Received: 13 July 2022 | Revised: 18 July 2022 | Accepted: 24 August 2022 | Published online: 8 October 2022

Journal of Computational and Cognitive Engineering
2022, Vol. 1(4) 181–186
DOI: 10.47852/bonviewJCCE2202324
BON VIEW PUBLISHING

RESEARCH ARTICLE

Coupling Adversarial Learning with Selective Voting Strategy for Distribution Alignment in Partial Domain Adaptation

Sandipan Choudhuri[1,*], **Hemanth Venkateswara**[1] **and Arunabha Sen**[1]

[1]*Arizona State University, USA*

Abstract: In contrast to a standard closed-set domain adaptation task, partial domain adaptation setup caters to a realistic scenario by relaxing the identical label set assumption. The fact of source label set subsuming the target label set, however, introduces few additional obstacles as training on private source category samples thwart relevant knowledge transfer and mislead the classification process. To mitigate these issues, we devise a mechanism for strategic selection of highly confident target samples essential for the estimation of class-importance weights. Furthermore, we capture class-discriminative and domain-invariant features by coupling the process of achieving compact and distinct class distributions with an adversarial objective. Experimental findings over numerous cross-domain classification tasks demonstrate the potential of the proposed technique to deliver superior and comparable accuracy over existing methods.

Keywords: partial domain adaptation, domain adaptation, adversarial learning, class-distribution alignment

1. Introduction

The effectiveness of deep-classification networks is conditioned on the accessibility of annotated data which is, often, not readily available. Domain adaptation (*DA*) methods (Ganin et al., 2016; Li et al., 2020) mitigate this annotation demand by transferring knowledge from an already-labeled dataset in a related domain. A majority of the existing *DA* approaches (Ganin & Lempitsky 2015; Ganin et al., 2016; Li et al., 2020) assumes that labeled and unlabeled domains share identical label set. In practice, however, obtaining a suitably labeled domain (source) with such a tight restriction on the label space is challenging. Partial domain adaptation (*pda*) (Cao et al., 2018a) relaxes this constraint by utilizing a large-scale source domain data that subsumes the target label space.

Such a relaxation introduces the issue of superfluous information transfer from samples private to source domain (*negative transfer*) (Cao et al., 2018a; Cao et al., 2019), thereby thwarting the classification performance. Existing solutions (Cao et al., 2018a, 2018b, 2019; Zhang et al., 2018a) have attempted to down-weight such samples by re-weighting them or by performing category-level aggregation of all target sample predictions, for estimating class-importance. However, poor classification at the initial stages of training may induce severe errors in such estimations, thereby misleading the optimization process. In this work, we address the negative transfer problem by strategically employing a subset of the target samples for adaptive estimation of class-importance weights.

Prior works (Cao et al., 2018a, 2018b, 2019; Choudhuri et al., 2020; Zhang et al., 2018a) have focused on the effectiveness of aligning the latent distributions of source and target via an adversarial objective. Achieving domain invariance is, however, a necessary but not a sufficient condition for improving target classification performance; mitigating the conditional distribution mismatch across two domains is equally essential (Jing et al., 2020). Citing this, we have designed a strategy to extract both *category-discriminative* and *domain-invariant features* by coupling an adversarial objective with a process aimed for obtaining more compact class distributions.

2. Related Work

A series of works, in recent years, have explored the effectiveness of deep-neural architectures for mitigating domain shift and transferring underlying knowledge across domains in domain adaptation tasks (Hoffman et al., 2014; Oquab et al., 2014; Yosinski et al., 2014). A different line of work (Zhang et al., 2018b) has utilized high-order statistical properties, like maximum-mean discrepancy, to align the data distribution across domains and successfully eliminate domain discrepancy. Authors

*Corresponding author: Sandipan Choudhuri, Arizona State University, USA. Email: s.choudhuri@asu.edu

in Ganin et al. (2016), Li et al. (2019) have utilized adversarial learning to design a mini-max game for obtaining domain-invariant features. Unfortunately, they operate on a restrictive *uda* environment and does not scale satisfactorily in a *pda* setup. To tackle *pda* tasks, Selective Adversarial Network (SAN) (Cao et al., 2018a) utilizes multiple adversarial networks to down-weight private source category samples. Cao et al. (2018b) extend this concept to formulate class-importance weighting using all target samples. Importance-Weighted Adversarial Nets (IWAN) (Zhang et al., 2018a) employ an auxiliary domain discriminator to determine the membership of a source sample in the target domain. Example Transfer Network (ETN) (Cao et al., 2019) operates on similar lines, by exploiting discriminative information as the transferability quantification for source domain samples. Albeit surpassing the performance of standard domain adaptation approaches, poor classification at the initial stages of training of these models may induce severe errors in private source category estimations, thereby misleading the classification task.

3. Methodology

3.1. Problem settings

A typical *Unsupervised Domain-Adaptation* (*uda*) setup assumes that the source s and target t samples are drawn from different probability distributions (Sugiyama et al., 2007). As witnessed in such an environment, we are furnished with a source dataset $\mathcal{D}_s = \{< x_s^i, y_s^i >\}_{i=1}^{|\mathcal{D}_t|}$ of $|\mathcal{D}_s|$ labeled points, sampled from a distribution $\mathbb{P}_s$, and an unlabeled dataset $\mathcal{D}_t = \{x_t^j\}_{j=1}^{|\mathcal{D}_t|}$ of $|\mathcal{D}_t|$ samples, representing target t, drawn from distribution $\mathbb{P}_t$ ($x_s{}^i, x_t^j \in \mathbb{R}^d$ and $\mathbb{P}_s \neq \mathbb{P}_t$). Since target class label information is unavailable during adaptation, the *closed-set* variant assumes samples in $\mathcal{D}_s$ and $\mathcal{D}_t$ categorized into classes from known label sets C_s and C_t respectively, where $C_s = C_t$. *Partial DA (pda)* generalizes this characterization and addresses a realistic scenario by alleviating the constraint of identical label space assumptions between the two domains (i.e., $C_t \subseteq C_s$).

We aim to design a classifier hypothesis $h : x_t \rightarrow y_t$ ($h \in \mathcal{H}$) that minimizes the target classification risk under a *PDA* setup. This is achieved by leveraging source domain supervision to capture class-semantic information, along with minimizing misalignment due to negative transfer from samples $< x_s, y_s > \in \mathcal{D}_{\overline{s}}$ in label space $C_s - C_t$, where $\mathcal{D}_{\bar{s}} = \{< x_s, y_s > | < x_s, y_s > \in \mathcal{D}_s, y_s \in C_s - C_t\}$.

3.2. Proposed approach

In this section, we present the proposed framework that couples adversarial learning with a selective voting strategy to mitigate domain discrepancy, while achieving class-distribution alignment.

3.2.1. Domain-invariant feature extraction with adversarial learning

The idea of coupling adversarial learning with domain-invariant feature extraction has been extensively researched in domain adaptation problems (Cao et al., 2018a; Ganin et al., 2016). Motivated by the effectiveness of Domain Adversarial Neural Network (DANN) (Ganin et al., 2016) frameworks, we aim to achieve domain alignment by formulating a similar two-player mini-max game to match the source and target feature distributions. In the proposed setup, the first player $G_d(\cdot)$ is modeled as a domain classifier and is trained to separate x_s from x_t. The feature extractor, $F(\cdot)$, poses as the second player and is designed to confuse the domain discriminator at the same time by generating domain-invariant features. The overall objective of the DANN is realized by minimizing the following term:

$$\begin{aligned}\mathcal{L}_{adv} = &-\frac{1}{|\mathcal{D}_s|}\sum_{x_s \in \mathcal{D}_s} w_{y_s}[log\ G_d(F(x_s))] \\ &-\frac{1}{|\mathcal{D}_t|}\sum_{x_t \in \mathcal{D}_t} [1 - log\ G_d(F(x_t))]\end{aligned} \tag{1}$$

The negative transfer, potentially caused by source samples in $\mathcal{D}_{\overline{s}}$ (Section 3.1), is checked by regulating the contributions of all source samples. The underlying intuition centers around down-weighting the proportion of learning from private source samples. This is achieved by utilizing weights w_{y_s} reflecting class-importance, estimated over the target samples (detailed process presented in Section 3.2.3).

3.2.2. Classification

The feature extractor output is processed by a label classifier $G_y(\cdot)$, trained using the categorical cross-entropy loss $\mathcal{L}_{cl}(\cdot,\cdot)$ between the ground-truth labels of source samples and their predicted classification results. Similar to the domain adversarial training as presented above, the learning process is regulated by the class-importance weight (see Section 3.2.3) estimated on target samples. The overall classification loss function is represented as:

$$\mathcal{L}_{class} = \frac{1}{|\mathcal{D}_s|}\sum_{x_s \in \mathcal{D}_s} w_{y_s}\mathcal{L}_{cl}\Big(G_y(F(x_s)), y_s\Big) \tag{4}$$

3.2.3. Target supervision and estimating class-importance weights through pseudo-labels

For eliminating the negative influence of samples private to s, we propose a strategy of regulated learning where the model, adaptively, selects out a subset of the target domain samples that are share a high degree of similarity to the source domain. In addition, we enable target domain supervision by incorporating pseudo-labels, generated from a nonparametric classifier. The adopted pseudo-labelling strategy is as follows:

- **Step 1:** For an input sample $x_{s/t}$ in $\mathcal{D}_{s/t}$, we obtain its encoded latent representation $F(x_{s/t})$, using the feature extractor $F(\cdot)$.
- **Step 2:** With the encoded representations $F(x_s)$ of source entries $< x_s, y_s > \in \mathcal{D}_s$, we compute the cluster centers $\{\mu_s{}^c\}, \forall c \in C_s$, where:

$$\mu_s^c = \frac{1}{|\mathcal{D}_s^c|}\sum_{x_s^c \in D_s^c} F(x_s^c) \tag{2}$$

where $D_s{}^c = \{ < x_s, y_s > | < x_s, y_s > \in \mathcal{D}_s, y_s = c\}$ ($|\cdot|$ represents set-cardinality).

- **Step 3:** A similarity function $\Phi(\cdot)$ (returns a vector of size $|C_s|$) processes each $F(x_t)$, $x_t \in \mathcal{D}_t$, that quantifies its closeness to the representative centers of each source class and is represented as:

$$\Phi(F(x_t)) = \left\langle \frac{2 - \mathcal{JS}\big(F(x_t)||\mu_s^c\big)}{2} \right\rangle_{c \in C_s} \tag{3}$$

As highlighted above, we utilize Jensen–Shannon divergence, $\mathcal{JS}(\cdot\|\cdot)$, to measure the divergence between the latent-target vector and representative centers of the source categories in latent space. Each divergence entry is normalized to [0,1], with a higher value signifying greater similarity.

- **Step 4:** Probability predictions for a target sample x_t is computed using a $softmax(\cdot)$ over $\Phi(F(x_t))$. that is,

$$\mathbf{p_t} = softmax\big(\Phi(F(x_t))\big) \tag{4}$$

$$y_t^p \leftarrow \underset{c}{\text{argmax}} \left[softmax\big(\Phi(F(x_t))\big)\right] \tag{5}$$

where y_t^p, in equation (5), represents the predicted pseudo-label for x_t, essential for optimizing the *inter-* and *intra-class* discrepancies (refer to Sections 3.2.4 and 3.2.5). $\mathbf{p_t}$, as highlighted in equation (4), represents a vector of *softmax* probabilities, with k^{th} entry representing the probability of x_t belonging to the k^{th} category in $|C_s|$ classes.

The confidence probability, $max(\mathbf{p_t})$, offers a monitoring mechanism of how likely the target sample x_t is to be mapped with its nearest cluster center. A low value indicates a significant degree of confusion existing in the model, while classifying x_t to a category in C_s. Using unreliable target samples with low confidence values for class-importance weight estimation might sabotage classification by diverting attention away from the desired objective. To eliminate such issues, we adopt a voting strategy for computing the class-importance weight vector $\mathbf{W}$ (with $|C_s|$ elements), in which only a subset of the target samples (those with high confidence predictions) can participate. The dataset $\mathcal{D}_\mathcal{T}$, thus constructed with these highly confident target samples, is mathematically represented as:

$$\mathcal{D}_\mathcal{T} = \left\{\langle x_t, y_t^p\rangle \mid x_t \in \mathcal{D}_t, y_t^p \leftarrow \underset{c}{\text{argmax}}(\mathbf{p_t}), max\big(\mathbf{p_t}\big) \geq \mathcal{T}\right\} \tag{6}$$

The threshold parameter, $\mathcal{T}$, is computed over the predicted outputs of the nonparametric classifier $softmax(\Phi(\cdot))$ and represents the average probability of source domain samples belonging to the ground-truth class, that is,

$$\mathcal{T} = \frac{1}{|\mathcal{D}_s|}\sum_{x_s \in \mathcal{D}_s} max\Big(softmax\big(\Phi(F(x_s))\big)\Big) \tag{7}$$

The class-importance weight vector $\mathbf{W} = [w_{c_1}, w_{c_2}, \ldots, w_{c_k}, \ldots, w_{c_{|C_s|}}]$ is estimated using the equation below:

$$\mathbf{W} \leftarrow \frac{\mathbf{W}'}{max(\mathbf{W}')}, \text{ where } \mathbf{W}' = \frac{1}{|\mathcal{D}_t^\mathcal{T}|}\sum_{x_t \in \mathcal{D}_t^\mathcal{T}} \mathbf{p_t} \tag{8}$$

In the proposed setup, given a source instance $< x_s, y_s > \in \mathcal{D}_s$, where $y_s = c_k$, the corresponding class weight $w_{y_s=c_k}$ in $\mathbf{W}$ regulates the degree of learning from instance $< x_s, y_s >$, as noted in Sections 3.2.1 and 3.2.2. To ensure convenience in representation, the class weight $w_{y_s=c_k}$ is collapsed to w_{y_s} in prior sections.

3.2.4. Maximizing inter-class separation

Having a sufficiently "well-organized" and "regular" latent space arrangement is conducive to realizing the goal of achieving superior classification performance. Therefore, it is essential that samples with same class label are clustered to their respective distribution, while data with different class labels are allocated to distinct class distributions, regardless of their domains. We reinforce this concept by maximizing the L_2-distance between the mean embeddings of two distinct classes. It is worth noting that the approach also seeks to maximize the distance between different classes within the same domain (captured by the first two terms in equation (9)). This objective is partially realized using the *between-class loss* $\mathcal{L}_{bc}$, as presented below:

$$\begin{aligned}\mathcal{L}_{bc} = -\Bigg[\alpha\Big(&\tfrac{1}{|C_s|(|C_s|-1)}\sum_{c \in C_s}\sum_{\substack{c' \in C_s \\ c' \neq c}} ||\mu_s^c - \mu_s^{c'}||_2 \\ &+ \tfrac{1}{|C_\mathcal{T}|(|C_\mathcal{T}|-1)}\sum_{c \in C_\mathcal{T}}\sum_{\substack{c' \in C_\mathcal{T} \\ c' \neq c}} ||\mu_\mathcal{T}^c - \mu_\mathcal{T}^{c'}||_2\Big) \\ &+ \tfrac{\beta}{|C_\mathcal{T}|(|C_\mathcal{T}|-1)}\sum_{c \in C_\mathcal{T}}\sum_{\substack{c' \in C_\mathcal{T} \\ c' \neq c}} ||\mu_s^c - \mu_\mathcal{T}^{c'}||_2\Bigg]\end{aligned} \tag{9}$$

where $||\cdot||_2$ represents L_2-norm. In the equation above, $C_\mathcal{T}$ represents the label set of $D_\mathcal{T}$ (refer to equation (6)), where $C_\mathcal{T} \subseteq C_s$. We follow a similar routine as presented in equation (2), for computing the mean-embedding $\mu_{s/\mathcal{T}}{}^c$ for a class c, over samples in dataset $D_{s/\mathcal{T}}$, respectively. Hyperparameters α and β control the contribution of cross-domain and within-domain components, in equation (9).

3.2.5. Minimizing within-class separation

As previously stated, in addition to maximizing the distance between two different clusters, it is essential to group samples of the same class together, to avoid classification errors on instances near the decision boundaries. We incorporate this concept into our framework by minimizing the distance between the embedded representations of any two samples in the same category, in a domain-agnostic setup. The *within-class loss* $\mathcal{L}_{wc}$, capturing this formulation, is presented as:

$$\mathcal{L}_{wc} = \frac{1}{|C_s|}\sum_{c \in C_s}\left[\frac{1}{|\mathcal{D}^c|(|\mathcal{D}^c| - 1)}\sum_{x^i \in \mathcal{D}^c}\sum_{\substack{x^j \in \mathcal{D}^c \\ x^j \neq x^i}} \left|\left|F(x^i) - F(x^j)\right|\right|^2\right] \tag{10}$$

where dataset $\mathcal{D}^c = \{< x, y > | < x, y > \in \mathcal{D}, y = c \in C_s\}$, with $\mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_\mathcal{T}$. In equation (10), we use indices i, j to represent distinct samples in $\mathcal{D}^c$.

3.2.6. Entropy minimization of target samples

The early stages of a classification process in a domain adaptation setup witness two major adverse effects: (a) difficulty in transferring sample information due to large domain shifts and (b) inducing uncertainty in the classifier. To circumvent such issues, we adopt the entropy minimization principle during model training, which is formulated as:

$$\mathcal{L}_{em} = -\frac{1}{|\mathcal{D}_t|}\sum_{x_t \in \mathcal{D}_t}\sum_{c \in C_s}\left[G_y(F(x_t))\right]^c log\big(\left[G_y(F(x_t))\right]^c\big) \tag{11}$$

where $[G_y(F(x_t))]^c$ represents the probability of x_t belonging to class c, as predicted by the classifier $G_y(\cdot)$.

3.2.7. Overall objective

To sum up, we propose the overall objective function as:

$$\mathcal{L} = \mathcal{L}_{class} + \eta\mathcal{L}_{adv} + \mathcal{L}_{bc} + \gamma\mathcal{L}_{wc} + \mathcal{L}_{em}, \tag{12}$$

where η and γ are user-defined hyperparameters controlling the contribution of each loss in model learning.

4. Experiments

In this section, we perform experiments on two benchmark datasets (Office-Home (Venkateswara et al., 2017) and Office-31 (Saenko et al., 2010)), to evaluate the efficacy of the proposed framework. The experiments are conducted in a *pda* setup, over multiple tasks for each dataset. Results of these tasks are reported and analyzed in the following sections.

4.1. Datasets

For overall performance assessment, we utilize standard datasets for domain adaptation, specifically *Office-Home* and *Office-31*. The Office-31 (Saenko et al., 2010) dataset is grouped into 31 distinct categories, representing three domains: Amazon (A), DSLR (D), and Webcam (D). For evaluation task setup, we replicate the technique adopted by Cao et al. (2018b), where the dataset representing the target domain contains images from 10 distinct classes. The assessment is conducted on the following source-target combinations: A→W, A→D, W→A, W→D, D→A and D→W. The larger Office-Home dataset (Venkateswara et al., 2017) is grouped into four distinct domains: Artistic (Ar), Clip Art (Cl), Product (Pr), and Real-world (Rw). Following the Partial Adversarial Domain Adaptation (PADA) setup (Cao et al., 2018b), we construct the source and target datasets from 65 and 25 different image classes, respectively. Twelve different permutations of source-target are used for evaluation purposes, namely Ar→Cl, Ar→Pr, Ar→Rw, Cl→Ar, Cl→Pr, Cl→Rw, Pr→Ar, Pr→Cl, Pr→Rw, Rw→Ar, Rw→Cl, and Rw→Pr.

4.2. Implementation details

The feature extractor $F(\cdot)$ uses ResNet-50 as backbone network (He et al., 2016), pretrained on ImageNet dataset (Deng et al., 2009). Following DANN (Ganin et al., 2016), we introduce a bottleneck layer of length 256 before the fully connected layers. This layer generates latent representations, utilized by $softmax(\Phi(\cdot))$, $G_y(\cdot)$, and $G_d(\cdot)$ and during optimizing *between-class* separation and *within-class* compactness. Similar to PADA (Cao et al., 2018b), we fine-tune $F(\cdot)$ and train $G_y(\cdot)$ and $G_d(\cdot)$ layers from scratch (learning rates of new layers set to 10 times of $F(\cdot)$). We use mini-batch stochastic gradient descent (SGD) with momentum of 0.9 and follow a similar learning rate update strategy, as in DANN (Ganin et al., 2016). $\mathbf{W}$ is initialized to 1 for all classes and is updated using equation (8) after every 150 epochs. The η is progressively increased from 0 to 1, as in DANN (Ganin et al., 2016). Parameters α, β and γ are set to 0.2, 0.9, 0.7, and 0.1, 0.9, 1.7 for Office-31 (Saenko et al., 2010) and Office-home (Venkateswara et al., 2017) datasets, respectively.

4.3. Comparison models

The model is evaluated against state-of-the-art models for closed-set and partial domain adaptation tasks, namely Resnet-50 (He et al., 2016), DANN (Ganin et al., 2016), Adversarial Discriminative Domain Adaptation (ADDA) network (Tzeng et al., 2017), IWAN (Zhang et al., 2018a), Deep Residual Correction Network (DRCN) (Li et al., 2020), SAN (Cao et al., 2018a), PADA (Cao et al., 2018b), and ETN (Cao et al., 2019).

Table 1
Classification accuracy (%) for partial domain adaptation tasks on Office-Home dataset (backbone: Resnet-50)

Method	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	Avg.
Resnet-50 (He et al., 2016)	46.33	67.51	75.87	59.14	59.94	62.73	58.22	41.79	74.88	67.40	48.18	74.17	61.35
DANN (Ganin et al., 2016)	45.23	68.79	79.21	64.56	60.01	68.29	57.56	38.89	77.45	70.28	45.23	78.32	62.82
ADDA (Tzeng et al., 2017)	45.23	68.79	79.21	64.56	60.01	68.29	57.56	38.89	77.45	70.28	45.23	78.32	62.82
PADA (Cao et al., 2018b)	51.95	67.00	78.74	52.16	53.78	59.03	52.61	43.22	78.79	73.73	56.60	77.09	62.06
DRCN (Li et al., 2020)	54.00	**76.40**	83.00	62.10	64.50	71.00	**70.80**	49.80	80.50	**77.50**	59.10	79.90	69.00
IWAN (Zhang et al., 2018a)	53.94	54.45	78.12	61.31	47.95	63.32	54.17	52.02	81.28	76.46	56.75	82.90	63.56
SAN (Cao et al., 2018a)	44.42	68.68	74.60	**67.49**	64.99	**77.80**	59.78	44.72	80.07	72.18	50.21	78.66	65.30
ETN (Cao et al., 2019)	52.02	63.64	77.95	65.66	59.31	73.48	70.49	51.54	**84.89**	76.25	**60.74**	80.86	68.07
Proposed model	**57.14**	75.45	**83.32**	63.60	**67.69**	72.76	67.39	**52.69**	83.33	74.24	60.13	**81.55**	**69.94**
W/o $\mathcal{L}_{adv}$	55.40	74.08	81.47	58.61	63.86	67.64	61.07	49.11	80.66	72.04	57.65	79.21	66.73
W/o $\mathcal{L}_{bc}$ & $\mathcal{L}_{wc}$	53.25	70.53	77.76	56.24	60.87	64.59	58.48	47.21	77.92	69.16	55.47	76.83	64.02
W/o *sv*	56.34	74.79	82.07	59.48	64.82	68.42	61.91	50.04	81.26	72.68	58.45	80.02	67.52

Table 2
Classification accuracy (%) for partial domain adaptation tasks on Office-31 dataset (backbone: Resnet-50)

Method	A → W	A → D	W → A	W → D	D → A	D → W	Avg.
Resnet-50 (He et al., 2016)	75.59	83.44	84.97	98.09	83.92	96.27	87.05
DANN (Ganin et al., 2016)	73.56	81.53	86.12	98.73	82.78	96.27	86.50
ADDA (Tzeng et al., 2017)	75.67	83.41	84.25	99.85	83.62	95.38	87.03
PADA (Cao et al., 2018b)	86.54	82.17	95.41	**100.00**	92.69	**99.32**	92.69
IWAN (Zhang et al., 2018a)	89.15	90.45	94.26	99.36	**95.62**	**99.32**	94.69
SAN (Cao et al., 2018a)	90.90	94.27	88.73	99.36	94.15	**99.32**	94.96
ETN (Cao et al., 2019)	91.52	90.87	94.36	98.94	90.61	92.88	93.20
Proposed model	**92.45**	**92.79**	**97.23**	**100.00**	95.29	98.61	**96.06**
W/o $\mathcal{L}_{adv}$	88.13	86.21	94.33	98.57	92.99	98.12	93.05
W/o $\mathcal{L}_{bc}$ & $\mathcal{L}_{wc}$	85.52	83.57	91.60	97.54	90.27	96.87	90.89
W/o sv	86.26	84.35	93.37	97.91	91.02	97.49	91.73

5. Results and Analysis

In the results summarized in Tables I and II, the proposed method achieves comparable accuracy results to the state-of-the-art models addressing closed-set and partial domain adaptation, on all the presented tasks (achieving highest accuracy in 5 out of 12 tasks and in 4 out of 6 tasks on Office-Home and Office-31 datasets, respectively). It outperforms the existing methods in the overall *average accuracy percentage* on both datasets. For further analysis, we conducted an ablation study on the proposed model by suppressing its three main components, one at a time:

- **Without $\mathcal{L}_{adv}$**: We restrict the learning of domain-invariant latent representations by masking the adversarial objective (setting η to 0).
- **Without $\mathcal{L}_{bc}$ and $\mathcal{L}_{wc}$**: The model is trained without class distribution alignment losses (setting α, β, γ to 0).
- **Without selective voting (*sv*)**: We limit the utilization of highly confident target samples for computation of class-importance weights (setting threshold $\mathcal{T}$ to 0).

The reduction in classification performance, as witnessed in Tables 1 and 2, after excluding these modules validates their significance in the classification task. Their utility is buttressed further with the model yielding best performance scores on a majority of the adaptation tasks described in Section 4.1.

6. Conclusion

This paper presents a novel framework for partial domain adaptation tasks, that aims to mitigate domain discrepancy, while achieving class-distribution alignment in the latent space. The proposed approach couples adversarial learning with a selective consensus strategy (using a nonparametric classifier) that prevents a set of target samples, with low confidence predictions, from participating in class-importance weight estimation and alignment of category-level distributions. From the experiments conducted over two benchmark datasets, it is established that our approach achieves superior classification results, when compared to state-of-the-art models.

Conflict of Interest

The authors declare that they have no conflicts of interest to this work.

References

Cao, Z., Long, M., Wang, J., & Jordan, M. I. (2018a). Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2724–2732).

Cao, Z., Ma, L., Long, M., & Wang, J. (2018b). Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 135–150).

Cao, Z., You, K., Long, M., Wang, J., & Yang, Q. (2019). Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2985–2994).

Choudhuri, S., Paul, R., Sen, A., Li, B., & Venkateswara, H. (2020). Partial Domain Adaptation Using Selective Representation Learning For Class-Weight Computation. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE (pp. 289–293).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. IEEE (pp. 248–255).

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International conference on machine learning, PMLR* (pp. 1180–1189).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, *1*, 2096–2030. https://jmlr.org/papers/volume17/15-239/15-239.pdf

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hoffman, J., Guadarrama, S., Tzeng, E. S., Hu, R., Donahue, J., Girshick, R., Darrell, T., & Saenko, K. (2014). LSDA: Large scale detection through adaptation. In *Advances in neural information processing systems, vol. 27*.

Jing, T., Xia, H., & Ding, Z. (2020). Adaptively-accumulated knowledge transfer for partial domain adaptation. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 1606–1614).

Li, S., Liu, C. H., Lin, Q., Wen, Q., Su, L., Huang, G., & Ding, Z. (2020). Deep residual correction network for partial domain adaptation. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, *7*, 2329–2344. https://doi.org/10.1109/TPAMI.2020.2964173

Li, S., Liu, C. H., Xie, B., Su, L., Ding, Z., & Huang, G. (2019). Joint adversarial domain adaptation. In *Proceedings of the 27th ACM International Conference on Multimedia* (pp. 729–737).

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1717–1724).

Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. In *European conference on computer vision*. Springer (pp. 213–226).

Sugiyama, M., Krauledat, M., & Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, *8*, 5. https://www.jmlr.org/papers/volume8/sugiyama07a/sugiyama07a.pdf

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7167–7176).

Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5018–5027).

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In *Advances in neural information processing systems, vol. 27.*

Zhang, J., Ding, Z., Li, W., & Ogunbona, P. (2018a). Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8156–8164).

Zhang, L., Wang, P., Wei, W., Lu, H., Shen, C., van den Hengel, A., & Zhang, Y. (2018b). Unsupervised domain adaptation using robust class-wise matching. *IEEE Transactions on Circuits and Systems for Video Technology*, *5*, 1339–1349. https://doi.org/10.1109/TCSVT.2018.2842206

How to Cite: Choudhuri, S., Venkateswara, H., & Sen, A. (2022). Coupling Adversarial Learning with Selective Voting Strategy for Distribution Alignment in Partial Domain Adaptation. *Journal of Computational and Cognitive Engineering* 1(4), 181–186, https://doi.org/10.47852/bonviewJCCE2202324