Open Set Domain Adaptation by Backpropagation

Kuniaki Saito[1][0000−0001−9446−5068], Shohei Yamamoto[1], Yoshitaka Ushiku[1], and Tatsuya Harada[1,2]

[1] The University of Tokyo
{ksaito,yamamoto,ushiku,harada}@mi.t.u-tokyo.ac.jp
[2] RIKEN

Abstract. Numerous algorithms have been proposed for transferring knowledge from a label-rich domain (source) to a label-scarce domain (target). Most of them are proposed for closed-set scenario, where the source and the target domain completely share the class of their samples. However, in practice, a target domain can contain samples of classes that are not shared by the source domain. We call such classes the "unknown class" and algorithms that work well in the open set situation are very practical. However, most existing distribution matching methods for domain adaptation do not work well in this setting because unknown target samples should not be aligned with the source. In this paper, we propose a method for an open set domain adaptation scenario, which utilizes adversarial training. This approach allows to extract features that separate unknown target from known target samples. During training, we assign two options to the feature generator: aligning target samples with source known ones or rejecting them as unknown target ones. Our method was extensively evaluated and outperformed other methods with a large margin in most settings.

Keywords: Domain Adaptation, Open Set Recognition, Adversarial Learning

1 Introduction

Deep neural networks have demonstrated significant performance on many image recognition tasks [1]. One of the main problems of such methods is that basically, they cannot recognize samples as unknown, whose class is absent during training. We call such a class as an "unknown class" and the categories provided during training is referred to as the "known class." If these samples can be recognized as unknown, we can arrange noisy datasets and pick out the samples of interest from them. Moreover, if robots working in the real-world can detect unknown objects and ask annotators to give labels to them, these robots will be able to easily expand their knowledge. Therefore, the open set recognition is a very important problem.

In domain adaptation, we aim to train a classifier from a label-rich domain (source domain) and apply it to a label-scarce domain (target domain). Samples

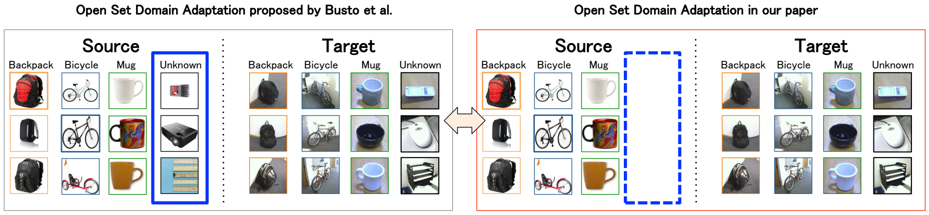


Fig. 1. A comparison between existing open set domain adaptation setting and our setting. **Left**: Existing setting of open set domain adaptation [2]. It is assumed that access is granted to the unknown source samples although the class of unknown source does not overlap with that of unknown target. **Right**: Our setting. We do not assume the accessibility to the unknown samples in the source domain. We propose a method that can be applied even when such samples are absent.

in different domains have diverse characteristics which degrade the performance of a classifier trained in a different domain. Most works on domain adaptation assume that samples in the target domain necessarily belong to the class of the source domain. However, this assumption is not realistic. Consider the setting of an unsupervised domain adaptation, where only unlabeled target samples are provided. We cannot know that the target samples necessarily belong to the class of the source domain because they are not given labels. Therefore, open set recognition algorithm is also required in domain adaptation. For this problem, the task called open set domain adaptation was recently proposed [2] where the target domain contains samples that do not belong to the class in the source domain as shown in the left of Fig. 1. The goal of the task is to classify unknown target samples as ”unknown” and to classify known target samples into correct known categories. They [2] utilized unknown source samples to classify unknown target samples as unknown. However, collecting unknown source samples is also expensive because we must collect diverse and many unknown source samples to obtain the concept of “unknown.” Then, in this paper, we present a more challenging open set domain adaptation (OSDA) that does not provide any unknown source samples, and we propose a method for it. That is, we propose a method where we have access to only known source samples and unlabeled target samples for open set domain adaptation as shown in the right of Fig. 1.

How can we solve the problem? We think that there are mainly two problems. First, in this situation, we do not have knowledge about which samples are the unknown samples. Thus, it seems difficult to delineate a boundary between known and unknown classes. The second problem is related to the domain's difference. Although we need to align target samples with source samples to reduce this domain's difference, unknown target samples cannot be aligned due to the absence of unknown samples in the source domain. The existing distribution matching method is aimed at matching the distribution of the target with that of the source. However, this method cannot be applied to our problem. In OSDA, we must reject unknown target samples without aligning them with the source.

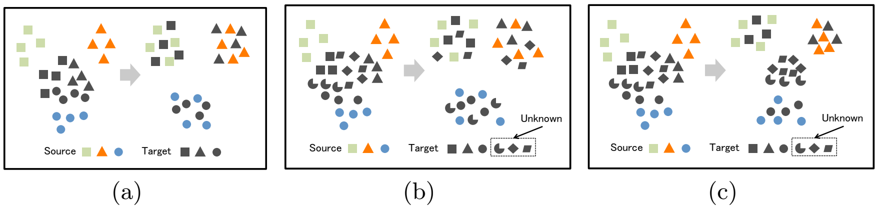


Fig. 2. (a): Closed set domain adaptation with distribution matching method. (b): Open set domain adaptation with distribution matching method. Unknown samples are aligned with known source samples. (c): Open set domain adaptation with our proposed method. Our method enables to learn features that can reject unknown target samples.

To solve the problems, we propose a new approach of adversarial learning that enables generator to separate target samples into known and unknown classes. A comparison with existing methods is shown in Fig. 2. Unlike the existing distribution alignment methods that only match the source and target distribution, our method facilitates the rejection of unknown target samples with high accuracy as well as the alignment of known target samples with known source samples. We assume that we have two players in our method, i.e., the feature generator and the classifier. The feature generator generates features from inputs, and the classifier takes the features and outputs $K + 1$ dimension probability, where K indicates the number of known classes. The $K + 1$ th dimension of output indicates the probability for the unknown class. The classifier is trained to make a boundary between source and target samples whereas the feature generator is trained to make target samples far from the boundary. Specifically, we train the classifier to output probability t for unknown class, where $0 < t < 1$. We can build a decision boundary for unknown samples by weakly training a classifier to classify target samples as unknown. To deceive the classifier, the feature generator has two options to increase or to decrease the probability. As such, we assign two options to the feature generator: aligning them with samples in the source domain or rejecting them as unknown.

The contribution of our paper is as follows.

1. We present the open set domain adaptation where unknown source samples are not provided. The setting is more challenging than the existing setting.
2. We propose a new adversarial learning method for the problem. The method enables training of the feature generator to learn representations which can separate unknown target samples from known ones.
3. We evaluate our method on adaptation for digits and objects datasets and demonstrate its effectiveness. Additionally, the effectiveness of our method was demonstrated in standard open set recognition experiments where we are provided unlabeled unknown samples during training.

2 Related Work

In this section, we briefly introduce methods for domain adaptation and open set recognition.

2.1 Domain Adaptation

Domain adaptation for image recognition has attracted attention for transferring the knowledge between different domains and reducing the cost for annotating a large number of images in diverse domains. Benchmark datasets are released [3], and many methods for unsupervised domain adaptation and semi-supervised domain adaptation have been proposed [4–11]. As previously indicated, unsupervised and semi-supervised domain adaptation focus on the situation where different domains completely share the class of their samples, which may not be practical especially in unsupervised domain adaptation.

One of the effective methods for unsupervised domain adaptation are distribution matching based methods [4, 6, 12–14]. Each domain has unique characteristics of their features, which decrease the performance of classifiers trained on a different domain. Therefore, by matching the distributions of features between different domains, they aim to extract domain-invariantly discriminative features. This technique is widely used in training neural networks for domain adaptation tasks [4, 15]. The representative of the methods harnesses techniques used in Generative Adversarial Networks (GAN) [16]. GAN trains a classifier to judge whether input images are fake or real images whereas the image generator is trained to deceive it. In domain adaptation, similar to GAN, the classifier is trained to judge whether the features of the middle layers are from a target or a source domain whereas the feature generator is trained to deceive it. Variants of the method and extensions to the generative models for domain adaptation have been proposed [13, 17–20]. Maximum Mean Discrepancy (MMD) [21] is also a representative way to measure the distance between domains. The distance is utilized to train domain-invariantly effective neural networks, and its variants are proposed [6, 7, 22, 23].

The problem is that these methods do not assume that the target domain has categories that are not included in the source domain. The methods are not supposed to perform well on our open set domain adaptation scenario. This is because all target samples including unknown classes will be aligned with source samples. Therefore, this makes it difficult to detect unknown target samples.

In contrast, our method enables to categorize unknown target samples into unknown class, although we are not provided any labeled target unknown samples during training. We will compare our method with MMD and domain classifier based methods in experiments. We utilize the technique of distribution matching methods technique to achieve open set recognition. However, the main difference is that our method allows the feature generator to reject some target samples as outliers.

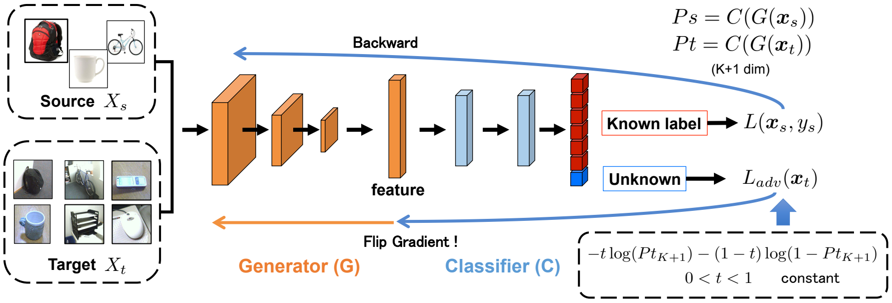

$Ps = C(G(\boldsymbol{x}_s))$
$Pt = C(G(\boldsymbol{x}_t))$
(K+1 dim)
Backward
Source X_s
Target X_t
feature
Known label
$L(\boldsymbol{x}_s, y_s)$
Unknown
$L_{adv}(\boldsymbol{x}_t)$
Flip Gradient !
Generator (G)
Classifier (C)
$-t\log(Pt_{K+1}) - (1-t)\log(1-Pt_{K+1})$
$0 < t < 1$ constant

Fig. 3. The proposed method for open set domain adaptation. The network is trained to correctly classify source samples. For target samples, the classifier is trained to output t for the probability of the unknown class whereas the generator is trained to deceive it.

2.2 Open Set Recognition

A wide variety of research has been conducted to reject outliers while correctly classifying inliers during testing. Multi-class open set SVM is proposed by [24]. They propose to reject unknown samples by training SVM that assign probabilistic decision scores. The aim is to reject unknown samples using a threshold probability value. In addition, method of harnessing deep neural networks for open set recognition was proposed [25]. They introduced OpenMax layer, which estimates the probability of an input being from an unknown class. Moreover, to give supervision of the unknown samples, a method to generate these samples was proposed [26]. The method utilizes GAN to generate unknown samples and use it to train neural networks, then combined it with OpenMax layer. In order to recognize unknown samples as unknown during testing, these methods defined a threshold value to reject unknown samples. Also, they do not assume that they can utilize unlabeled samples including known and unknown classes during training.

In our work, we propose a method that enables us to deal with the open set recognition problem in the setting of the domain adaptation. In this setting, the distribution of the known samples in the target domain is different from that of the samples in the source domain, which makes the task more difficult.

3 Method

First, we provide an overview of our method, then we explain the actual training procedure and provide an analysis of our method by comparing it with existing open set recognition algorithm. The overview is shown in Fig. 3.

3.1 Problem Setting and Overall Idea

We assume that a labeled source image $\boldsymbol{x}_s$ and a corresponding label y_s drawn from a set of labeled source images $\{X_s, Y_s\}$ are available, as well as an unlabeled target image $\boldsymbol{x}_t$ drawn from unlabeled target images X_t. The source images are drawn only from known classes whereas target images can be drawn from unknown class. In our method, we train a feature generation network G, which takes inputs $\boldsymbol{x}_s$ or $\boldsymbol{x}_t$, and a network C, which takes features from G and classifies them into $K+1$ classes, where the K denotes the number of known categories. Therefore, C outputs a $K+1$-dimensional vector of logits $\{l_1, l_2, l_3...l_{K+1}\}$ per one sample.

The logits are then converted to class probabilities by applying the softmax function. Namely, the probability of $\boldsymbol{x}$ being classified into class j is denoted by $p(y=j|\boldsymbol{x}) = \frac{\exp(l_j)}{\sum_{k=1}^{K+1}\exp(l_k)}$. $1 \sim$ K dimensions indicate the probability for the known classes whereas $K+1$ dimension indicates that for the unknown class. We use the notation $p(\boldsymbol{y}|\boldsymbol{x})$ to denote the $K+1$-dimensional probabilistic output for input $\boldsymbol{x}$.

Our goal is to correctly categorize known target samples into corresponding known class and recognize unknown target samples as unknown. We have to construct a decision boundary for the unknown class, although we are not given any information about the class. Therefore, we propose to make a pseudo decision boundary for unknown class by weakly training a classifier to recognize target samples as unknown class. Then, we train a feature generator to deceive the classifier. The important thing is that feature generator has to separate unknown target samples from known target samples. If we train a classifier to output $p(y=K+1|\boldsymbol{x}_t) = 1.0$ and train the generator to deceive it, then ultimate objective of the generator is to completely match the distribution of the target with that of the source. Therefore, the generator will only try to decrease the value of the probability for unknown class. This method is used for training Generative Adversarial Networks for semi-supervised learning [27] and should be useful for unsupervised domain adaptation. However, this method cannot be directly applied to separate unknown samples from known samples.

Then, to solve the difficulty, we propose to train the classifier to output $p(y=K+1|\boldsymbol{x}_t) = t$, where $0 < t < 1$. We train the generator to deceive the classifier. That is, the objective of the generator is to maximize the error of the classifier. In order to increase the error, the generator can choose to increase the value of the probability for an unknown class, which means that the sample is rejected as unknown. For example, consider when t is set as a very small value, it should be easier for generator to increase the probability for an unknown class than to decrease it to maximize the error of the classifier. Similarly, it can choose to decrease it to make $p(y=K+1|\boldsymbol{x}_t)$ lower than t, which means that the sample is aligned with source. In summary, the generator will be able to choose whether a target sample should be aligned with the source or should be rejected. In all our experiments, we set the value of t as 0.5. If t is larger than 0.5, the sample is necessarily recognized as unknown. Thus, we assume that this value can be a good boundary between known and unknown. In our experiment, we will analyze the behavior of our model when this value is varied.

Algorithm 1 Minibatch training of the proposed method.

```
for the number of training iterations do
    • Sample minibatch of $m$ source samples $\{\{\boldsymbol{x_s}, y_s\}^{(1)}, \ldots, \{\boldsymbol{x_s}, y_s\}^{(m)}\}$ from $\{X_s, Y_s\}$.
    • Sample minibatch of $m$ target samples $\{\boldsymbol{x_t}^{(1)}, \ldots, \boldsymbol{x_t}^{(m)}\}$ from $X_t$.
    Calculate $L_s(\boldsymbol{x}_s, y_s)$ by cross-entropy loss and $L_{adv}(\boldsymbol{x_t})$ following Eq. 3.
    Update the parameter of $G$ and $C$ following Eq. 4, Eq. 5. We used gradient reversal layer for this operation.
end for
```

3.2 Training Procedure

We begin by demonstrating how we trained the model with our method. First, we trained both the classifier and the generator to categorize source samples correctly. We use a standard cross-entropy loss for this purpose.

$$L_s(\boldsymbol{x}_s, y_s) = -\log(p(y = y_s|\boldsymbol{x}_s)) \tag{1}$$

$$p(y = y_s|\boldsymbol{x}_s) = (C \circ G(\boldsymbol{x}_s))_{y_s} \tag{2}$$

In order to train a classifier to make a boundary for an unknown sample, we propose to utilize a binary cross entropy loss.

$$L_{adv}(\boldsymbol{x}_t) = -t\log(p(y = K+1|\boldsymbol{x}_t)) - (1-t)\log(1 - p(y = K+1|\boldsymbol{x}_t)) \tag{3}$$

, where t is set as 0.5 in our experiment. The overall training objective is,

$$\min_C L_s(\boldsymbol{x}_s, y_s) + L_{adv}(\boldsymbol{x}_t) \tag{4}$$

$$\min_G L_s(\boldsymbol{x}_s, y_s) - L_{adv}(\boldsymbol{x}_t) \tag{5}$$

The classifier attempts to set the value of $p(y = K+1|\boldsymbol{x}_t)$ equal to t whereas the generator attempts to maximize the value of $L_{adv}(\boldsymbol{x}_t)$. Thus, it attempts to make the value of $p(y = K+1|\boldsymbol{x}_t)$ different from t. In order to efficiently calculate the gradient for $L_{adv}(\boldsymbol{x}_t)$, we utilize a gradient reversal layer proposed by [4]. The layer enables flipping of the sign of the gradient during the backward process. Therefore, we can update the parameters of the classifier and generator simultaneously. The algorithm is shown in Alg. 1.

3.3 Comparison with Existing Methods

We think that there are three major differences from existing methods. Since most existing methods do not have access to unknown samples during training, they cannot train feature extractors to learn features to reject them. In contrast, in our setting, unknown target samples are included in training samples. Under the condition, our method can train feature extractors to reject unknown samples. In addition, existing methods such as open set SVM reject unknown samples if the probability of any known class for a testing sample is not larger than the threshold value. The value is a pre-defined one and does not change across testing samples. However, with regard to our method, we can consider that the threshold value changes across samples because our model assigns different classification outputs to different samples. Thirdly, the feature extractor is informed of the pseudo decision boundary between known and unknown classes. Thus, feature extractors can recognize the distance between each target sample and the boundary for the unknown class. It attempts to make it far from the boundary. It makes representations such that the samples similar to the known source samples are aligned with known class whereas ones dissimilar to known source samples are separated from them.

4 Experiments

We conduct experiments on Office [3], VisDA [28] and digits datasets.

4.1 Implementation Detail

We trained the classifier and generator using the features obtained from AlexNet [1] and VGGNet [29] pre-trained on ImageNet [30]. In the experiments on both Office and VisDA dataset, we did not update the parameters of the pre-trained networks. We constructed fully-connected layers with 100 hidden units after the FC8 layers. Batch Normalization [31] and Leaky-ReLU layer were employed for stable training. We used momentum SGD with a learning rate 1.0×10^{-3}, where the momentum was set as 0.9. Other details are shown in our supplementary material due to a limit of space.

We implemented three baselines in the experiments. The first baseline is an open set SVM (OSVM) [24]. OSVM utilizes the threshold probability to recognize samples as unknown if the predicted probability is lower than the threshold for any class. We first trained CNN only using source samples, then, use it as a feature extractor. Features are extracted from the output of generator networks when using OSVM. OSVM does not require unknown samples during training. Therefore, we trained OSVM only using source samples and tested them on the target samples. The second one is a combination of Maximum Mean Discrepancy(MMD) [21] based training method for neural networks [6] and OSVM. MMD is used to match the distribution between different domains in unsupervised domain adaptation. For an open set recognition, we trained the networks with MMD and trained OSVM using the features obtained by the networks. A comparison with this baseline should indicate how our proposed method is different from existing distribution matching methods. The third one is a combination of a domain classifier based method, BP [4] and OSVM. BP is also a representative of a distribution matching method. As was done for MMD, we first trained BP and extracted features to train OSVM. We used the same network architecture to train the baseline models. The experiments were run a total of 3 times for each method, and the average score was reported. We report the standard deviation only in Table 2 because of the limit of space.

4.2 Experiments on Office

11 Class Classification Firstly, we evaluated our method using Office following the protocol proposed by [2]. The dataset consists of 31 classes, and 10 classes were selected as shared classes. The classes are also common in the Caltech dataset [8]. In alphabetical order, 21-31 classes are used as unknown samples in the target domain. The classes 11-20 are used as unknown samples in the source domain in [2]. However, we did not use it because our method does not require such samples. We have to correctly classify samples in the target domain into 10 shared classes or unknown class. In total, 11 class classification was performed. Accuracy averaged over all classes is denoted as OS in all Tables. OS $= \frac{1}{K+1}\sum_{k=1}^{K+1} Acc_k$, where K indicates number of known classes and $K+1$ th class is an unknown class. We also show the accuracy measured only on the known classes of the target domain (OS*). OS* $= \frac{1}{K}\sum_{k=1}^{K} Acc_k$. Following [2], we show the accuracy averaged over the classes in the OS and OS*. We also compared our method with a method proposed by [2]. Their method is developed for a situation where unknown samples in the source domain are available. However, they applied their method using OSVM when unknown source samples were absent. In order to better understand the performance of our method, we also show the results which utilized the unknown source samples during training. The values are cited from [2].

The results are shown in Table 1. Compared with the baseline methods, our method exhibits better performance in almost all scenarios. The accuracy of the OS is almost always better than that of OS*, which means that many known target samples are regarded as unknown. This is because OSVM is trained to detect outliers and is likely to classify target samples as unknown. When comparing the performance of OSVM and MMD+OSVM, we can see that the usage of MMD does not always boost the performance. The existence of unknown target samples seems to perturb the correct feature alignment. Visualizations of features are shown in our supplementary material.

Number of Unknown Samples and Accuracy We further investigate the accuracy when the number of target samples varies in the adaptation from DSLR to Amazon. We randomly chose unknown target samples from Amazon and varied the ratio of the unknown samples. The accuracy of OS is shown in Fig. 4(a). When the ratio changes, our method seems to perform well.

Value of t We observe the behavior of our model when the training signal, t in Eq. 3 is varied. As we mentioned in the method section, When t is equal to 1, the objective of the generator is to match the whole distribution of the target features with that of the source, which is exactly the same as an existing distribution matching method. Accordingly, the accuracy should degrade in this case. According to Fig. 5(b), as we increase the value of t, the accuracies of OS and OS* decrease and the overall accuracy increases. This result means

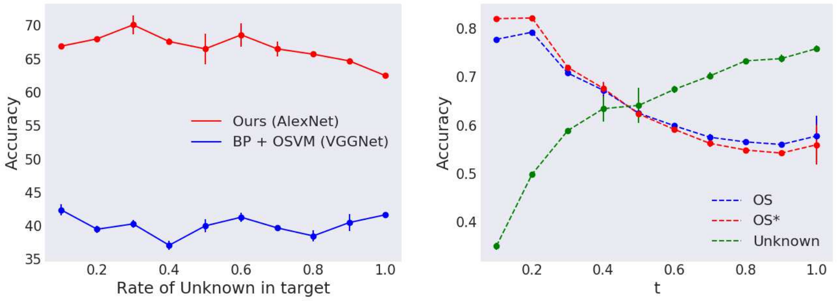
	Adaptation Scenario													
	A-D		A-W		D-A		D-W		W-A		W-D		AVG	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
Method w/ unknown classes in source domain (AlexNet)														
BP [4]	78.3	77.3	75.9	73.8	57.6	54.1	89.8	88.9	64.0	61.8	98.7	98.0	77.4	75.7
ATI-λ [2]	79.8	79.2	77.6	76.5	71.3	70.0	93.5	93.2	76.7	76.5	98.3	99.2	82.9	82.4
Method w/o unknown classes in source domain (AlexNet)														
OSVM	59.6	59.1	57.1	55.0	14.3	5.9	44.1	39.3	13.0	4.5	62.5	59.2	40.6	37.1
MMD + OSVM	47.8	44.3	41.5	36.2	9.9	0.9	34.4	28.4	11.5	2.7	62.0	58.5	34.5	28.5
BP+OSVM	40.8	35.6	31.0	24.3	10.4	1.5	33.6	27.3	11.5	2.7	49.7	44.8	29.5	22.7
ATI-λ[2] + OSVM	72.0	-	65.3	-	**66.4**	-	82.2	-	71.6	-	92.7	-	75.0	-
Ours	**76.6**	**76.4**	**74.9**	**74.3**	62.5	**62.3**	**94.4**	**94.6**	**81.4**	**81.2**	**96.8**	**96.9**	**81.1**	**80.9**
Method w/o unknown classes in source domain (VGGNet)														
OSVM	82.1	83.9	75.9	75.8	38.0	33.1	57.8	54.4	54.5	50.7	83.6	83.3	65.3	63.5
MMD + OSVM	84.4	**85.8**	75.6	75.7	41.3	35.9	61.9	58.7	50.1	45.6	84.3	83.4	66.3	64.2
BP+OSVM	83.1	84.7	76.3	76.1	41.6	36.5	61.1	57.7	53.7	49.9	82.9	82.0	66.4	64.5
Ours	**85.8**	**85.8**	**85.3**	**85.1**	**88.7**	**89.6**	**94.6**	**95.2**	**83.4**	**83.1**	**97.1**	**97.3**	**89.1**	**89.4**

Table 1. Accuracy (%) of each method in 10 shared class situation. A, D and W correspond to Amazon, DSLR and Webcam respectively.

that the model does not learn representations where unknown samples can be distinguished from known samples.

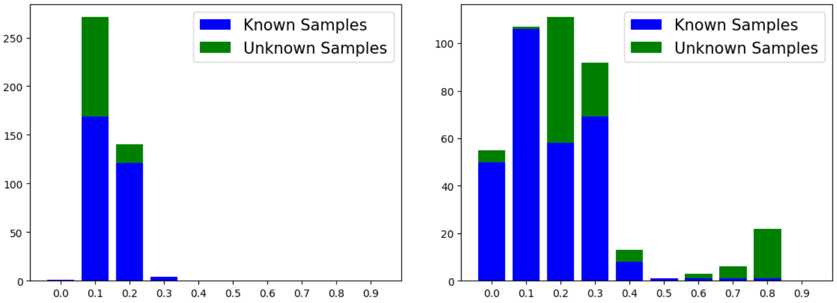
Probability for Unknown Class In Fig. 5(a)(b), frequency diagram of the probability for an unknown class is shown in the adaptation from Webcam to DSLR dataset. At the beginning of training, Fig. 5(a), the probability is low in most samples including the known and unknown samples. As shown in Fig. 5(b), many unknown samples have high probability for unknown class whereas many known samples have low probability for the class after training the model for 500 epochs. We can observe that unknown and known samples seem to be separated from the result.

21 Class Classification In addition, we observe the behavior of our method when the number of known classes increases. We add the samples of 10 classes which were not used in the previous setting. The 10 classes are the ones used as unknown samples in the source domain in [2]. In total, we conducted 21 class classification experiments in this setting. We also evaluate our method on VGG Network. With regard to other details of the experiment, we followed the setting of the previous experiment. The results are shown in Table 2. Compared to the baseline methods, the superiority of our method is clear. The usefulness of MMD and BP is not observed for this setting too. An examination of the result of adaptation from Amazon to Webcam (A-W) reveals that the accuracy of other methods is better than our approach based on OS* and OS. However, "ALL" of the measurements are inferior to our method. The value of "ALL" indicates the accuracy measured for all the samples without averaging over classes. Thus, the result means that existing methods are likely to recognize target samples as



(a) Ratio of unknown samples (b) Value of t and accuracy

Fig. 4. (a): The behavior of our method when we changed the ratio of unknown samples. As we increase the number of unknown target samples, the accuracy decreases. **(b)**: The change of accuracy with the change of the value t. The accuracy for unknown target samples is denoted as green line. As t increases, target samples are likely classified as "unknown". However, the entire accuracy OS and OS* decrease.



(a) Epoch 50 (b) Epoch 500

Fig. 5. (a)(b): Frequency diagram of the probability of target samples for unknown class in adaptation from Webcam to DSLR.

one of known classes in this setting. From the results, the effectiveness of our method is verified when the number of class increases.

4.3 Experiments on VisDA Dataset

We further evaluate our method on adaptation from synthetic images to real images. VisDA dataset [28] consists of 12 categories in total. The source domain images are collected by rendering 3D models whereas the target domain images consist of real images. We used the training split as the source domain and validation one as the target domain. We choose 6 categories (bicycle, bus, car, motorcycle, train and truck) from them and set other 6 categories as the unknown class (aeroplane, horse, knife, person, plant and skateboard). The training procedure of the networks is the same as that used for Office dataset.

	Adaptation Scenario								
	A-D			A-W			D-A		
	OS	OS*	ALL	OS	OS*	ALL	OS	OS*	ALL
OSVM	73.6±0.4	**75.8**±0.6	57.6	**72.0**±0.5	**74.1**±0.5	58.0	44.9±0.1	43.9±0.1	51.1
MMD + OSVM	72.1±0.9	73.9±1.0	57.8	69.1±0.8	71.2±0.9	54.9	29.8±0.6	26.5±0.6	50.3
BP + OSVM	70.4±0.2	72.1±0.3	57.1	70.9±0.5	72.9±0.4	57.6	30.9±0.2	27.6±0.2	51.3
Ours	**74.8**±0.5	74.6 ±0.5	**73.9**	66.8±3.5	66.1±3.7	**69.7**	**64.6**±1.2	**65.9**±4.9	**68.5**

	D-W			W-A			W-D			AVG		
	OS	OS*	ALL	OS	OS*	ALL	OS	OS*	ALL	OS	OS*	ALL
OSVM	63.1±1.1	61.9±1.2	69.9	34.0±0.9	31.8±1.3	48.3	82.9±2.3	82.9±1.7	84.2	61.8	61.7	61.5
MMD + OSVM	58.3±0.6	56.6±0.6	68.8	39.7±2.1	37.1±2.4	55.9	84.5±1.2	84.2±1.3	87.2	58.9	58.2	62.3
BP+OSVM	63.2±2.8	61.7±3.0	71.3	40.0±2.7	37.4±3.0	56.0	83.5±0.8	83.1±0.8	86.4	59.8	59.1	63.2
Ours	**83.1**±0.6	**82.5**±0.6	**84.9**	**65.9**±0.1	**65.3**±0.2	**69.0**	**92.8**±0.2	**93.3**±0.2	**90.3**	**74.7**	**74.6**	**76.1**

Table 2. Accuracy (%) of experiments on Office dataset in 20 shared class situation. We used VGG Network to obtain the results.

Method	bcycle	bus	car	mcycle	train	truck	unknwn	Avg	Avg knwn
AlexNet									
OSVM	4.8	45.0	44.2	43.5	59.0	10.5	57.4	37.8	34.5
OSVM+MMD	0.2	30.9	49.1	54.8	56.1	8.1	61.3	37.2	33.2
OSVM+BP	9.1	50.5	**53.9**	79.8	69.0	8.1	42.5	44.7	45.1
Ours	**48.0**	**67.4**	39.2	**80.2**	**69.4**	**24.9**	**80.3**	**58.5**	**54.8**
VGGNet									
OSVM	31.7	51.6	66.5	70.4	**88.5**	20.8	38.0	52.5	54.9
OSVM+MMD	39.0	50.1	64.2	79.9	86.6	16.3	44.8	54.4	56.0
OSVM+BP	31.8	56.6	**71.7**	77.4	87.0	22.3	41.9	55.5	57.8
Ours	**51.1**	**67.1**	42.8	**84.2**	81.8	**28.0**	**85.1**	**62.9**	**59.2**

Table 3. Accuracy (%) on VisDA dataset. The accuracy per class is shown.

The results are shown in Table 3. Our method outperformed the other methods in most cases. *Avg* indicates the accuracy averaged over all classes. *Avg known* indicates the accuracy averaged over only known classes. In both evaluation metrics, our method showed better performance, which means that our method is better both at matching distributions between known samples and rejecting unknown samples in open set domain adaptation setting. In this setting, the known classes and unknown class should have different characteristics because known classes are picked up from vehicles and unknown samples are from others. Thus, in our method, the accuracy for the unknown class is better than that for the known classes. We further show the examples of images in Table 4. Some of the known samples are recognized as unknown. As we can see from the three images, most of them contain multiple classes of objects or are hidden by other objects. Then, look at the second columns from the left. The images are categorized as motorcycle though they are unknown. The images of motorcycle often contain persons and the appearance of the person and horse have similar features to such images. In the third and fourth columns, we demonstrate the correctly classified known and unknown samples. If the most part of the image is occupied by the object of interest, the classification seems to be successful.

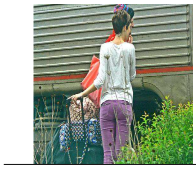
Ground Truth Class → Predicted Class			
Known → Unknown ×	**Unknown → Known ×**	**Known → Known √**	**Unknown → Unknown √**
Train → Unknown 	Unknown → Motorcycle 	Truck → Truck 	Unknown → Unknown
Motorcycle → Unknown 	Unknown → Motorcycle 	Bicycle → Bicycle 	Unknown → Unknown
Car → Unknown 	Unknown → Motorcycle 	Motorcycle → Motorcycle 	Unknown → Unknown

Table 4. Examples of recognition results on VisDA dataset.

4.4 Experiments on Digits Dataset

We also evaluate our method on digits dataset. We used SVHN [32],USPS [33] and MNIST for this experiment. In this experiment, we conducted 3 scenarios in total. Namely, adaptation from SVHN to MNIST, USPS to MNIST and MNIST to USPS. These are common scenarios in unsupervised domain adaptation. The numbers from 0 to 4 were set as known categories whereas the other numbers were set as unknown categories. In this experiment, we also compared our method with two baselines, OSVM and MMD combined with OSVM. With regard to OSVM, we first trained the network using source known samples and extracted features using the network, then applied OSVM to the features. When training CNN, we used Adam [34] with a learning rate 2.0×10^{-5}.

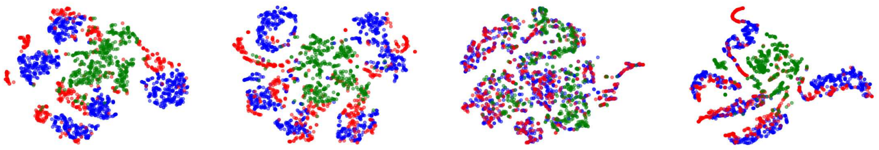
Adaptation from SVHN to MNIST In this experiment, we used all SVHN training samples with numbers in the range from 0 to 4 to train the network. We used all samples in the training splits of MNIST.

Adaptation between USPS and MNIST When using the datasets as a source domain, we used all training samples with number from 0 to 4. With regard to the target datasets, we used all training samples.

Result The quantitative results are shown in Table 5. Our proposed method outperformed other methods. In particular, with regard to the adaptation between USPS and MNIST, our method achieves accurate recognition. In contrast, the adaptation performance on for SVHN to MNIST is worse compared to the adaptation between USPS and MNIST. Large domain difference between SVHN

	SVHN-MNIST				USPS-MNIST				MNIST-USPS				Average			
Method	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK
OSVM	54.3	63.1	37.4	10.5	43.1	32.3	63.5	97.5	79.8	77.9	84.2	**89.0**	59.1	57.7	61.7	65.7
MMD+OSVM	55.9	64.7	39.1	12.2	62.8	58.9	69.5	82.1	80.0	79.8	81.3	81.0	68.0	68.8	66.3	58.4
BP+OSVM	62.9	**75.3**	39.2	0.7	84.4	**92.4**	72.9	0.9	33.8	40.5	21.4	44.3	60.4	69.4	44.5	15.3
Ours	**63.0**	59.1	**71.0**	**82.3**	**92.3**	91.2	**94.4**	**97.6**	**92.1**	**94.9**	**88.1**	78.0	**82.4**	**81.7**	**84.5**	**85.9**

Table 5. Accuracy (%) of experiments on digits datasets.



(a) Source Only (b) MMD (c) BP (d) Ours

Fig. 6. Feature visualization of adaptation from USPS to MNIST. Visualization of source and target features. **Blue points** are source features. **Red points** are target known features. **Green points** are target unknown features.

and MNIST causes the bad performance. We also visualized the learned features in Fig. 6. Unknown classes (5∼9) are separated using our method whereas known classes are aligned with source samples. The method based on distribution matching such as BP [4] fails in adaptation for this open set scenario. When examining the learned features, we can observe that BP attempts to match all of the target features with source features. Consequently, unknown target samples are made difficult to detect, which is obvious from the quantitative results for BP. The accuracy of UNK in BP+OSVM is much worse than the other methods.

5 Conclusion

In this paper, we proposed a novel adversarial learning method for open set domain adaptation. Our proposed method enables the generation of features that can separate unknown target samples from known target samples, which is definitely different from existing distribution matching methods. Moreover, our approach does not require unknown source samples. Through extensive experiments, the effectiveness of our method has been verified. Improving our method for the open set recognition will be our future work.

6 Acknowledgements

The work was partially supported by CREST, JST, and was partially funded by the ImPACT Program of the Council for Science, Technology, and Innovation (Cabinet Office, Government of Japan). We would like to thank Kate Saenko for her great advice on our paper.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)
2. Busto, P.P., Gall, J.: Open set domain adaptation. In: ICCV. (2017)
3. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. (2010)
4. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML. (2015)
5. Gong, B., Grauman, K., Sha, F.: Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In: ICML. (2013)
6. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML. (2015)
7. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. In: NIPS. (2016)
8. Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR. (2012)
9. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: ICML. (2017)
10. Sener, O., Song, H.O., Saxena, A., Savarese, S.: Learning transferrable representations for unsupervised domain adaptation. In: NIPS. (2016)
11. Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W.: Deep reconstruction-classification networks for unsupervised domain adaptation. In: ECCV. (2016)
12. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474 (2014)
13. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: CVPR. (2017)
14. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. arXiv preprint arXiv:1712.02560 (2017)
15. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)
17. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., Erhan, D.: Domain separation networks. In: NIPS. (2016)
18. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR. (2017)
19. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. In: ICLR. (2016)
20. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: NIPS. (2017)
21. Gretton, A., Borgwardt, K.M., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: NIPS. (2007)
22. Long, M., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML. (2017)
23. Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W.: Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: CVPR. (2017)

24. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: ECCV. (2014)
25. Bendale, A., Boult, T.E.: Towards open set deep networks. In: CVPR. (2016)
26. Ge, Z., Demyanov, S., Chen, Z., Garnavi, R.: Generative openmax for multi-class open set classification. In: BMVC. (2017)
27. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: NIPS. (2016)
28. Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., Saenko, K.: Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924 (2017)
29. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
30. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009)
31. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. (2015)
32. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. (2011)
33. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
34. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)