both have 2048 number of outputs. h_ψ has five same modules as defined in Section 4.3. Image classification tasks on CIFAR100 and ImageNet require extensive computational power and long training time. Again, we limit the search space by merging superclasses into 2 groups. For CIFAR100, we have two groups, which are animals and non-animals. In the training phase, we adopt the same mixed optimization strategy as defined in Section 4.3 for a trade-off between generalization and discrimination. See Table 4 for the results.

model	**5-way Acc**
Random Guess	20.0%
MN	23.2%
MN + ADDA* [27]	23.5%
MN + ADA (ours)	24.7%
MN + ADA + RSS (2) (ours)	25.6%

Table 4: Results of ADA with RSS in complex settings.
$*$ We reimplement ADDA [27] to have similar network architecture as ADA.

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. 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)
2. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: International Conference on Machine Learning. pp. 41–48. ACM (2009)
3. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychological Review 94(2), 115 (1987)
4. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: Emnist: an extension of mnist to handwritten letters. arXiv preprint arXiv:1702.05373 (2017)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88(2), 303–338 (2010)
7. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 594–611 (2006)
8. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research 17(1), 2096–2030 (2016)
9. Gluck, M.A., Mercado, E., Myers, C.E.: Learning and memory: From brain to behavior
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2672–2680 (2014)
11. 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)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)
13. 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)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
15. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: International Conference on Machine Learning *Deep Learning Workshop* (2015)

16. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech Report (2009)
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. Liang-Chieh, 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)
19. Motiian, S., Jones, Q., Iranmanesh, S., Doretto, G.: Few-shot adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 6673–6683 (2017)
20. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: International Conference on Learning Representations (2017)
21. 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)
22. 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)
23. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems (2017)
24. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. IEEE Transactions on Neural Networks 9(5), 1054–1054 (1998)
25. 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)
26. Thrun, S.: Lifelong learning algorithms. In: Learning to learn, pp. 181–209. Springer (1998)
27. 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)
28. 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)
29. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning 8, 229–256 (1992)
30. 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)
31. 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)