applied sciences

MDPI

Article

Extending Partial Domain Adaptation Algorithms to the Open-Set Setting

George Pikramenos [1], Evaggelos Spyrou [2,3,*] and Stavros J. Perantonis [2]

[1] Department of Informatics and Telecommunications, University of Athens, 157 72 Athens, Greece
[2] Institute of Informatics and Telecommunications, National Center for Scientific Research—"Demokritos", 153 41 Athens, Greece
[3] Department of Informatics and Telecommunications, University of Thessaly, 351 00 Lamia, Greece
* Correspondence: espyrou@uth.gr

Abstract: Partial domain adaptation (PDA) is a framework for mitigating the covariate shift problem when target labels are contained in source labels. For this task, adversarial neural network (ANN) methods proposed in the literature have been proven to be flexible and effective. In this work, we adapt such methods to tackle the more general problem of open-set domain adaptation (OSDA), which further allows the existence of target instances with labels outside the source labels. The aim in OSDA is to mitigate the covariate shift problem and to identify target instances with labels outside the source label space. We show that the effectiveness of ANN methods utilized in the PDA setting is hindered by outlier target instances, and we propose an adaptation for effective OSDA.

Keywords: domain adaptation; open-set setting; adversarial neural networks

Citation: Pikramenos, G.; Spyrou, E.; Perantonis, S.J. Extending Partial Domain Adaptation Algorithms to the Open-Set Setting. *Appl. Sci.* **2022**, *12*, 10052. https://doi.org/10.3390/app121910052

Academic Editor: Vincent A. Cicirello

Received: 16 September 2022
Accepted: 29 September 2022
Published: 6 October 2022

1. Introduction

Supervised learning techniques work under the assumption that training and testing datasets are drawn from the same distribution. Thus, traditional techniques require at least some labeled data for the problem at hand so as to obtain a useful model upon training, which, in many cases, is not possible. Domain adaptation (DA) [1] is a technique for generalizing classifiers to different domains. More specifically, given two datasets with the same label space, with one of them labeled (the *source*), a DA algorithm may retrieve labels for the other (the *target*). Note that this does not make any assumption on the underlying distributions of data, yet it is still quite restrictive, since it requires the two datasets to have the same labels.

In practice, we are typically interested in extracting labels from datasets with much larger label spaces than any specific task dataset [2]. To this end, the partial domain adaptation (PDA) [2,3] framework was proposed, in which the assumption is that labels for target data are contained within the larger label space of source data. Of course, the creation of such general-purpose source datasets, which represent the entire label space of any given specific task well, is unrealistic. As an example, some of the largest image datasets currently available [4,5] are comprised of millions of images, but many classes are under-represented.

To address this issue, open-set domain adaptation (OSDA) was proposed [6], essentially combining open-set recognition (OSR) [7] and DA. The aim in OSDA is to automatically extract the information from the source relevant to the target and to identify target instances relevant to the source. That is, a non-empty intersection between the target and source label spaces is assumed, and the goal is to identify source and target instances with labels in the common label subspace. The advantage of identifying such instances is two-fold: It allows us to minimize negative transfer (i.e., the inclusion of source-specific features in the adapted representation) and to identify the subset of target instances that can be reliably labeled through the source.

Note that we cannot hope to predict the label of a target instance belonging to a class not represented in the source label space; however, the advantage of identifying which instances belong to such classes is two-fold. Firstly, it allows us to minimize negative transfer (i.e., the inclusion of source-specific features in the adapted representation) during the adaptation procedure. Secondly, it allows us to identify the subset of target instances that can be reliably labeled using this source.

In this paper, we propose an algorithm for tackling OSDA, which we term the =*Doubly Importance Weighted Adversarial Network* (DIWAN), which was inspired by the distribution reweighing techniques introduced in [8]. Our algorithm bridges the gap between the adversarial neural network (ANN) techniques used in PDA and OSDA by taking into account outlier target instances during adaptation. We prove that our algorithm constructs a representation in the which source and target distributions of transfer-relevant instances (TRI) are aligned. Moreover, we empirically demonstrate that in the open-set setting, DIWAN outperforms non-adapted versions of the DA and PDA algorithms.

The rest of this paper is organized as follows: In Section 2, we present related work and the contribution of DIWAN. Then, in Section 3, we provide the theoretical background and the methodology used by DIWAN. Moreover, the theoretical support for this work is presented in Section 4. An experimental evaluation of DIWAN is presented and discussed in Section 5, and conclusions are drawn in Section 7.

2. Related Work and Contributions

Our algorithm relies on ANNs for DA, as they were successful in classical DA [9,10] and PDA [2,3,8,11,12]. Such adversarial schemes for DA typically include a source model, which is comprised of a classifier and a representer network, a target representer network, and a domain discriminator. The source and target representer networks are embeddings of the source and target data, respectively, into some *latent space*. During the adaptation, the target representer and the domain discriminator antagonize each other, while the source representer and classifier are either fixed with pre-trained weights or trained in a supervised fashion.

At each iteration, the discriminator is presented with data from the source and target domains embedded in latent space and trained to discriminate between the two. The target representer is then trained using reversed discriminator gradients. This process can be shown to minimize the Jensen–Shannon Divergence [13] between the distributions of source and target data in latent space, mitigating the covariate shift problem. This standard scheme is typically augmented by other networks to deal with PDA problems. For example, in [2,3], a collection of domain discriminators were used to reweigh source instances so as to ignore outlier classes in the source domain. In [8], a second domain discriminator was used to reweigh source instances in a similar fashion. Moreover, in [11], both a domain discriminator and classifier information were used to separate outlier source classes. Finally, in [12], a shared-label classifier was used to re-weight instances, and it was trained using information from the source classifier.

Contrary to previous works, we reweigh *both* source and target instances to correct for target outliers. As in [8,11], we re-weight using a second domain discriminator, but our scheme may be modified to use *any* heuristic for TRIs. Moreover, we use the obtained weights to identify target outlier instances by setting a threshold on the target instance weights.

In [6], OSDA was introduced and an algorithm was proposed for tackling OSDA problems based on a constrained integer programming model. A rejection parameter was used to tune open-space risk tolerance. In [14], the authors introduced a method that relied on classifying outlier target instances as "unknown", removing the need for a rejection parameter but making classification harder. The model training utilized techniques introduced in [15] for OSR that generated "unknown" target instances.

The main contribution of this work is the adaptation of existing algorithms for PDA to achieve improved performance in the OSDA setting. Our method mitigates the covariate-

shift problem only for source and target instances that have labels in the common labels' subspace. We further show that our algorithm gives rise to a natural heuristic for identifying target instances that are probably transfer relevant. We perform experiments and empirically validate our approach. In Figure 1, we illustrate the effect of domain adaptation in the presence of source and target outliers.

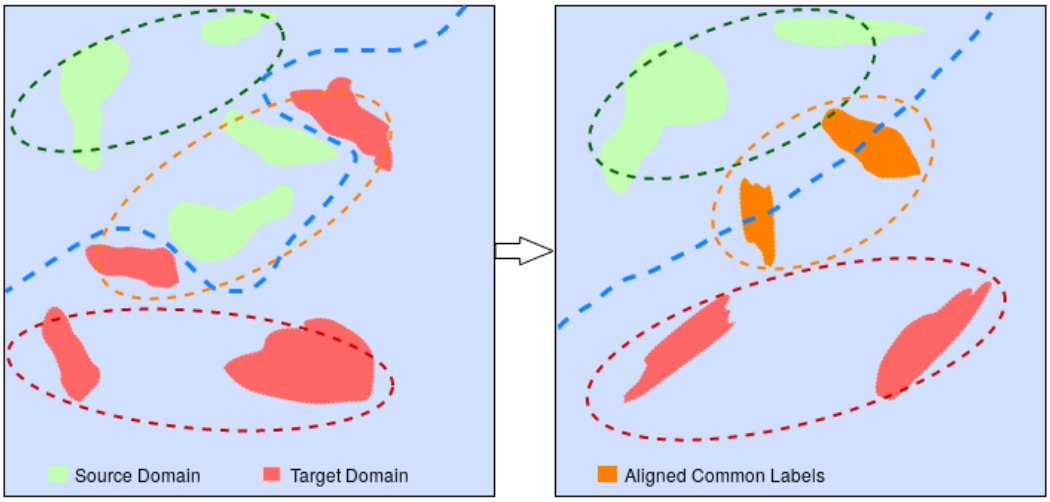

Figure 1. Alignment with source and target outliers. (**Left**): Source and target domain distributions. The blue dashed line denotes a decision boundary *before* adaptation. (**Right**): The effect of adaptation; common labels align, while source/target outlier instances are mainly ignored. This is best viewed in color.

3. Theory and Methodology

The upshot of OSDA is the development of methods for partially labeling an unlabeled dataset by using a model trained on a related dataset. Unlike typical DA, we assume that there exist target instances with labels that are not contained in the source label space, i.e., outlier instances. We are interested in identifying those instances so as to ignore them during the adaptation procedure. The rest of the instances, i.e., transfer-relevant ones, can potentially be reliably labeled through the adaptation procedure.

As in typical DA works, we also assume a machine learning problem described by a domain $\mathcal{D} = \{\mathcal{X}, X\}$ and a task $\mathcal{T} = \{\mathcal{Y}, Y|X\}$, where X is a random variable (the covariates) taking values in $\mathcal{X}$ and Y is a random variable (the labels) taking values in $\mathcal{Y}$. We assume the existence of two problems $(\mathcal{D}_S, \mathcal{T}_S)$ and $(\mathcal{D}_T, \mathcal{T}_T)$, where (a) $\mathcal{X}_S = \mathcal{X}_T$ and (b) $Pr(X_S) \neq Pr(X_T)$. In the OSDA setting, we assume that $\mathcal{Y}_S \cap \mathcal{Y}_T \neq \emptyset$ and

$$P(Y_S = y \mid X_S = x, y \in \mathcal{Y}_S \cap \mathcal{Y}_T) = Pr(Y_T = y \mid X_T = x, y \in \mathcal{Y}_S \cap \mathcal{Y}_T)\,. \tag{1}$$

Our aim is to align the distributions of instances in the source and target domains with labels in $\mathcal{Y}_S \cap \mathcal{Y}_T$. Moreover, we want to identify instances that are *probably* transfer relevant with respect to some confidence measure or heuristic. In the following, we describe DIWAN. Specifically, five neural networks are used: the source and target representer networks M_S, M_T, a *re-weighting network* $\mathcal{W}$, a *constrained* domain discriminator D, and a source classifier C. M_S and C are pre-trained in a standard supervised way on the source dataset. The target representer is typically initialized as M_S. During training, weights are assigned to each instance x. These are:

$$w^S(x) := \frac{1 - \mathcal{W}(M_S(x))}{1 - E[\mathcal{W}(M_S(x))]},\; w^T(x) := \frac{\mathcal{W}(M_T(x))}{E[\mathcal{W}(M_T(x))]}\,, \tag{2}$$

for the source and target, respectively. These choices are explained in the next section. W^S, W^T denote the collection of weights for a fixed M_S, M_T and $\mathcal{W}$. The expectation above is estimated by averages over mini-batches. The objective function used is

$$\min_{\mathcal{W}} \mathcal{L}_{\mathcal{W}} = -E_{x_S \sim X_S}[\log(\mathcal{W}(M_S(x_S)))] - E_{x_T \sim X_T}[\log(1 - \mathcal{W}(M_T(x_T))] \,. \tag{3}$$

For $\mathcal{W}$ and for M_T, D, we have, given W^S, W^T,

$$\max_{M_T} \min_{D} \mathcal{L} = -E_{x_S \sim X_S}[W^S \log(D(M_S(x_S)))] - E_{x_T \sim X_T}[W^T \log(1 - D(M_T(x_T))] \,. \tag{4}$$

Here, $W^S(x), W^T(x)$ is used to emphasize that the weights are computed for all instances before M_T is updated. The pseudocode for DIWAN is provided in Algorithm 1. A visual overview of the five neural networks and the training procedure is illustrated in Figure 2.

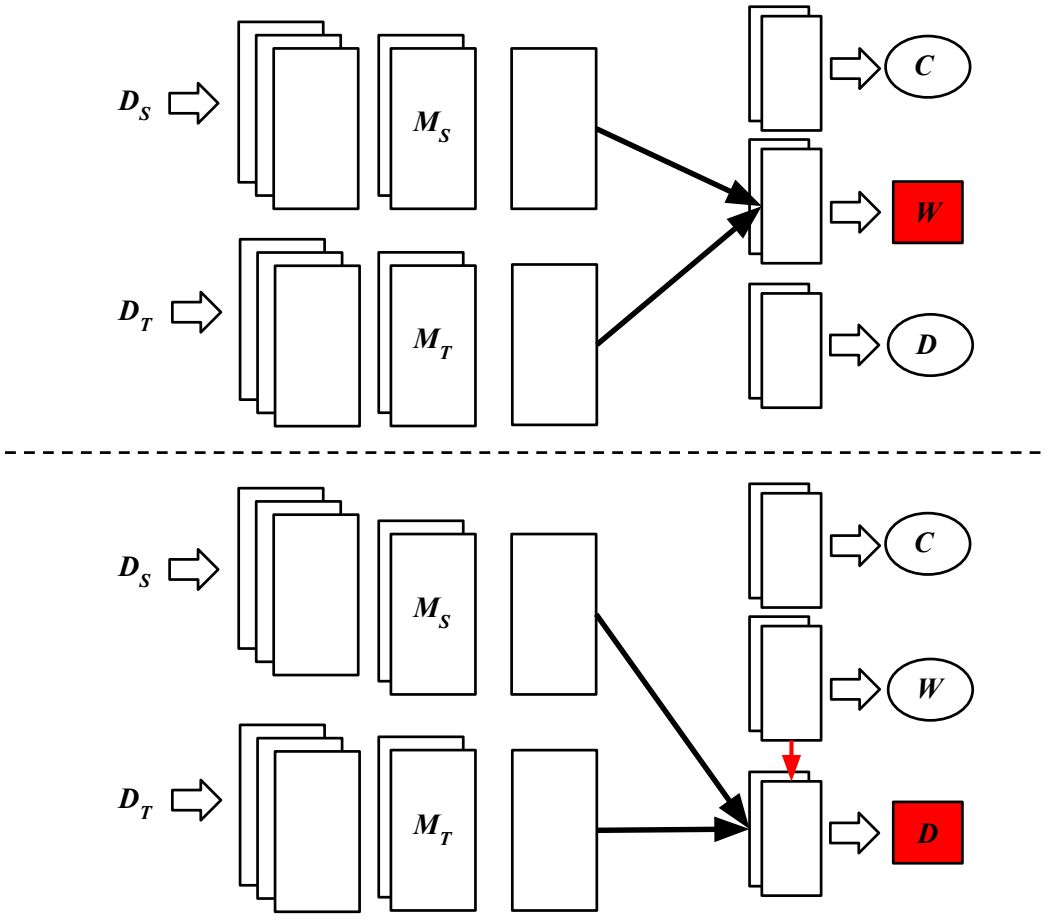

Figure 2. Illustration of the five neural networks involved in the DIWAN algorithm. On the **top**, the re-weighting network is trained from the source and target representer network representations of the source and target data, respectively. On the **bottom**, the domain discriminator is trained using the weights obtained from the re-weighting network.

Algorithm 1: Doubly Importance Weighted Adversarial Network (DIWAN).

Input: A source dataset D_S, a target dataset D_T, a source representer M_S, a source classifier C, and a hyperparameter vector $\vec{\lambda}$.
Output: A target model $C(M_T)(\cdot)$ and a subset $TR_T \subseteq D_T$ of probable transfer-relevant target instances.
INITIALIZE():
$M_T \leftarrow M_S$; $\mathcal{W}, D \leftarrow$ random_initialization();
TRAIN():
for $\vec{\lambda}(MAXITER)$ **do**
  M_T.freeze(); D.unfreeze();
  for $\vec{\lambda}(\mathcal{W}\_iter\_per\_cycle)$ **do**
    $\mathcal{W}$.train_on_source_batch($\lceil 0.5\vec{\lambda}$(batch_size)$\rceil$);
    $\mathcal{W}$.train_on_target_batch($\lfloor 0.5\vec{\lambda}$(batch_size)$\rfloor$);
  end
  $W^S, W^T \leftarrow$ calculate_instance_weights();
  for $\vec{\lambda}(D\_iter\_per\_cycle)$ **do**
    D.train_on_source_batch($\lceil 0.5\vec{\lambda}$(batch_size)$\rceil$);
    D.train_on_target_batch($\lfloor 0.5\vec{\lambda}$(batch_size)$\rfloor$);
  end
  D.freeze(); M_T.unfreeze();
  $D \circ M_T$.train_on_target_batch($\vec{\lambda}$(batch_size));
end
return: $C \circ M_T$, W^T. Alternatively, indices of W^T with value $> \vec{\lambda}$(cutoff_threshold).

4. Analysis

We now provide theoretical support regarding the design of the proposed algorithm.

Proposition 1. *Let p_1 and p_2 be distributions over some bounded space $\mathcal{X} \subseteq \mathbb{R}^n$ and let $F : \mathcal{X} \to \mathbb{R}$ be some real-valued function. We define*

$$\tilde{\mathcal{L}}[F, p_1, p_2] = -E_{x\sim p_1}[\log(F(x))] - E_{x\sim p_2}[\log(1 - F(x))] \,. \tag{5}$$

Then,

$$F^*_{p_1,p_2}(x) = \arg\min_F \tilde{\mathcal{L}}[F, p_1, p_2] = \frac{p_1(x)}{p_1(x) + p_2(x)} \,. \tag{6}$$

Proof. We will use a variational calculus argument. We denote $F' = \nabla_x F = [\partial F/\partial x_1, \ldots, \partial F/\partial x_n]$, define the functional

$$\hat{F}_{p_1,p_2}[F(x), F'(x), x] = -p_1(x)\log(F(x)) - p_2(x)\log(1 - F(x)),$$

and note that $\tilde{\mathcal{L}}[F, p_1, p_2] = \int_{\mathcal{X}} \hat{F}[F(x), F'(x), x]\, dx$. We solve the Euler–Lagrange equations, yielding

$$\frac{\partial \hat{F}}{\partial F} = 0 \Leftrightarrow \frac{p_1(x)}{F^*(x)} = \frac{p_2(x)}{1 - F^*(x)} \Leftrightarrow F^*(x) = \frac{p_1(x)}{p_2(x) + p_1(x)}$$

□

Corollary 1. *Let p_S, p_T have bounded support. Then,*

$$\mathcal{W}^*(x) = \arg\min_{\mathcal{W}} \mathcal{L}_W = \frac{p_S(x)}{p_S(x) + p_T(x)} \tag{7}$$

Lemma 1. *Let p_1, p_2 have bounded support. Let w^1, w^2 be non-negative weight functions that satisfy $E[w^1] = E[w^2] = 1$. Then, $\tilde{\mathcal{L}}[F, w^1 p_1, w^2 p_2]$ is well defined.*

Proof. Let $P^1(x) = w^1(x)p_1(x)$ and $P^2(x) = w^2(x)p_2(x)$. Since $\int_{\mathcal{X}} dP^1(x) = E[w^1] = \int_{\mathcal{X}} dP^2(x) = E[w^2] = 1$ and $w^1, w^2 \geq 0$, these are valid probability densities. Furthermore, for $i = 1, 2$, the support of P^i is contained in the support of p_i. □

Corollary 2. *Let p_S, p_T have bounded support. Let w^S, w^T be as in (2) and let D^* be the minimizer of $\mathcal{L}$ for a fixed M_T. Then,*

$$D^*(x) = \frac{w^S(x)p_S(x)}{w^S(x)p_S(x) + w^T(x)p_T(x)} \tag{8}$$

Proposition 2.

$$-\min_D \mathcal{L} \equiv -\mathcal{L}[D^*] = -\log(4) + JSD(P^{w^S} || P^{w^T}) \tag{9}$$

Proof. We have

$$\begin{aligned} &-\min_D \mathcal{L} = E_{p_S}[w^S(x)\log(D^*)] + E_{p_T}[w^T(x)\log(1 - D^*)] \overset{(Corrolary1)}{=} \\ &E_{P^{w^S}}[\log(\frac{P^{w^S}}{P^{w^T} + P^{w^S}})] + E_{P^{w^T}}[\log(\frac{P^{w^T}}{P^{w^T} + P^{w^S}})] = \\ &-\log(4) + JDS(P^{w^S} || P^{w^T}) \end{aligned} \tag{10}$$

□

Essentially, we see that OSDA can be cast as an alternating optimization problem. Moreover, we see that a heuristic for identifying probable TRIs can define an algorithm for OSDA by simply normalizing its heuristic scores by using them as weights in $\mathcal{L}$. Intuitively, we aim to down-weight the outliers in the source and target domains. We use an adapted heuristic similar to the one used in [8,11] for re-weighting. The idea behind it is that TRIs will lie near the decision boundary of a "good" domain discriminator, and outlier classes will be far from it; a good domain discriminator will easily distinguish that an outlier source (target) instance belongs to the source (target) domain. In particular, source instances x_S for which $\mathcal{W}^*(x_S) \approx 1$ are likely to be outliers, since $\mathcal{W}^*$ can easily tell that they originate from the source domain. Thus, they should be down-weighted. Similarly, for target instances x_T, $\mathcal{W}^*(x_T) \approx 0$.

5. Experiments

For the empirical verification of our approach, we performed three different experiments. In the first experiment, our aim was to show that the adversarial neural network techniques commonly used in DA are prone to negative transfer when applied in settings with target outlier classes. In particular, we demonstrated that if adaptation is performed using the ADDA algorithm [9], the accuracy on transfer-relevant instances will be lower when target outliers are present.

In the second experiment, our goal was to demonstrate that DIWAN mitigated this negative transfer and to compare the results obtained with ADDA and IWPDA [8]. To this end, we used only transfer-relevant instances to evaluate the methods. Clearly, target outlier instances were misclassified by all three methods and were thus ignored. However,

DIWAN offers a heuristic for identifying transfer-relevant instances, and this was evaluated in the final experiment.

Lastly, we demonstrated that we could select a threshold for the assigned weights of target instances after DIWAN was run, such that target instances with weights above this threshold were very likely to be transfer relevant. In particular, we calculated the accuracy of the target model obtained through DIWAN over all data instances with weights above certain thresholds, and we showed that there were threshold choices that could be selected through cross-validation, yielding high-quality classification.

5.1. Dataset and Task Description

Our experiments were conducted for four different DA tasks for images. These were image obstruction, rotation, displacement, and re-scaling. The samples used within these tasks are depicted in Figure 3. Our dataset was comprised of the MNIST [16] and USPS [17] handwritten digit datasets, as in [8,9]. The former contained 28×28 pixel gray-scale images, while the latter contained 16×16 pixel gray-scale images. Both datasets contained 10 classes corresponding to each of the 10 digits. The USPS images were padded to obtain a homogeneous DA problem. That is, we appended zeros on all four edges of each USPS image so that it became a 28×28 pixel image, where the original image occupied the central pixels. In all of our experiments, MNIST was used as a source dataset and USPS was used to generate a synthetic dataset for each of the tasks.

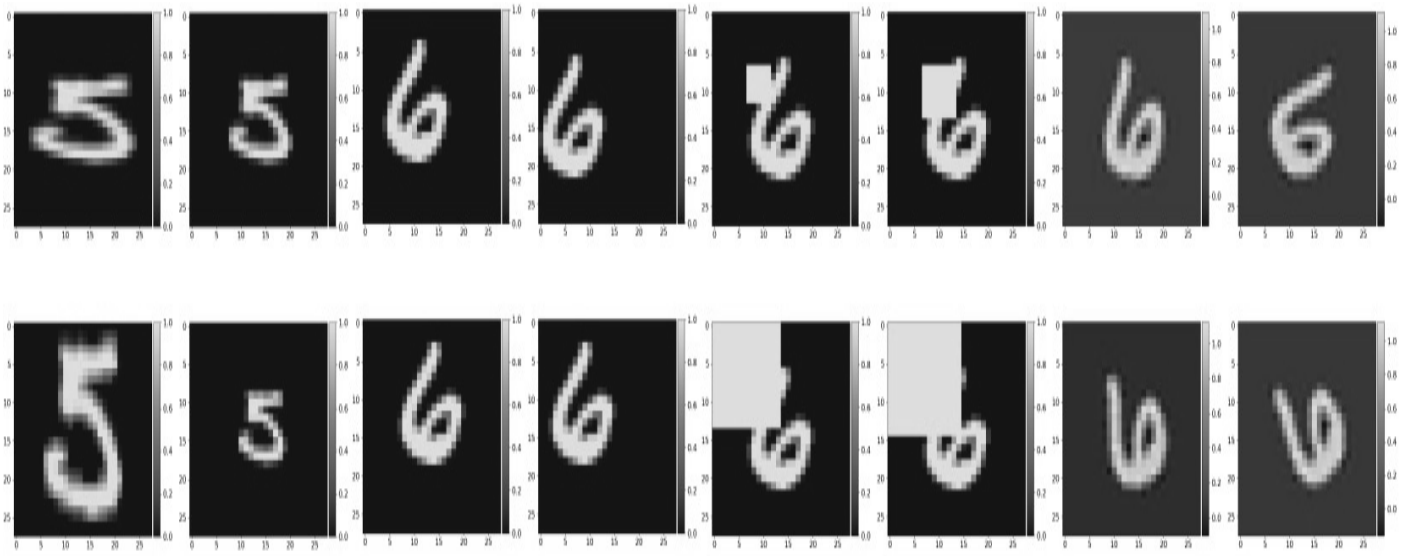

Figure 3. Sample images from the obstruction, rotation, rescaling, and displacement tasks with different parameters.

For the obstruction task, we set a 14 pixel area starting from the top-left corner in each image in the USPS dataset to have an intensity of 1. Similarly, each image in USPS was rotated counter-clockwise by 40° for the rotation task. For the re-scaling task, the image was shrunk by a factor of 0.6 on both the vertical and horizontal axes, and then it was re-centered. Finally, for the displacement task, each image was displaced by five pixels to the left and three pixels upwards. The covariate shift introduced in all four tasks was systematic; the same transformation was applied on all USPS images.

5.2. Experiments and Methodology

We started by running experiments to test whether outlier target labels were a source of negative transfer in DA. ADDA was run for each task for 50 trials on a full DA scenario with five source labels. The labels were chosen randomly for each task and trial. We then repeated for a scenario where there were 10 target labels and five source labels, and the accuracy was only measured on transfer-relevant instances. The source network was kept the same for each scenario, and the hyperparameters were tuned to optimize the performance for each task. A convolutional architecture was used for the source representer network. We performed this experiment for all four tasks and plotted average accuracies $\pm 2\sigma$ against the number of iterations.

For the second experiment, for each task, we varied the number of outlier instances in both the source and target domains, and for each combination, we reported the mean results of four algorithms on ten trials. The algorithms that we used were ADDA, IWPDA, DIWAN, and the source model (without any adaptations). For each trial, the target and source labels were selected randomly. Note also that the number of source labels was always equal to 5.

For our final experiment, recall that once training was finished, we could use $\mathcal{W}^*$ to calculate (2), where the expectation was replaced by the empirical average over the entire target dataset. We present histograms of weights for each task, where representative setups of labels are chosen. Our aim was to illustrate that the weights tended to be larger for transfer-relevant instances. We then went on to compute the adapted model accuracies after a threshold was imposed. We kept the models from experiment 2 for the case in which there were 10 target labels. The average over the ten trials is presented for each threshold.

For each task, we used four different thresholds, and for each, we computed the accuracy of our model on the subset of the target dataset that we obtained after the cut-off procedure. We further gave the percentage of total transfer-relevant target instances that remained in the dataset after applying the threshold and the percentage of transfer-relevant instances in the remaining dataset.

6. Results and Discussion

6.1. Evidence for Negative Transfer

There was a clear indication that the target outlier classes had a negative impact on the adaptation procedure and could have led to higher instability during training, as indicated by the higher variance obtained for the displacement and re-scaling tasks. The results are presented in Figure 4.

6.2. Effectiveness of DIWAN

The results for our second experiment indicated that full DA would often fail in the OSDA setting. This was observed across all tasks. The PDA model performed better than ADDA, but still performed worse than DIWAN. The results are summarized in Tables 1 and 2. We further noted that in the partial DA setting (when the target labels were all common labels), DIWAN performed comparably well with respect to IWPDA, while ADDA could completely fail if there were too many outlier source labels.

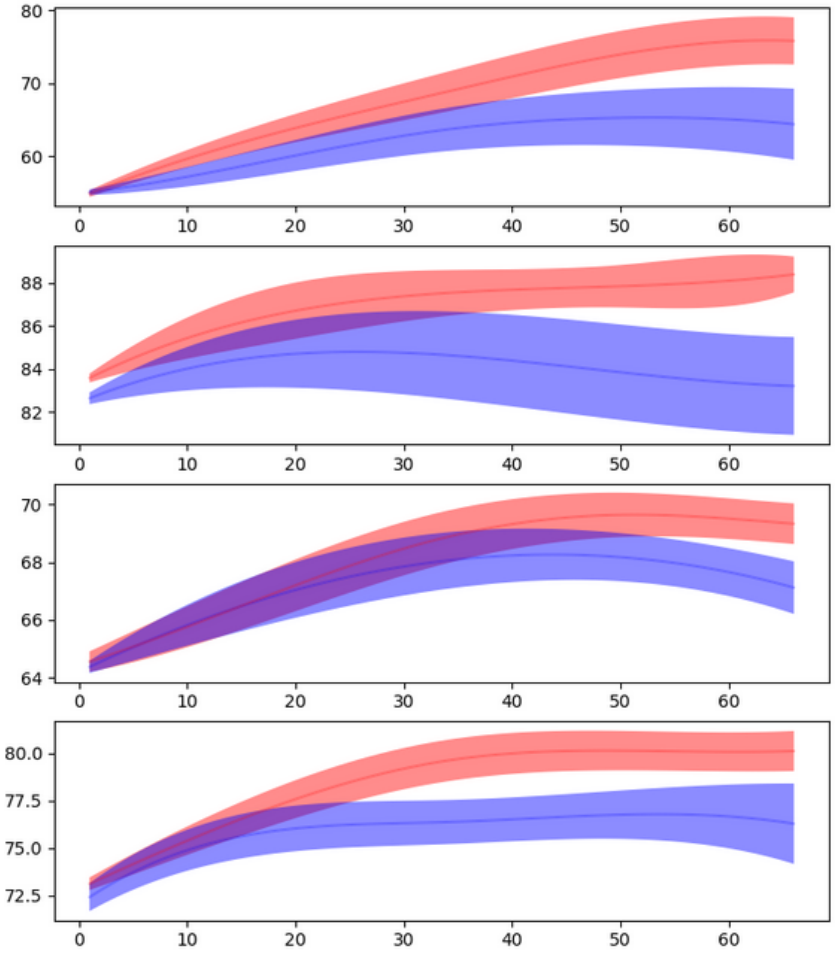

Figure 4. From top to bottom: Plots of accuracy vs. number of iterations for the obstruction, displacement, rotation, and re-scaling tasks. The mean over 50 trials is drawn with the shaded region, representing $\pm 2\sigma$ calculated empirically. The source domain included five random classes from the source domain, which were different for each trial. The red-shaded region illustrates the mean accuracy for the experiment where no target outliers were present. The blue-shaded region illustrates the mean accuracy when five outlier labels were added to the target domain. The figure is best viewed in color.

Table 1. The results of Experiment 2 for the obstruction and displacement cases. "T" is the number of target labels and "C" is the number of target labels common to the source. "A" is the performance of ADDA [9], "I" is the performance of IWPDA [8], "D" is the performance of DIWAN, and "S" is the performance of the source model without adaptations. The results are the average accuracy percentages over 10 trials. Best accuracy indicated in bold.

		Obstruction				**Displacement**			
T	**C**	**A**	**I**	**D**	**S**	**A**	**I**	**D**	**S**
3	3	90.1	**93.2**	91.4	34.8	59.8	**85.7**	84.6	80.6
4	4	80.1	**83.0**	78.6	53.2	74.0	**85.3**	85.1	66.1
4	2	57.3	85.6	**87.8**	57.4	60.2	60.0	**65.2**	56.4
5	2	3.5	70.7	**88.4**	7.8	55.3	60.1	**78.9**	72.2
6	2	60.9	78.5	**80.1**	49.2	79.9	79.2	**90.5**	59.2
4	3	33.6	63.9	**64.5**	29.3	83.6	88.3	**92.7**	69.9
5	3	38.9	**74.9**	**74.8**	41.5	34.6	35.2	**55.8**	35.3

Table 1. *Cont.*

		Obstruction				Displacement			
T	**C**	**A**	**I**	**D**	**S**	**A**	**I**	**D**	**S**
6	3	67.7	69.5	**70.9**	61.6	39.0	41.4	**59.8**	39.0
7	3	70.2	74.6	**90.0**	68.6	34.0	36.1	**42.6**	35.3
5	4	75.2	77.3	**85.6**	53.2	79.2	81.3	**82.0**	74.4
6	4	79.9	77.5	**81.8**	69.7	39.7	41.6	**66.8**	38.7
10	5	63.1	67.3	**76.2**	57.0	66.6	73.1	**82.3**	70.7

Table 2. Results of Experiment 2 for the rotation and rescaling cases. "T" is the number of target labels and "C" is the number of target labels common to the source. "A" is the performance of ADDA [9], "I" is the performance of IWPDA [8], "D" is the performance of DIWAN, and "S" is the performance of the source model without adaptation. The results are the average accuracy percentages over 10 trials. Best accuracy indicated in bold.

		Rotation				Rescaling			
T	**C**	**A**	**I**	**D**	**S**	**A**	**I**	**D**	**S**
3	3	**66.9**	66.7	64.8	62.1	77.3	**81.3**	81.0	60.6
4	4	73.3	**74.9**	73.9	68.8	77.3	**79.4**	77.9	62.5
4	2	**70.0**	65.1	68.9	65.0	55.8	67.5	**82.7**	69.1
5	2	74.0	73.8	**74.5**	72.8	81.0	74.5	**85.3**	65.1
6	2	66.1	60.1	**67.2**	49.3	72.3	83.7	**94.6**	83.4
4	3	76.3	77.0	**77.8**	73.6	66.0	85.2	**90.0**	79.2
5	3	71.2	**68.4**	**71.7**	62.2	59.0	87.9	**93.8**	79.2
6	3	84.8	83.2	**84.9**	73.3	44.7	47.0	**53.3**	46.5
7	3	75.4	77.2	**81.5**	62.8	49.1	56.4	**58.1**	54.1
5	4	70.2	70.4	**70.8**	69.1	53.0	55.0	**56.8**	55.2
6	4	75.6	77.0	**78.9**	68.8	70.1	76.2	**83.0**	74.6
10	5	61.3	65.8	**67.0**	65.8	74.2	77.8	**84.5**	68.6

6.3. Testing Cut-Off Thresholds

Finally, we present our results for using thresholds on the obtained weights to identify probable transfer-relevant instances. Histograms of the weights are presented in Figures 5–8. The histograms suggest that the heuristic that we utilized was useful; the transfer-relevant instances (depicted in red) tended to have above-average weights, while the outlier instances (in blue) tended to have below-average weights. The separation was much more apparent in certain tasks (e.g., obstruction). We then demonstrated that a threshold on target weights could be used to select only instances that could be reliably classified from the adapted model. In practice, the threshold should be selected by using cross-validation to fit the needs of the problem at hand; in some cases, we may not mind a few mistakes if we can label more instances correctly, and in other cases, we may only want to produce labels with very high certainty.

In addition, note that, especially when increasing the threshold above 1.0, many transfer-relevant instances were removed from the dataset, as it may be seen in Table 3. These, however, were transfer-relevant instances that did not benefit much from the adaptation. This was seen because the accuracies on subsets resulting from higher thresholds were better than the overall accuracy on transfer-relevant instances.

For example, for the obstruction task, the adapted model had an accuracy of 76.22% on transfer-relevant instances, but after a threshold of 1.5 was imposed, the accuracy went up to 99.22%. For the rescaling task, the accuracy on the dataset with a threshold of 1.5 reaches 100%. To sum up, using our heuristic not only yielded transfer-relevant instances, but also yielded the "best" transfer-relevant instances in the sense that they were more likely to have benefited from the adaptation. This was not observed for the rotation task, which raises an interesting question that may be investigated in future work.

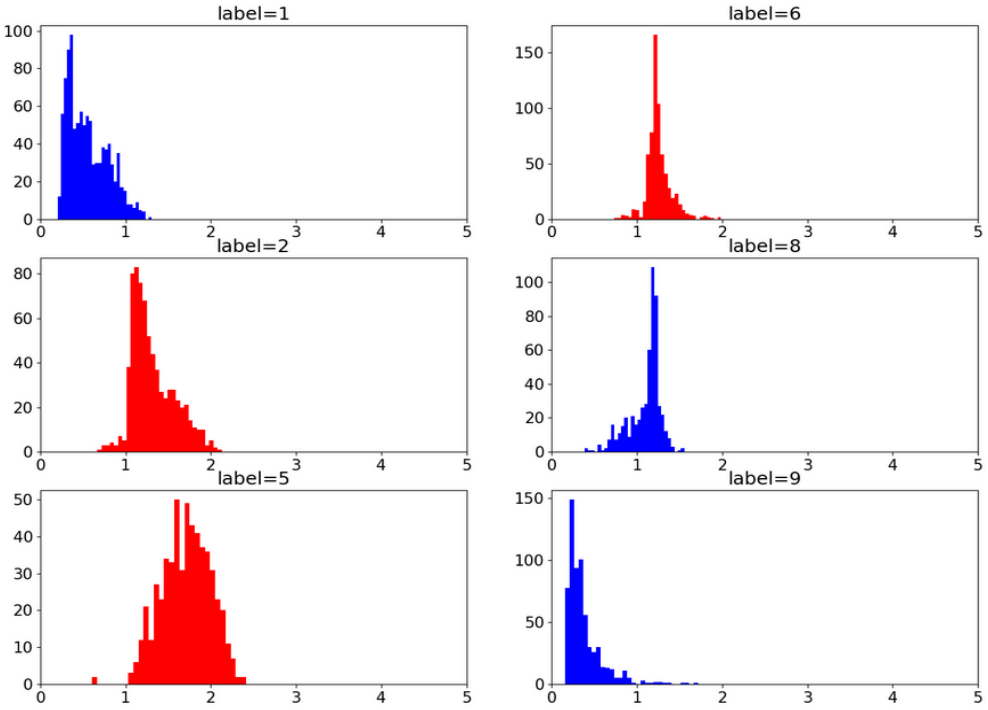

Figure 5. Obstruction task: Transfer-relevant (outlier) target instance histograms are depicted in red (blue). The figure is best viewed in color.

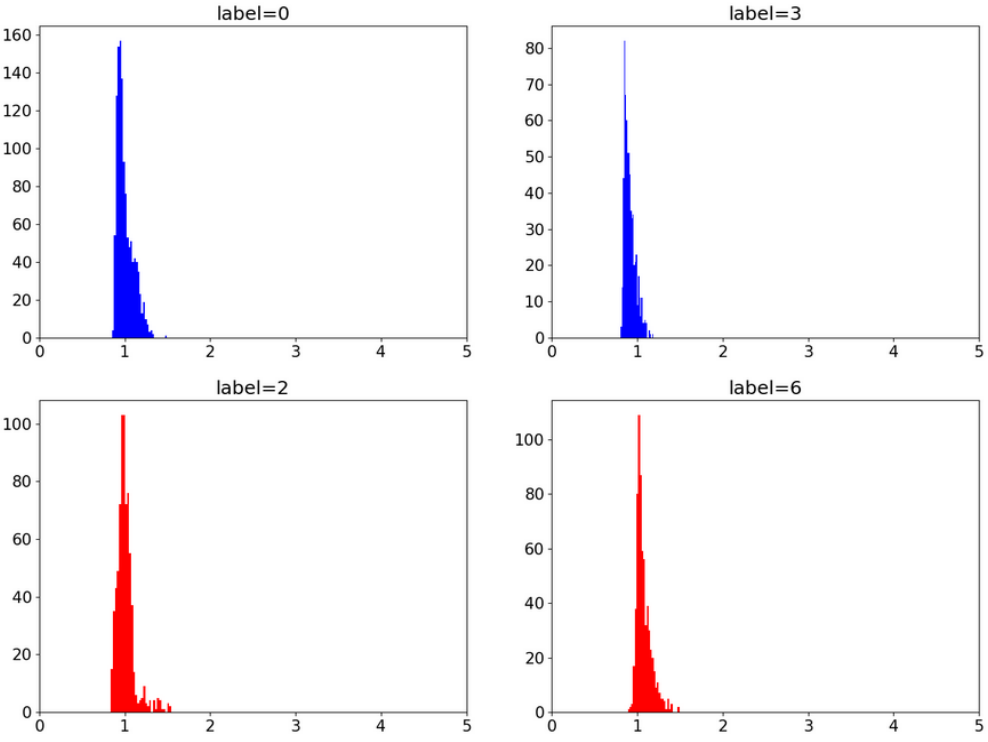

Figure 6. Rotation task: Transfer-relevant (outlier) target instance histograms are depicted in red (blue). The figure is best viewed in color.

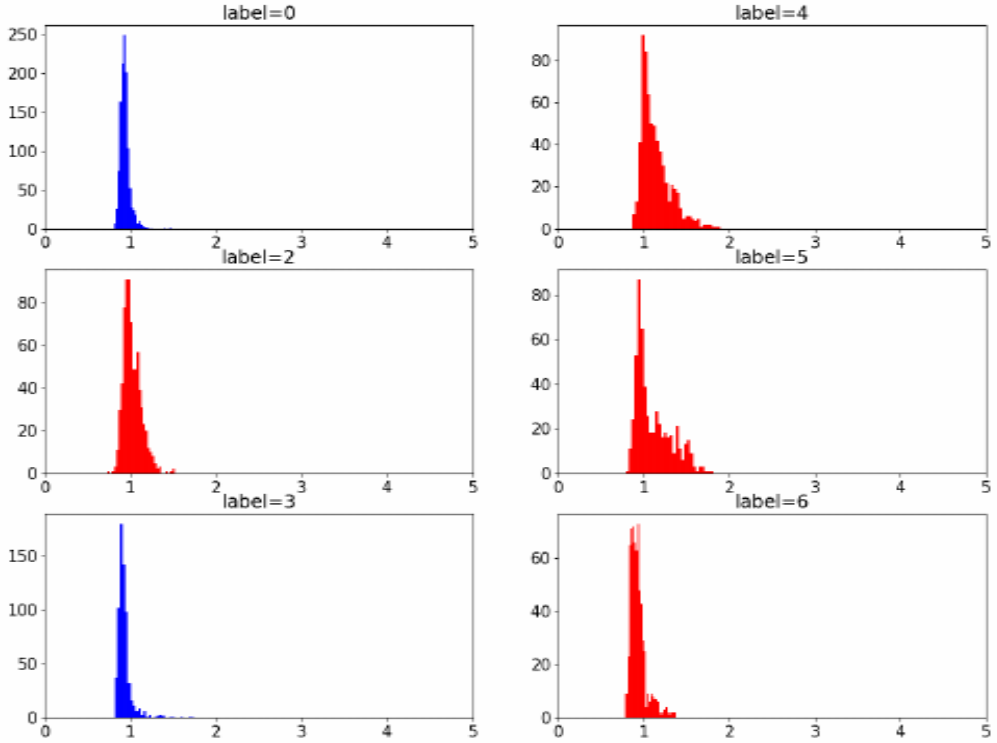

Figure 7. Rescaling task: Transfer-relevant (outlier) target instance histograms are depicted in red (blue). The figure is best viewed in color.

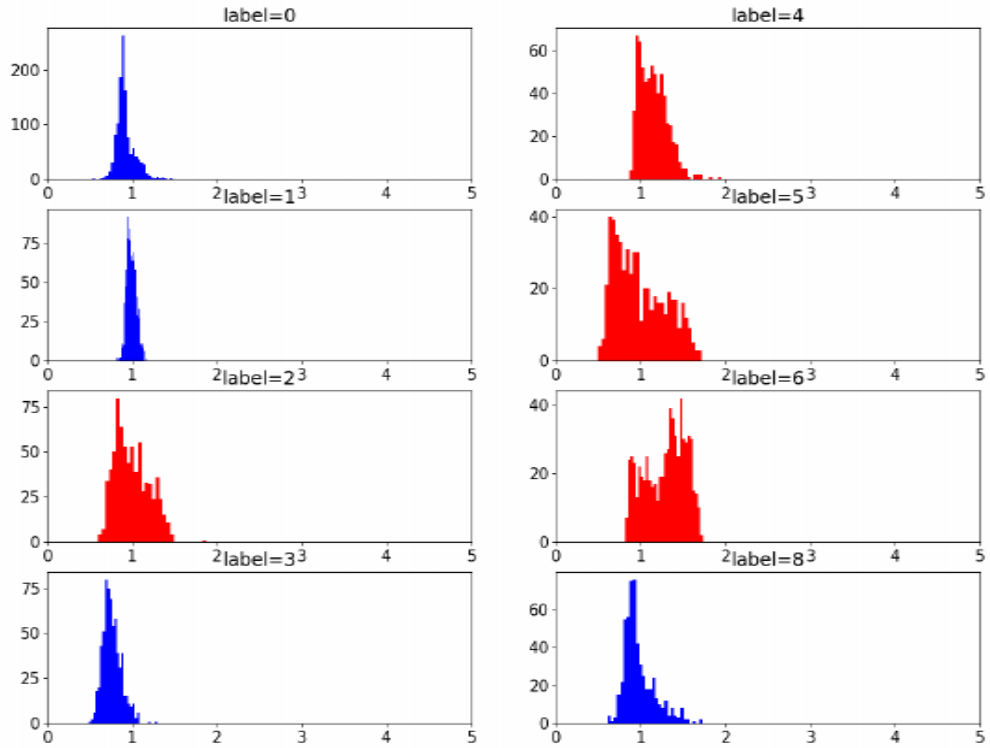

Figure 8. Displacement task: Transfer-relevant (outlier) target instance histograms are depicted in red (blue). The figure is best viewed in color.

Table 3. TTR denotes the percentage of total transfer-relevant instances captured by the threshold. TR denotes the percentage of TRIs in the dataset after the threshold is applied.

Case	Threshold	TTR	TR	#Target Instances	Accuracy
Obstruction	0	100.00	47.10	4142	36.12
	0.5	100.00	62.53	3120	47.95
	1	97.54	80.13	2375	62.95
	1.5	32.39	99.22	637	98.43
Rescaling	0	100.00	58.42	4455	37.15
	1	53.51	78.39	1777	67.36
	1.25	32.85	98.16	871	92.66
	1.5	20.86	100.00	543	100.00
Rotation	0	100.00	42.96	3247	42.22
	1	55.84	59.19	1316	38.98
	1.25	21.57	86.24	349	50.43
	1.5	5.96	98.96	97	35.16
Displacement	0	100.00	43.37	6002	49.92
	1	61.93	65.90	2446	60.95
	1.25	30.43	92.63	855	84.21
	1.5	7.99	96.74	216	91.16

7. Conclusions

We have extended the popular ANN algorithms used in the DA and PDA settings to work on the OSDA setting. We showed how to utilize heuristics for transfer relevance in order to obtain algorithms for the constrained latent distribution alignment problem and cut-off thresholds for identifying subsets of the target dataset that can be reliably labeled. In particular, we introduced the DIWAN algorithm, which uses an adapted popular heuristic from the literature. Extensive experimentation illustrated the benefits of our method in the OSDA setting. This methodology can be very effective for improving the applicability of machine learning models to real-world data. It could be used for image processing that is generated in an online manner in uncontrolled environments to detect objects or events of interest. Examples include social media content, assisted living, surveillance data, and traffic monitoring. The proposed methodology can also be extended in the future to utilize other weighting methods for target domain classes.

Author Contributions: Conceptualization, G.P.; methodology, G.P., E.S. and S.J.P.; validation, G.P.; writing—original draft preparation, G.P. and E.S.; writing—review and editing, G.P., E.S. and S.J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, under the call RESEARCH–CREATE–INNOVATE (project code: T1EDK-02070).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADDA	Adversarial Discriminative Domain Adaptation
ANN	Adversarial Neural Networks
DA	Domain Adaptation
DIWAN	Doubly Importance Weighted Adversarial Network
IWPDA	Importance Weighted Partial Domain Adaptation
MNIST dataset	Modified National Institute of Standards and Technology dataset
OSDA	Open-Set Domain Adaptation
OSR	Open-Set Recognition
PDA	Partial Domain Adaptation
TRI	Transfer-Relevant Instance
USPS dataset	United States Postal Service dataset

References

1. Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; Vaughan, J.W. A theory of learning from different domains. *Mach. Learn.* **2010**, *79*, 151–175. [CrossRef]
2. Cao, Z.; Long, M.; Wang, J.; Jordan, M.I. Partial transfer learning with selective adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
3. Cao, Z.; Ma, L.; Long, M.; Wang, J. Partial adversarial domain adaptation. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018.
4. Wu, B.; Chen, W.; Fan, Y.; Zhang, Y.; Hou, J.; Liu, J.; Zhang, T. Tencent ml-images: A large-scale multi-label image database for visual representation learning. *IEEE Access* **2019**, *7*, 172683–172693. [CrossRef]
5. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009.
6. Panareda, B.P.; Gall, J. Open set domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017.
7. Geng, C.; Huang, S.J.; Chen, S. Recent advances in open set recognition: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3614–3631. [CrossRef] [PubMed]
8. Zhang, J.; Ding, Z.; Li, W.; Ogunbona, P. Importance weighted adversarial nets for partial domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
9. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial discriminative domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 22–25 July 2017.
10. Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M. Domain-adversarial neural networks. *arXiv* **2014**, arXiv:1412.4446.
11. Cao, Z.; You, K.; Long, M.; Wang, J.; Yang, Q. Learning to transfer examples for partial domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
12. Hu, J.; Tuo, H.; Wang, C.; Qiao, L.; Zhong, H.; Jing, Z. Multi-Weight Partial Domain Adaptation. In Proceedings of the BMVC 2019, Cardiff, UK, 9–12 September 2019; p. 5.
13. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; Wiley Series in Telecommunications and Signal Processing; John Wiley Sons: Hoboken, NJ, USA, 2006.
14. Saito, K.; Yamamoto, S.; Ushiku, Y.; Harada, T. Open set domain adaptation by backpropagation. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018.
15. Ge, Z.; Demyanov, S.; Chen, Z.; Garnavi, R. Generative openmax for multi-class open set classification. *arXiv* **2017**, arXiv:1707.07418.
16. Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
17. Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning (Volume 1, No. 10)*; Springer: New York, NY, USA, 2001.