DADIL: Data Augmentation for Domain-Invariant Learning

Shigeru Maya
Corporate Research & Development Center
Toshiba Corporation
Kanagawa, Japan
shigeru1.maya@toshiba.co.jp

Ken Ueno
Corporate Research & Development Center
Toshiba Corporation
Kanagawa, Japan
ken.ueno@toshiba.co.jp

ABSTRACT

In this paper, we propose a new framework named Data Augmentation for Domain-Invariant Learning (DADIL). In the field of manufacturing, labeling sensor data as normal or abnormal is helpful for improving productivity and avoiding problems. In practice, however, the status of equipment may change due to changes in maintenance and settings (referred to as a "domain change"), which makes it difficult to collect sufficient homogeneous data. Therefore, it is important to develop a discriminative model that can use a limited number of data samples. Moreover, real data might contain noise that could have a negative impact. We focus on the following aspect: The difficulties of a domain change are also due to the limited data. Although the number of data samples in each domain is low, we make use of data augmentation which is a promising way to mitigate the influence of noise and enhance the performance of discriminative models. In our data augmentation method, we generate "pseudo data" by combining the data for each label regardless of the domain and extract a domain-invariant representation for classification. We experimentally show that this representation is effective for obtaining the label precisely using real datasets.

CCS CONCEPTS

• **Computing methodologies** → *Supervised learning by classification*; *Classification and regression trees*; *Neural networks*;

KEYWORDS

few shot learning, domain adaptation, non-linear function

ACM Reference Format:
Shigeru Maya and Ken Ueno. 1997. DADIL: Data Augmentation for Domain-Invariant Learning. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97),* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

In the era of the Internet of Things (IoT), sensor data are used for many applications. Information about the manufacturing processes of various products can be recorded using sensor data, making it possible to determine the relationship between the information about each process and the normality / abnormality of the products. However, the status of the equipment may change frequently mainly due to maintenance, changes in settings, and changes in operators. This makes it difficult in practice to obtain a large amount of data after these kinds of changes. Therefore, limited data and domain changes are the two main problems for this task. In this paper, we address these problems by using data augmentation with the aim of constructing an accurate discriminative model.

A typical approach is to focus on only the data obtained under the current conditions. However, we generally need a large number of data samples in order to achieve positive performance. Recently, "few-shot learning," which constructs a discriminative model from few samples has received much attention [5]. This method has been developed in the field of image recognition. Many kinds of problem settings have been proposed for few-shot learning. For example, in the field of image recognition, it is a heavy burden to obtain label data for each image, whereas the images themselves are easy to collect. To make use of this particular problem setting, a method that constructs a generative model from a large amount of unlabeled data and a small amount of labeled data has been proposed [12] [18]. A convolutional neural network (CNN) is the de facto standard for generating features for image recognition and is used to learn the proper embedded space for this task [13] [20] [24]. In another approach, CNN is used to construct the generative model for each label [12] [18]. However, even collecting sufficient unlabeled data is difficult because the status of equipment changes frequently.

We assume that the data for each domain are limited and that normal/abnormal data are generated from different distributions. This setting is illustrated in Figure 1. The corresponding separation boundary based on the data is denoted $\hat{w}$. In this case, we try

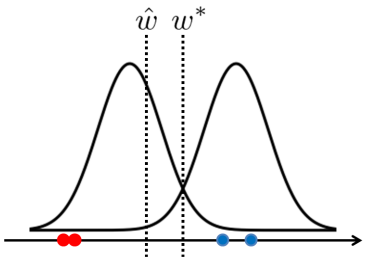

Figure 1: Schematic figure of data distribution. Blue circles show normal data and red circles show abnormal data. The line w^* shows the optimal separator and the line $\hat{w}$ shows the separator based on the data. When the data size is inadequate, the bias between $\hat{w}$ and w^* can be large.

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

to minimize the max-margin loss over data. The optimal separator is denoted w^*. According to this figure, the bias between the ideal boundary and the one derived from the data differs due to insufficient data. Although these approaches are promising in image recognition, difficulties remain in other fields such as manufacturing due to the limited data.

Next, another difficulty in this problem is the change of domain. In this paper, we refer to the former domain as the "source domain" and the latter domain as the "target domain." Domain adaptation [1] is the task of handling different distributions due to domain change. A typical approach is to align the distribution of the source domain and target domain [14] [22]. This approach uses the Maximum Mean Discrepancy (MMD [2]) to measure the distance between domains. In this approach, we need much data for estimating the distributions of the source domain and target domain precisely. We show this setting in Figure 2. Figure 2 (a) shows examples of source and target domain data. Then, we force the data to align based on the small amount of observed data in Figure 2 (b). As shown in this figure, we cannot capture the distribution of each domain from a small amount data and the alignment is inappropriate. Figure 2 (c) shows the optimal alignment of the distributions.

Up to this point, we have reviewed the challenges of few-shot learning task and the domain adaptation task. The common challenge in few-shot learning and domain adaptation in fields other than image recognition is an insufficient number of data samples. When data are scarce, it is difficult to build a precise discriminative model and estimate the distribution of both domains. Moreover, real data such as sensor data might contain noise. Overcoming this problem would allow practical use of this method in most industrial fields, enabling quick decision-making because users would not need to wait for a large amount of data to be stored in databases.

We deal with these problems by data augmentation. We focus on the fact that the types of labels are the same in the source and target domains (two types of labels: normal and abnormal). In our method named Data Augmentation for Domain-Invariant Learning (DADIL), we generate pseudo data by combining the data for each label across domains. Although the data in each domain are limited, we can generate plenty of pseudo data by combining both source domain and target domain data. Moreover, this method has the effect of being able to mitigate the impact of noise from a probabilistic perspective. We apply the pseudo data and map them to the embedded space using contrastive loss in order to learn the proper metric for classification. Contrastive loss forces data with the same labels to be nearer and data with different labels to be more distant [3]. We combine data augmentation and contrastive loss in order to seek a domain-invariant feature that is robust to noise. We illustrate the basic idea in Figure 3.

The contributions of this paper are as follows. We propose a new method for dealing with limited data with domain adaptation. We focus on the fact that the essence of the problem is the limited amount of data and propose the use of data augmentation. In the proposed method, data augmentation enables us to generate plenty of data and obtain a domain-invariant representation for each label, which is useful for this task. We explain the effectiveness of data augmentation for mitigating the influence of noise. Deep learning allows an end-to-end learning and our approach makes use of it to obtain the appropriate embedded space from input data directly. Our method using deep learning is not specific to image recognition and can be used in many fields. Our method enables to learn the precise model based on a small number of data samples. This can be highly utilized for the improvement of productivity in many areas. In that sense, we will contribute to Utility-Driven Mining.

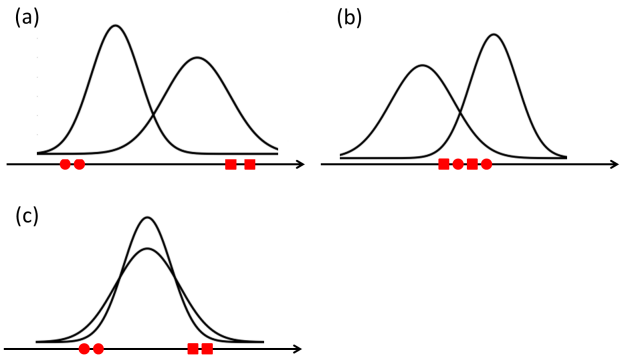

Figure 2: Schematic figure of domain adaptation. Red circles show source domain data and red boxes show target domain data. (a) Examples of source and target domain data. (b) Alignment of the distributions of the source domain and target domain based on the data. (c) Optimal alignment of the source domain and target domain distributions.

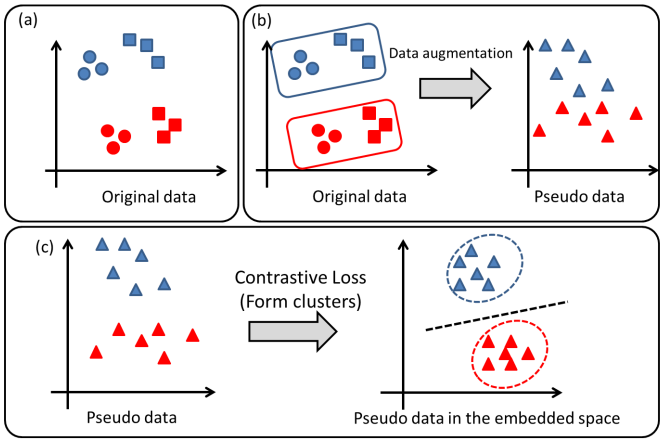

Figure 3: Schematic illustration of the proposed method DADIL. (a) Original normal and abnormal data in the source and target domain. Blue and red show normal and abnormal data. Circles and boxes show source and target domain data. (b) Pseudo data for normal and abnormal data generated by combining the source and target domain data. (c) Learning of the proper embedded space using contrastive loss. Both normal and abnormal data become more clustered compared with (b).

The rest of the paper is organized as follows. We introduce related works in Section 2. In Section 3, we describe the proposed method. We show the experimental results in Section 4. We conclude our paper in Section 5.

2 RELATED WORKS

Approaches have been proposed for dealing with few-shot learning (one-shot learning) and domain adaptation. Fei-Fei et al. firstly proposed the task of one-shot learning for image classification based on a Bayesian approach [5]. One-shot learning using deep learning can be broadly divided into two categories. One category is using generative models [12] [18]. These approaches build a generative model for each label and have the advantage of being able to use unlabeled data. This approach is effective if we have plenty of unlabeled data. In our scenario, we are interested in the other approach of methods based on manifold learning. This approach involves finding the proper embedded space for few-shot learning. A Siamese network is a typical method that learns an embedded space such that samples from the same class are mapped as close as possible to each other using contrastive loss [13]. Recently, several papers based on Siamese networks have been proposed [17] [24]. One-shot learning is related to the task of class imbalanced. Class imbalance is the setting where the number of samples in each class is unequal and sometimes low. Wallace et al. advocate the approach using both undersampling and bagging [25].

Next, we deal with domain adaptation. The basic approach for domain adaptation is to find an invariant representation between the source domain and target domain [21]. Ganin et al. proposed a method that combines domain adaptation and deep feature learning [6]. This method attempts to maximize the classification loss for the domain in order to obtain the embedded space where the difference between the source domain and target domain is indistinguishable. Many approaches based on Maximum Mean Discrepancy (MMD [2]) have been proposed [14] [22]. This approach aims to confuse the data from both the source and target domain to find an invariant data representation. Motiian et al. proposed a method for domain adaptation exploiting a Siamese network [17]. Recently, adversarial learning has been used to find the domain-invariant representation. Motiian et al. proposed a new domain adaptation method using adversarial learning [16]. Jiang et al. proposed a 2-step domain adaptation task that can be used in fields other than image recognition [10]. The first step is to obtain the generalized feature across domains and the second step is to obtain the domain specific feature. Hu et al. learned a deep metric network by maximizing the inter-class variations and minimizing the intra-class variations [9]. The concept of this approach is similar to that of Siamese networks. These domain adaptation approaches treat source and domain distributions separately and we need sufficient data samples to capture each distribution well. Therefore, there is a high probability that these approaches will not work when the data size is limited.

In this paper, we use the data augmentation technique. Next, we introduce methods of data augmentation. Noise injection has been used in many fields including speech recognition [7]. Recently, Zhang et al. increased the amount of data by considering the combinations of pairs of examples [26]. The effectiveness of this approach from the view point of waveforms has been demonstrated ([19]).

3 DATA AUGMENTATION FOR DOMAIN-INVARIANT LEARNING

In this section, we propose a new framework for dealing with domain adaptation based on a small number of data samples.

In our problem setting, we deal with the binary classification task where there are two labels both in the source domain and the target domain. We assume that the normal and abnormal data are generated from different probabilistic distributions.

Our goal is to construct an accurate discriminative model for the target domain data ($\mathcal{D}^{\mathrm{T}}_{\mathrm{normal}}, \mathcal{D}^{\mathrm{T}}_{\mathrm{abnormal}}$). In Section 3.1, we introduce our novel data augmentation method for generating pseudo data by combining the data in across domains and explain the feature of this method from the probabilistic perspective. In Section 3.2, we consider the proper embedded space of pseudo data for classification. In Section 3.3, we introduce the objective function of DADIL. In Section 3.4, we describe how to optimize the objective function. In Section 3.5, we introduce an extension of DADIL named Latent-DADIL. In Section 3.6, we introduce the method for obtaining the label for evaluation.

3.1 Generating pseudo data and its feature

In this section, we introduce the method for generating pseudo data and explain its feature from a probabilistic perspective. We assume that the data for each domain is limited and that normal/abnormal data are generated from different distributions.

Mix-up [26] has recently been proposed as a data augmentation technique for increasing the amount of data by combining pairs of data and their labels. We are motivated by Mix-up and explain our data augmentation method from a probabilistic perspective. Let $\boldsymbol{x}_i^{\mathrm{normal,S}} \in \mathbb{R}^n$ and $\boldsymbol{x}_j^{\mathrm{normal,T}} \in \mathbb{R}^n$ be the ith and jth normal data in the source domain and target domain, respectively. We denote abnormal data in a similar manner.

Let the pseudo data $x_{i,j}^{\mathrm{normal}}$ be the combination of the ith normal data in source domain and jth normal data in target domain. We represent $x_{i,j}^{\mathrm{normal}}$ as

$$x_{i,j}^{\mathrm{normal}} = \lambda x_i^{\mathrm{normal,S}} + (1-\lambda) x_j^{\mathrm{normal,T}}. \tag{1}$$

By combining the data and varying the value of λ, we can increase the number of data. We need lots of data to measure the distribution for each domain for domain adaptation (such as MMD), which is difficult in practice. We consider a wide variety of combinations between source and target domain data to generate pseudo data. Therefore, we can avoid this problem.

Next, we explain the effectiveness of Eq. (1) from a probabilistic viewpoint. Let $\mathcal{D}_{\mathrm{normal,S}}, \mathcal{D}_{\mathrm{normal,T}}$ be the normal dataset in the source and target domains, respectively. We also assume that the data are drawn from the following normal distribution:

$$\boldsymbol{X}_{\mathrm{normal,S}} \sim P_{\mathrm{normal,S}} = \mathcal{N}(\mu_{\mathrm{normal,S}}, \sigma^2_{\mathrm{normal,S}}) \tag{2}$$

$$\boldsymbol{X}_{\mathrm{normal,T}} \sim P_{\mathrm{normal,T}} = \mathcal{N}(\mu_{\mathrm{normal,T}}, \sigma^2_{\mathrm{normal,T}}). \tag{3}$$

If the value of $\sigma^2_{\mathrm{normal}}$ is large, it leads to the data tending to be affected by noise. According to Eq. (1), two variables $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$

follow normal distributions $(P_{\text{normal,S}}, P_{\text{normal,T}})$, independently. In this case, according to the reproductive property of the normal distribution, the variable $\lambda \boldsymbol{X}_1 + (1-\lambda)\boldsymbol{X}_2$ follows

$$\lambda \boldsymbol{X}_1 + (1-\lambda)\boldsymbol{X}_2 \sim \tag{4}$$

$$\mathcal{N}(\lambda \mu_{\text{normal,S}} + (1-\lambda)\mu_{\text{normal,T}}, \lambda^2 \sigma^2_{\text{normal,S}} + (1-\lambda)^2 \sigma^2_{\text{normal,T}}). \tag{5}$$

We focus on the variance of this variable. If the value of $\sigma^2_{\text{normal,S}}$ and $\sigma^2_{\text{normal,T}}$ are the same $(\sigma^2_{\text{normal}})$, the variance of $\lambda \boldsymbol{X}_1 + (1-\lambda)\boldsymbol{X}_2$ is $(\lambda^2 + (1-\lambda)^2)\sigma^2_{\text{normal}}$. It follows that the variance becomes small as long as we set the values of λ in the range $[0, 1]$. If the value of $\sigma^2_{\text{normal,S}}$ is smaller than $\sigma^2_{\text{normal,T}}$, the same can be said. Therefore, we set the value of λ uniformly within $[0, 1]$ to generate pseudo data. Since λ follows a uniform distribution, we cannot tell whether the pseudo data comes from the source or target domain and can generate the domain-invariant pseudo data. Furthermore, our data augmentation technique can mitigate the impact of noise.

3.2 Learning embedded space

In this section, we map the data into the embedded space to capture the features of the normal and abnormal data clearly. Let $\boldsymbol{x} \in \mathbb{R}^n$ be the data, d be the dimension of the embedded space, and $\psi(\cdot) : \mathbb{R}^n \to \mathbb{R}^d$ be the mapping function. The corresponding representation in the embedded space is $\psi(\boldsymbol{x}) \in \mathbb{R}^d$.

To find the proper embedded space, we utilize contrastive loss, which is used for the few-shot learning task [13]. The contrastive loss is the loss defined for each pair of data. Taking a pair of data $\boldsymbol{y}_1 \in \mathbb{R}^d, \boldsymbol{y}_2 \in \mathbb{R}^d$, the contrastive loss is

$$L(\boldsymbol{y}_1, \boldsymbol{y}_2) = \frac{1}{2}(z(||\boldsymbol{y}_1 - \boldsymbol{y}_2||_2^2 + (1-z)\max(1 - ||\boldsymbol{y}_1 - \boldsymbol{y}_2||_2^2, 0)^2), \tag{6}$$

where

$$z = 1 \text{ if the label of } \boldsymbol{y}_1 \text{and } \boldsymbol{y}_2 \text{is different} \tag{7}$$

$$z = 0 \text{ if the label of } \boldsymbol{y}_1 \text{and } \boldsymbol{y}_2 \text{is the same.} \tag{8}$$

This means that the corresponding data should be near in the embedded space if the labels are the same and vice versa. It follows that the normal and abnormal data should each be clustered. Although the pseudo data described in Section 3.1 are helpful for decreasing the variance of pseudo data of each label, the feature of each label of pseudo data can be further enhanced by combining contrastive loss. Since the normal and abnormal pseudo data described in Section 3.1 is domain-invariant, the corresponding representation in the embedded space is also domain-invariant and normal and abnormal pseudo data form clusters that are separate from each other due to contrastive loss (see Figure 3(c)). To the best of our knowledge, this is the first paper to generate domain-invariant data using data augmentation applied to this task.

3.3 Objective function to be minimized

In this section, we describe the objective function for finding the embedded space. Let $\psi(\cdot) : \mathbb{R}^n \to \mathbb{R}^d$ be the mapping function.

Algorithm 1 The algorithm of DADIL.

Input: Normal and abnormal data in the source and target domains $(\mathcal{D}^{\text{S}}_{\text{normal}}, \mathcal{D}^{\text{S}}_{\text{abnormal}}, \mathcal{D}^{\text{T}}_{\text{normal}}, \mathcal{D}^{\text{T}}_{\text{abnormal}})$.
Batch size:B.
Output: The parameters of model $\psi(\cdot)$.

```
• Build the model.
Initialize the parameters of ψ(·).
for l = 1 → # of iterations do
    Select B data from D^S_normal, D^S_abnormal, D^T_normal, and D^T_abnormal allowing duplication.
    Generate pseudo data according to Eqs. (9) and (10).
    Obtain the objective function according to Eq. (11).
    Update the parameters of the model ψ(·) by decreasing the objective function with Adam.
end for
```

We denote the normal data in the source domain and target domain by $\mathcal{D}_{\text{normal,S}}$ and $\mathcal{D}_{\text{normal,T}}$, respectively. We denote abnormal data in a similar manner. We represent the pseudo data of normal and abnormal data combining the ith and jth data samples in the source domain and target domain as $\hat{\boldsymbol{x}}^{\text{normal}}_{i,j}$ and $\hat{\boldsymbol{x}}^{\text{abnormal}}_{i,j}$. Therefore, $\hat{\boldsymbol{x}}^{\text{normal}}_{i,j}$ and $\hat{\boldsymbol{x}}^{\text{abnormal}}_{i,j}$ are as follows:

$$\hat{\boldsymbol{x}}^{\text{normal}}_{i,j} = \lambda \boldsymbol{x}^{\text{normal,S}}_{i} + (1-\lambda)\boldsymbol{x}^{\text{normal,T}}_{j} \tag{9}$$

$$\hat{\boldsymbol{x}}^{\text{abnormal}}_{i,j} = \lambda \boldsymbol{x}^{\text{abnormal,S}}_{i} + (1-\lambda)\boldsymbol{x}^{\text{abnormal,T}}_{j}. \tag{10}$$

Using this notation, the objective function to be minimized is given by

$$\text{Loss}(\mathcal{D}, \lambda) + \lambda_1 ||\psi(\cdot)||_2^2 \tag{11}$$

,where

$$\text{Loss}(\mathcal{D}, \lambda) = \sum_{i=1,j=1}^{|\mathcal{D}_{\text{normal,S}}|\,|\mathcal{D}_{\text{normal,T}}|} \sum_{k=1,l=1} L(\psi(\hat{\boldsymbol{x}}^{\text{normal}}_{i,k}), \psi(\hat{\boldsymbol{x}}^{\text{normal}}_{j,l})) \tag{12}$$

$$+ \sum_{i=1,j=1}^{|\mathcal{D}_{\text{abnormal,S}}|\,|\mathcal{D}_{\text{abnormal,T}}|} \sum_{k=1,l=1} L(\psi(\hat{\boldsymbol{x}}^{\text{abnormal}}_{i,k}), \psi(\hat{\boldsymbol{x}}^{\text{abnormal}}_{j,l})) \tag{13}$$

$$+ \sum_{i=1,j=1}^{|\mathcal{D}_{\text{normal,S}}|\,|\mathcal{D}_{\text{abnormal,T}}|} \sum_{k=1,l=1} L(\psi(\hat{\boldsymbol{x}}^{\text{normal}}_{i,j}), \psi(\hat{\boldsymbol{x}}^{\text{abnormal}}_{k,l})). \tag{14}$$

We employ a multi-layer neural network as $\psi(\cdot)$ and $||\psi(\cdot)||_2^2$ is the L2-norm of the weight matrices.

3.4 Optimization

Since we employ a sigmoid function and the contrastive loss is not a convex loss, it is impractical to obtain the global solution. This method can be regarded as a deep learning method and we optimize the objective function Eq. (11) by Stochastic Gradient Decent (SGD). We use Adam [11], which is a kind of stochastic gradient method for optimization. We set the batchsize (B) and take B data from each domain and aim to decrease the objective function. In Alg. 1 we show the algorithm of DADIL.

3.5 Latent-DADIL

In the proposed method, we merge the pair of data to generate plenty of data. In this section, we extend DADIL and name this extension Latent-DADIL. One of the main advantages of deep learning using non-linear functions is that it generates features automatically.

We try to extract the essential information via non-linear functions and utilize them to build the model. In DADIL we merge data across domains. We map the data into a low dimensional latent space using non-linear function and merge the data in the latent space to generate pseudo data. After obtaining the pseudo data, the following part is the same as in DADIL. We denote the size of the latent space by m. Let $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ be the mapping function to the latent space and we denote the pseudo data of normal and abnormal data combining the ith and jth data samples in the source domain and target domain by $\tilde{\boldsymbol{x}}_{i,j}^{\text{normal}}$ and $\tilde{\boldsymbol{x}}_{i,j}^{\text{abnormal}}$. Therefore, $\tilde{\boldsymbol{x}}_{i,j}^{\text{normal}}$ and $\tilde{\boldsymbol{x}}_{i,j}^{\text{abnormal}}$ are as follows:

$$\tilde{\boldsymbol{x}}_{i,j}^{\text{normal}} = \lambda\phi(\boldsymbol{x}_i^{\text{normal,S}}) + (1-\lambda)\phi(\boldsymbol{x}_j^{\text{normal,T}}), \tag{15}$$

$$\tilde{\boldsymbol{x}}_{i,j}^{\text{abnormal}} = \lambda\phi(\boldsymbol{x}_i^{\text{abnormal,S}}) + (1-\lambda)\phi(\boldsymbol{x}_j^{\text{abnormal,T}}). \tag{16}$$

The objective function of Latent-DADIL is similar to that of DADIL, except for how the pseudo data are generated. Note that we can update the parameters of $\phi(\cdot)$ and $\psi(\cdot)$ simultaneously using SGD in an end-to-end fashion and do not need knowledge of the proper latent space.

3.6 Method for inference

Up to this point, we have discussed how the model ($\psi(\cdot)$) is built. Next, we describe how the test data is classified based on the model. The k nearest neighbor (k-nn) method in embedded space is widely used in few-shot learning and generally k is set to be 1 [13]. We follow this procedure for evaluation. We measure the distance between the test data and the target domain data in the embedded space and take the label of the nearest one within the target domain data. Let $\boldsymbol{x}$ be the test data and $\boldsymbol{x}_i^{\text{test}}$ be the ith data in the target domain. We estimate the label of target data $\boldsymbol{x}$ as the label of the jth data in the target domain, where $j = \arg\min_i ||\psi(\boldsymbol{x}) - \psi(\boldsymbol{x}_i^{\text{test}})||_2^2$. For Latent-DADIL, we measure the distance in the similar manner.

4 EXPERIMENTS

In this section, we evaluate the effectiveness of our method using real datasets.

4.1 Datasets

Firstly, we introduce two kinds of real datasets. One dataset is the MNIST[1], which comprises handwritten grayscale images. Each piece of image data corresponds to a character from 0 to 9. The size of each image in the data is 28×28 and the dimension of the data is $784(28\times28)$. The other dataset is the USPS[2]. This is also a dataset of handwritten grayscale images. Each piece of image data corresponds to characters from 0 to 9. The size of each image is 16×16 and the dimensions of the data are $256(16\times16)$. These datasets are widely used for the few-shot learning task and domain adaptation task [13][17]. We show examples of images from MNIST and USPS in Figures 4.

Next, we explain the preprocessing of the dataset. We utilize the MNIST and USPS datasets as the source and target domain data. In our proposed methods, we use the common function ($\psi(\cdot)$) to map the data in both the source and target domains to the embedded space. However, the dimensions of the MNIST and USPS are different and we cannot apply the same mapping function. Therefore, as preprocessing we resize both datasets to 32×32 (= 1024 dimension). Originally, the value in each dimension (x) of the grayscale image takes the value from 0 to 255. We then divide the data x by 255 to obtain a value from 0 to 1.

Examples of promising ways to handle image datasets include using specific features such as HOG [4] or SIFT [15] or Resnet [8]. However, these methods are specialized for image data. Since we assume that we are utilizing structured data such as sensor data, we do not use any methods specialized for image data.

4.2 Methods in comparison

We introduce five methods for comparison. We denote the source and target domain data by $\mathcal{D}_{\text{S}}$ and $\mathcal{D}_{\text{T}}$, respectively.

4.2.1 k-nn. The first comparison method is the traditional k-nn method. We set the value of k to 1. We use the L2-norm to measure the distance and obtain the label of the test data referring to the target domain data that is closest to the test data.

4.2.2 t-SNE. tSNE [23] aims to map the data into low dimensional space while preserving the distance among data. This method is utilized in many situations including visualization and embedded learning. In this paper, we set the dimension of the embedded space to 2 and infer the label using 1-nn. We learn the embedded space for each set of test data and the target-domain data.

4.2.3 MMD. This method is based on MMD [2]. We try to align the data within the source and target domain data in the embedded space. We use hinge loss and 1-nn to obtain the label for evaluation. The objective function to be minimized is

$$\text{Loss}_{\text{MMD}_{\text{S}}}(\mathcal{D}_{\text{S}}) + \text{Loss}_{\text{MMD}_{\text{T}}}(\mathcal{D}_{\text{T}}) \tag{17}$$

$$+\lambda_1||\frac{1}{|\mathcal{D}_{\text{S}}|}(\sum_i^{|\mathcal{D}_{\text{S}}|}\psi(x_i)) - \frac{1}{|\mathcal{D}_{\text{T}}|}(\sum_i^{|\mathcal{D}_{\text{T}}|}\psi(x_i))||_2^2 + \lambda_1||\psi(\cdot)||_2^2 \tag{18}$$

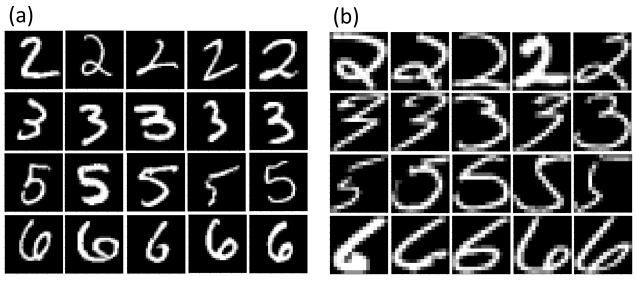

Figure 4: (a)MNIST dataset. (b) USPS dataset. Each row represents 2,3,5, and 6.

[1] http://yann.lecun.com/exdb/mnist/
[2] https://cs.nyu.edu/~roweis/data.html

,where we use hinge loss for $\text{Loss}_{\text{MMD}_\text{S}}(\mathcal{D})$ and $\text{Loss}_{\text{MMD}_\text{T}}(\mathcal{D})$ over the dataset $\mathcal{D}$ applied $\psi(\cdot)$. Note that we provide two classifiers for source and target domains using hinge loss.

4.2.4 Siamese. This method is based on a Siamese network [13] and uses only target data to build the model. To obtain the label of the test data, we use 1-nn. The objective function to be minimized is $\text{Loss}_{\text{Siamese}}(\mathcal{D}_\text{T}) + \lambda_1||\psi(\cdot)||_2^2$, where $\text{Loss}_{\text{Siamese}}(\mathcal{D})$ is the contrastive loss over the dataset $\mathcal{D}$ applied $\psi(\cdot)$.

4.2.5 Siamese-domains (S-domains). This method is inspired by Siamese networks [13] and uses the same function to map the data into the embedded space for the source and target domains. To obtain the label of the test data, we use 1-nn. The objective function to be minimized is $\text{Loss}_{\text{Siamese}}(\mathcal{D}_\text{S}) + \text{Loss}_{\text{Siamese}}(\mathcal{D}_\text{T}) + \lambda_1||\psi(\cdot)||_2^2$.

4.3 Parameter settings

In this section, we describe the parameter settings. We use a 3-layer neural network for $\psi(\cdot)$ for all methods except for k-nn, tSNE, and Latent-DADIL. For the network architecture of $\psi(\cdot)$, the size of each layer is [1024,10,2] and the activation function between the input layer and the hidden layer is a Rectified Linear Unit (ReLU) and the one between the hidden layer and the output layer is a sigmoid function. For Latent-DADIL both $\phi(\cdot)$ and $\psi(\cdot)$ are 2-layer networks and $\phi(\cdot)$ maps the data into 10-dimensional space and $\psi(\cdot)$ maps the data into 2-dimensional space. Activation functions of $\phi(\cdot)$ and $\psi(\cdot)$ are ReLU and sigmoid functions, respectively.

The hyperparameter λ_1 are selected over $\left\{10^{-7}, 10^{-6}, \ldots, 10^{6}, 10^{7}\right\}$ following each problem setting. We use default hyperparameters for Adam, set the batch size B to 30, and the number of iterations in Alg. 1 to 4000. Figure 5 shows the network architecture of the proposed method.

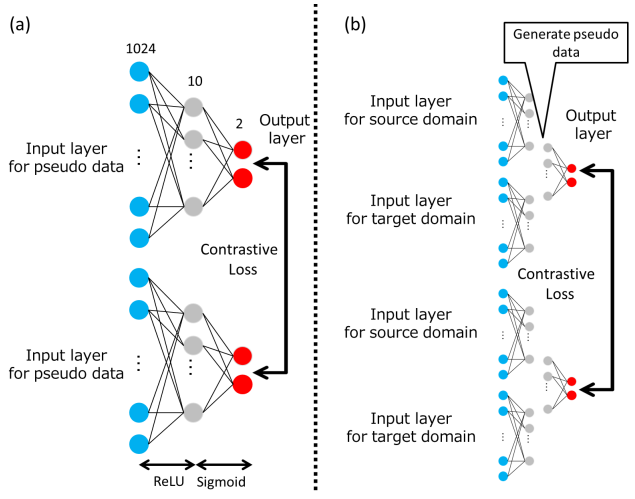

Figure 5: Network architecture of the proposed method. (a) DADIL. (b) Latent-DADIL.

4.4 Experimental settings

In this section, we describe the experimental settings. We provided a pair of sets (Set A and Set B) for each experiment. We varied the number of normal and abnormal data in each source and target domain in $\{1, 3, 5, 7, 9\}$. For evaluation, we generated the test data comprising of 100 normal and abnormal data belonging to the target domain. We performed 100 repetitions of shuffling source domain data, target domain data, and test data and evaluate the performance for the each experimental setting. Here, we set the number of normal and abnormal data to be equal and we measure the performance using accuracy. Let n_1,n_2,n_3, and n_4 are the number of true positives, false positives, false negatives, and true negatives, respectively. Accuracy is given as $\frac{n_1+n_4}{n1+n_2+n_3+n_4}$.

In each experiment, we utilize either or both the MNIST and USPS dataset. For Set A, data labeled 2 and 3 are normal data and abnormal data, respectively. For Set B, data labeled 5 and 6 are normal data and abnormal data, respectively.

Next, we describe how to choose the hyperparameter λ_1 of methods for Set A. In each experiment, we apply all $\lambda_1 \in \left\{10^{-7}, 10^{-6}, \ldots, 10^{6}, 10^{7}\right\}$ to methods for Set B. Then, we take the one that gives the highest accuracy and set is as the hyperparameter of the corresponding method for Set A. We set the hyperparameter of methods for Set B in a similar manner. Note that we do not use any information of Set A to choose the hyperparameter for Set A. Finally, we introduce five scenarios for few shot learning with domain adaptation.

Noise shift (MNIST)
It is conceivable that the influence of noise is altered by the domain change. In this setting, we use the MNIST dataset and consider the domain change to be noise injection. For the target domain data, we add noise following a normal distribution with $\mathcal{N}(0, \sigma^2)$, where σ is selected from $\{0.0, 1.0, 2.0\}$.

Parallel shift (MNIST)
In another case of domain change, the behavior might be different. In this setting, we use the MNIST and consider the domain change of adding a constant value. For the target domain data, we add the value $C \in \{0.0, 1.0, 2.0\}$.

Noise shift (USPS)
This experimental setting is similar to noise shift (MNIST). The difference is that we use the USPS dataset instead of the MNIST dataset.

Parallel shift (USPS)
This experimental setting is similar to parallel shift (MNIST). The difference is that we use the USPS dataset instead of the MNIST dataset.

Dataset shift
When a domain change occurs, features in the data might change slightly. To consider this situation, we use both the MNIST and USPS datasets. Although both MNIST and USPS datasets represent numbers from 0 to 9, the features of each dataset are slightly different as shown in Figure 4. In this setting, we regard the USPS dataset as the source domain and the MNIST dataset as the target domain.

4.5 Results on MNIST

In this section, we describe the results of Set A for noise shift (MNIST) and parallel shift (MNIST). We show the mean accuracy

by varying the number of samples in Figures 6 and 7. In terms of noise shift (MNIST), the proposed method worked well even when the number of samples was only one, as shown in Figure 6(a). The proposed method effectively captured normal and abnormal features as we increased the number of samples (particularly when the number of samples was 7 or 9) when we add noise, as shown in Figures 6 (b) and (c). This implies that our method is robust to noise. In terms of parallel shift (MNIST), the proposed method clearly outperformed the others particularly when the number of samples

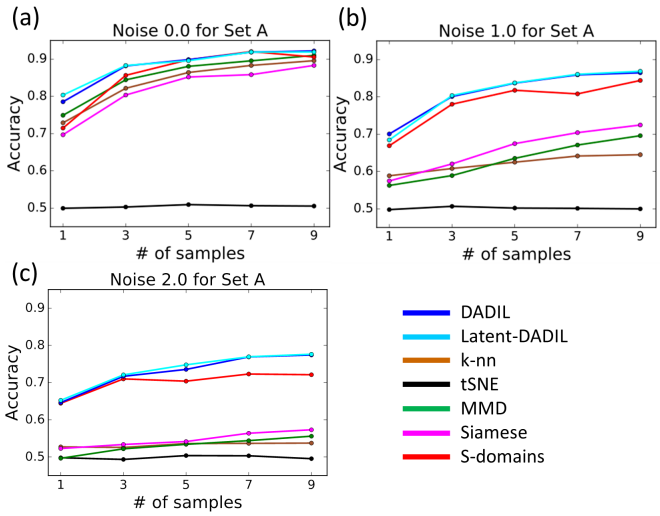

Figure 6: Mean accuracy of Noise shift (MNIST) by varying the number of samples. (a) Noise 0.0. (b) Noise 1.0. (c) Noise 2.0.

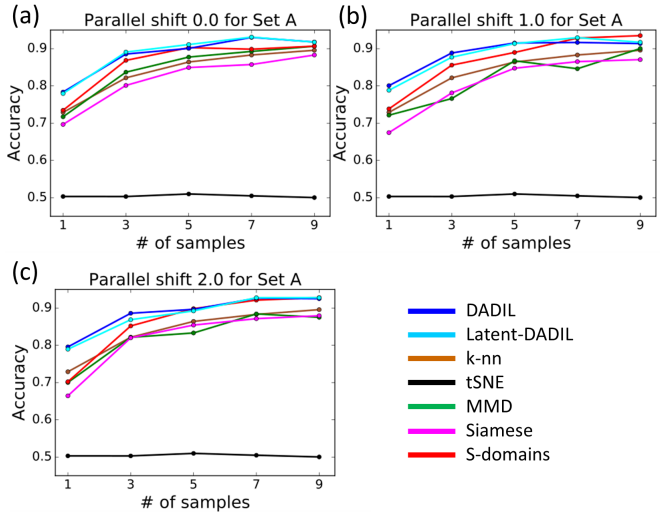

Figure 7: Mean accuracy of Parallel shift (MNIST) when varying the number of samples. (a) Parallel shift 0.0. (b) Parallel shift 1.0. (c) Parallel shift 2.0.

Table 1: Mean accuracy for "Noise shift: 2.0; number of samples: 9" and "Parallel shift: 2.0; number of samples: 1".

	k-nn	t-SNE	MMD	Siamese	S-domains	DADIL	Latent-DADIL
Noise shift	0.537	0.495	0.556	0.573	0.721	0.774	0.776
Parallel shift	0.729	0.503	0.700	0.664	0.702	0.796	0.790

Table 2: One-sided binomial test for Noise shift between the proposed method and the comparison methods. Each cell is described by "# of wins / # of loses". Cells marked * show cases where $p < 0.01$.

Noise shift	k-nn	t-SNE	MMD	Siamese	S-domains
DADIL	100/0*	100/0*	100/0*	100/0*	63/31*
Latent-DADIL	100/0*	100/0*	100/0*	100/0*	61/27*
Parallel shift	k-nn	t-SNE	MMD	Siamese	S-domains
DADIL	77/20*	100/0*	81/19*	73/21*	82/14*
Latent-DADIL	78/21*	100/0*	81/19*	73/23*	82/15*

was small (1 or 3). For parallel shift, the proposed method constantly performed better regardless of the degree of parallel shift.

According to these figures, S-domains was powerful in cases such as "Noise shift: 2.0; number of samples: 1" and "Parallel shift: 1.0; number of samples: 9," although the performance of other methods including k-nn and MMD were not promising. tSNE represents the distance between data in the embedded space based on the probabilistic distribution. Since the data size is limited, it is difficult to capture the proper probabilistic distribution and so it showed the worst performance among all methods. However, the proposed method achieved positive performance overall. In particular, in the setting of "Noise shift: 2.0; number of samples: 9" and "Parallel shift: 2.0; number of samples: 1," the difference in mean accuracy between the proposed method and comparison methods was clear. The corresponding values of mean accuracy are summarized in Table 1.

To check the statistical significance of these results, we conducted one-sided binomial tests between our methods and the comparison methods. For each experimental setting, we performed 100 repetitions of shuffling the source, target, and test data and evaluated the performance and count the number of wins and losses of the proposed method against the comparison methods and obtained the p value. The null hypothesis was that the accuracy of our method and the other method were equal. These results for "Noise shift: 2.0; number of samples: 9" and "Parallel shift: 2.0; number of samples: 1" are summarized in Table 2. Note that we ignored the case where the accuracy was exactly the same between the proposed method and comparison methods. According to this table, our method was significantly better than the comparison methods in these cases. Although the S-domains method was powerful for this dataset, the proposed method performed well constantly and showed a significant improvement in some cases.

4.6 Results on USPS

In this section, we describe the results of Set A for noise shift (USPS) and parallel shift (USPS). We show the mean accuracy by varying the number of samples in Figures 8 and 9.

For noise shift (USPS), our methods worked well when the number of sample was only one and the influence of noise was relatively low as shown to Figure 8(a) and (b). When we added noise, however, our proposed methods were superior to other methods especially when the number of samples was 5, 7, and 9 based on the results in Figure 8(c). These results also indicate that our proposed methods are robust to noise. For parallel shift (USPS), our proposed methods constantly outperformed methods for comparison when the number of samples was 1,3, and 5. In some cases such as "Noise shift: 2.0; number of samples: 3" and "Parallel shift: 0.0; number of samples: 7", S-domains showed the positive performance. However, our proposed methods constantly performed well and greatly outperformed the other methods in the settings of "Noise shift: 2.0; number of samples: 7" and "Parallel shift: 2.0; number of samples: 1".

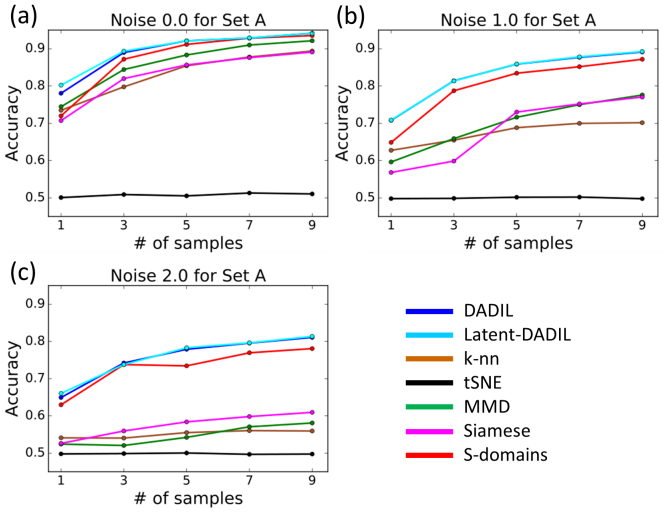

Figure 8: Mean accuracy of Noise shift (USPS) varying the number of samples. (a) Noise 0.0. (b) Noise 1.0. (c) Noise 2.0.

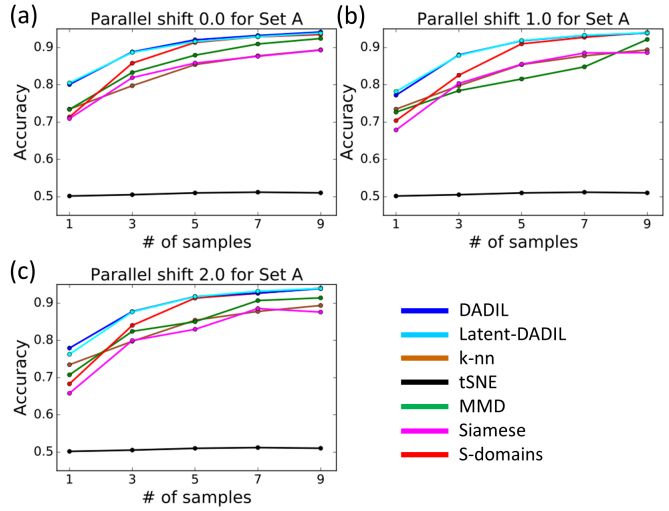

Figure 9: Mean accuracy of Parallel shift (USPS) varying the number of samples. (a) Parallel shift 0.0. (b) Parallel shift 1.0. (c) Parallel shift 2.0.

Table 3: Mean accuracy for " number of samples: 3".

k-nn	t-SNE	MMD	Siamese	S-domains	DADIL	Latent-DADIL
0.816	0.505	0.796	0.759	0.803	0.862	0.862

Next, we show how hyperparameter λ_1 affects the performance and check the effectiveness of our proposed method from the robustness of the hyperparameter. As shown in Figure 10, we obtain the mean accuracy by varying the value of λ_1. In these case, we confirmed that our proposed methods had consistently positive performance as long as the value of λ_1 was less than 10^2.

4.7 Results for Dataset shift

In this section, we focus on the Dataset shift. In this experiment we show the results for both Set A and Set B. We set the hyperparameter λ_1 for Set B based on the accuracy of Set A. The results are shown in Figure 11 by varying the number of samples. For Set A, the proposed method performed well regardless of the number of samples and outperformed the comparison methods particularly when the number of samples was 3, as shown in Figure 11(a). The corresponding mean accuracy is summarized in Table 3. To check the statistical significance of the result, we conducted the one-sided binomial test introduced in Section 4.5. The results are summarized in Table 4. The proposed method showed significantly better performance.

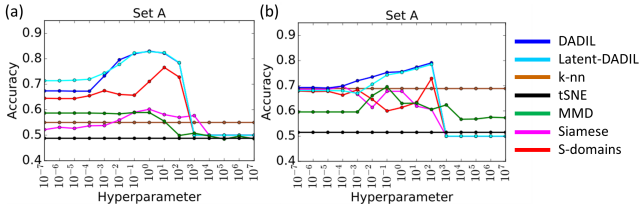

Figure 10: Mean accuracy varying the value of hyperparameter λ_1. (a) "Noise shift: 2.0 and number of samples: 7". (b) "Parallel shift: 2.0 and number of samples: 1".

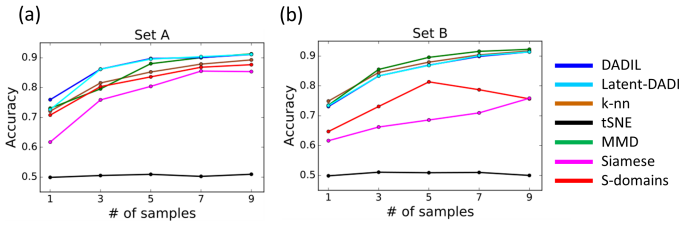

Figure 11: Mean accuracy of Dataset shift for Set A and Set B by varying the number of samples. (a) Set A. (b) Set B.

Table 4: One-sided binomial test for Set A of Dataset shift. Each cell is described as "# of wins / # of loses". Cells marked * are cases where the p value is less than 0.01.

	k-nn	t-SNE	MMD	Siamese	S-domains
DADIL	79/18*	100/0*	72/26*	89/10*	76/17*
Latent-DADIL	80/18*	100/0*	71/28*	78/17*	89/10*

Next, we focus on Set B. As indicated by the results in Figure 11(b), k-nn and MMD performed better than our proposed methods. In our methods, we aim to generate domain-invariant features across domains. If the domain changes greatly, it is difficult to extract an invariant feature and the performance might deteriorate. According to Figure 4, the images labeled "2" and "3" are similar between the MNIST and USPS datasets. In contrast, the shapes of "5" and "6" are more complicated than those of "2" and "3" and the images labeled "5" and "6" are quite different between the MNIST and USPS datasets, intuitively. Therefore, we consider there was a great domain change in Set B. We consider this as the main reason why the performance of the proposed method deteriorated. In contrast, k-nn uses only the target domain data and was not affected by domain shift. For MMD, this method employed different classifiers for the source and target domain and was effective when the domain changed drastically. Although k-nn and MMD were powerful in this experiment, our methods were comparable to the comparison methods. The proposed method can be regarded as an extension of S-domains, but greatly outperformed S-domains. This supports the idea that data augmentation is effective for this problem setting.

According to Figures 6, 7, 8, and 9, k-nn and MMD did not work well, although proposed methods were promising. However, according to Figure 11, S-domains was not effective, although our methods were comparable. This implies that only our proposed methods can be used in many situations.

5 CONCLUSION

In this paper, we proposed a new method DADIL for dealing with domain adaptation with limited data. We focus on the aspect that limited data area common difficulty in domain adaptation and few-shot learning and we utilize data augmentation to extract a domain-invariant feature. We explained that our proposed method is robust against noise from a probabilistic perspective. In experiments using real datasets, our proposed method was effective when noise was injected and parallel shift occurred. Although, the domain shift is quite large, it is difficult to extract the domain-invariant feature and the performance can deteriorate. In the experiment using real datasets, only our proposed method showed stable performance regardless of the experimental settings and can be used in various situations.

REFERENCES

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J.W. Vaughan. 2010. A Theory of Learning from Different Domains. *Machine Learning* 79, 1-2 (May 2010), 151–175.

[2] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.P. Kriegel, B. Schölkopf, and A.J. Smola. 2006. Integrating Structured Biological Data by Kernel Maximum Mean Discrepancy. *Bioinformatics* 22, 14 (July 2006), e49–e57.

[3] S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 539–546 vol. 1.

[4] N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 886–893 vol. 1.

[5] L. Fe-Fei, R. Fergus, and P. Perona. 2003. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings Ninth IEEE International Conference on Computer Vision*. 1134–1141 vol.2.

[6] Y. Ganin and V. Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 1180–1189.

[7] A. Graves, A. r. Mohamed, and G. Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 6645–6649.

[8] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[9] J. Hu, J. Lu, Y. P. Tan, and J. Zhou. 2016. Deep Transfer Metric Learning. *IEEE Transactions on Image Processing* 25, 12 (Dec 2016), 5576–5588.

[10] J. Jiang and C. Zhai. 2007. A Two-stage Approach to Domain Adaptation for Statistical Classifiers. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management (CIKM '07)*. ACM, New York, NY, USA, 401–410. https://doi.org/10.1145/1321440.1321498

[11] D.P. Kingma and L.J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[12] D.P. Kingma, S. Mohamed, D.J. Rezende, and M. Welling. 2014. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3581–3589.

[13] G Koch, R Zemel, and R Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, Vol. 2.

[14] M. Long, Y. Cao, J. Wang, and M.I. Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 97–105.

[15] D. G. Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Vol. 2. 1150–1157 vol.2.

[16] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto. 2017. Few-Shot Adversarial Domain Adaptation. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 6673–6683.

[17] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto. 2017. Unified Deep Supervised Domain Adaptation and Generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 5716–5726.

[18] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. 2015. Semi-supervised Learning with Ladder Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 3546–3554.

[19] Y. Tokozume, Y. Ushiku, and T. Harada. 2018. Between-class Learning for Image Classification. In *Proceedings of the International Conference on Learning Representations (ICLR)*. to appear.

[20] E. Triantafillou, R. Zemel, and R. Urtasun. 2017. Few-Shot Learning Through an Information Retrieval Lens. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 2252–2262.

[21] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. 2015. Simultaneous Deep Transfer Across Domains and Tasks. In *2015 IEEE International Conference on Computer Vision (ICCV)*. 4068–4076.

[22] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. 2014. Deep Domain Confusion: Maximizing for Domain Invariance. *CoRR* abs/1412.3474 (2014). arXiv:1412.3474 http://arxiv.org/abs/1412.3474

[23] L. van der Maaten and G. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

[24] Oriol Vinyals, Charles Blundell, Tim Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 3630–3638.

[25] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. 2011. Class Imbalance, Redux. In *2011 IEEE 11th International Conference on Data Mining*. 754–763.

[26] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. to appear.