

Domain Consistency Regularization for Unsupervised Multi-source Domain Adaptive Classification

Zhipeng Luo^{a,b}, Xiaobing Zhang^c, Shijian Lu^{a,*}, Shuai Yi^b

^aNanyang Technological University, Singapore

^bSensetime Research, 182 Cecil Street, 36-02 Frasers Tower, Singapore

^cUniversity of Electronic Science and Technology of China, China

Abstract

Deep learning-based multi-source unsupervised domain adaptation (MUDA) has been actively studied in recent years. Compared with single-source unsupervised domain adaptation (SUDA), domain shift in MUDA exists not only between the source and target domains but also among multiple source domains. Most existing MUDA algorithms focus on extracting domain-invariant representations among all domains whereas the task-specific decision boundaries among classes are largely neglected. In this paper, we propose an end-to-end trainable network that exploits domain Consistency Regularization for unsupervised Multi-source domain Adaptive classification (CRMA). CRMA aligns not only the distributions of each pair of source and target domains but also that of all domains. For each pair of source and target domains, we employ an intra-domain consistency to regularize a pair of domain-specific classifiers to achieve *intra-domain alignment*. In addition, we design an inter-domain consistency that targets joint *inter-domain alignment* among all domains. To address different similarities between multiple source domains and the target domain, we design an authorization strategy that assigns different authorities to domain-specific classifiers adaptively for optimal pseudo label prediction and self-training. Extensive experiments show that CRMA tackles unsupervised domain adaptation effectively

*Corresponding author

Email addresses: zhipeng001@e.ntu.edu.sg (Zhipeng Luo), zxbing_uestc@163.com (Xiaobing Zhang), shijian.lu@ntu.edu.sg (Shijian Lu), yishuai@sensetime.com (Shuai Yi)

under a multi-source setup and achieves superior adaptation consistently across multiple MUDA datasets.

Keywords: Domain Adaptation, Transfer Learning, Adversarial Learning, Feature Alignment

1. Introduction

In recent years, deep neural networks have brought great improvements to a variety of visual learning tasks, such as classification [1], segmentation [2, 3], and detection [4, 5]. These achievements mainly attribute to the availability of large-scale labeled data for supervised learning. However, it is prohibitively labor-intensive and time-consuming to collect abundant labeled data for each new task. Domain Adaptation (DA) aims to tackle this problem by utilizing labeled data in relevant domains. Specifically, it leverages a label-rich domain(s) (i.e., source domain(s)) to learn a discriminative model that generalizes well on a label-scarce domain (i.e., target domain). Most DA methods focus on single-source unsupervised domain adaptation (SUDA), where the labeled data in a single source domain are adapted via discrepancy minimization [6, 7], adversarial learning [8, 9], prototypical networks [10], etc.

In real life, information often comes in a wide variety of formats and origins and this makes the learning process more complicated. Multi-source unsupervised domain adaptation (MUDA) aims to adapt from multiple labeled source domains of different distributions to a single target domain. It has been tackled by learning domain invariant features and predicting pseudo labels for target-domain samples and has achieved promising results in various benchmarks [11, 12, 13, 14, 15]. On the other hand, existing methods have some common constraints. First, using domain classifiers (e.g., discriminators) to learn domain invariant features [16, 15, 13] tends to suffer from over-alignment problems since it neglects the task-specific decision boundaries of different categories. Second, predicting pseudo labels for target-domain samples [11] often suffers from label noises due to the different distributions of source-domain sam-

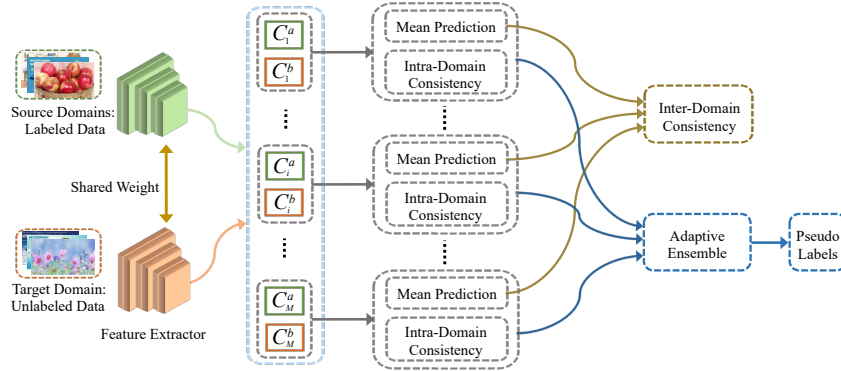


Figure 1: The architecture of our proposed MUDA network CRMA: We adapt from multiple labeled *Source Domains* to one unlabeled *Target Domain*. For each source domain, we train two domain-specific classifiers and employ *Intra-Domain Consistency* for intra-domain alignment between the source and target domains. We design *Inter-Domain Consistency* to fuse multiple *Mean Predictions* for jointly aligning across all domains. We also design an *Adaptive Ensemble* strategy based on *Intra-Domain Consistency* which predicts *Pseudo Labels* adaptively for handling negative transfer. C_i^a and C_i^b denote the classifier pair for the i -th source domain, while M is the number of source domains. Best viewed in color.

ples and the trained models. It often requires heuristic thresholds for identifying high-confidence predictions. However, selecting heuristic thresholds is a challenging task in MUDA where multiple source domains often have different similarities with the target domain and require different heuristic thresholds for optimal pseudo label prediction.

To address the aforementioned issues, we propose an end-to-end trainable network that exploits Consistency Regularization for unsupervised Multi-source domain Adaptive classification (CRMA). CRMA performs both intra-domain alignment and inter-domain alignment as illustrated in Fig. 1. On top of training a feature extractor and a pair of classifiers for each source domain [17], we compute the intra-domain consistency of target predictions for each classifier pair and adopt a min-max optimization strategy for domain-specific feature alignment between the target domain and each source domain. In addition, we maximize the inter-domain consistency as computed from the target predictions of different domain-specific classifiers to boost the feature space alignment across

all domains. To facilitate model selection and avoid the *negative transfer* [18] issue, we design an adaptive self-training strategy that treats the intra-domain consistency as a confidence indicator and uses it to fuse the domain-specific predictions to generate pseudo labels and refine the network. Despite employing multiple classifiers, the proposed CRMA introduces little overheads in model size and computational complexity as the feature extractor is shared which carries most weights in a typical CNN architecture. Extensive experiments show that CRMA outperforms state-of-the-art methods consistently with clear margins across multiple MUDA datasets.

The contributions of this work can be summarized in three aspects. First, we propose an end-to-end trainable MUDA network that leverages intra-domain alignment and inter-domain alignment for effective adaptation from multiple source domains to one target domain. Second, we develop an adaptive self-training strategy that serves the function of model selection and tackles *negative transfer* effectively. Third, extensive experiments show that our method achieves superior domain adaptation performance consistently across multiple MUDA datasets.

2. Related Works

Single-source Unsupervised Domain Adaptation (SUDA). SUDA aims to learn a model well-performing on the target domain given a labeled source domain and an unlabeled target domain. It is usually achieved by discrepancy-based methods [6, 19, 20], adversarial learning [8, 21, 22, 23, 24], and self-training methods [25]. Tzeng et al. [6] first proposed Maximum Mean Discrepancy (MMD) to measure the distance between domain distributions. Han et al. [19] designed a modified \mathcal{A} -distance to preserve the internal structures for target domain examples during domain adaptation. Yao et al. [20] introduced a unified framework that incorporates discriminative embedding constraints and distribution alignment. Ganin et al. [8] first utilized a domain discriminator to align feature representations with adversarial learning. Zuo et al. [21] applied

different strategies to easy and tough examples to improve the domain adaptation performance. Rahman et al. [22] combined correlation alignment with adversarial learning to tackle the domain adaptation and domain generalization problems. Liang et al. [25] proposed to leverage the uncertainty of pseudo labels to achieve optimal feature transformation. Several studies address category-level feature alignment using dual task-specific classifiers [17] and prototypical networks [10]. SUDA methods usually suffer from sub-optimal performance when directly applied to MUDA tasks since different source domains might have different levels of similarity to the target domain. Our proposed method extends [17] but works under a more complicated multi-source setting. The key difference is that the feature alignment needs to be performed between the target domain and multiple source domains that have different similarities with the target domain.

Multi-source Unsupervised Domain Adaptation (MUDA). MUDA aims to adapt from multiple labeled source domains to one unlabeled target domain. Yang et al. [26] first introduced the output ensemble of source-domain classifiers for tuning the target-domain categorization model. This idea was later extended by several shallow models via feature representation [27] and a combination of pre-learned classifiers [28] under certain theoretical supports [29, 30]. In recent years, deep learning-based approaches have been developed to tackle the MUDA challenge by extracting domain invariant representations among all domains. Xu et al. [11] presented a deep cocktail network (DCTN) that adopts adversarial learning and employs perplexity scores for target prediction voting. Zhao et al. [31] introduced a multi-source domain adversarial network (MDAN) that combines the gradient of domain classifiers for bound optimization. M³SDA [32] applies moment matching to align the feature representations of multiple domains dynamically. Zhao et al. [13] proposed a multi-source distilling domain adaptation (MDDA) network to handle different similarities between multiple source domains and the target domain by generating a weighted ensemble of multiple target predictions. Wang et al. [14] applied prototypical networks and knowledge graph for knowledge aggregation.

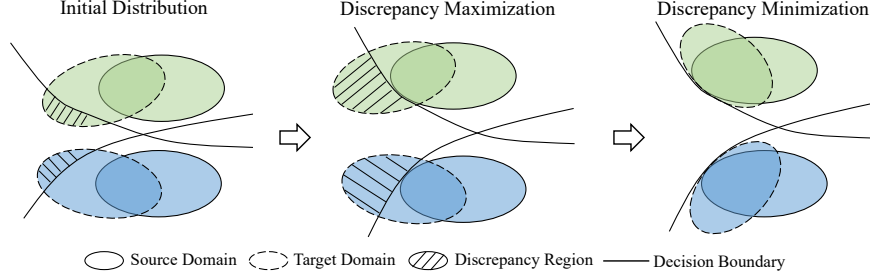


Figure 2: Illustration of feature alignment with maximum classifier discrepancy (MCD) [17]: The discrepancy between two classifiers (for each source domain) is first maximized to detect target samples that are misaligned with the source domain. It is then minimized to guide the feature extractor to learn domain-invariant feature representations for the source and target domains. Green and blue colors denote two different classes. Best viewed in color.

Existing adversarial learning methods focus on learning domain-invariant representation across all domains, while the task-specific decision boundaries among different classes are neglected. Moreover, when aggregating multiple target predictions, existing methods rely on pre-training [11, 32] or require multi-stage training [13], which are complicated and sub-optimal. In this work, we design an end-to-end framework that leverages the consistency of target predictions for domain-specific and cross-domain feature alignment. We also incorporate model selection into the training process through a novel self-training mechanism.

3. Methods

3.1. System Overview

Suppose we have M labeled source domains $\{\{X_1, Y_1\}, \{X_2, Y_2\}, \dots, \{X_M, Y_M\}\}$ and one unlabeled target domain $\{X_T\}$. We train a model that consists of a feature extractor F shared across all domains as well as two domain-specific classifiers for each source domain which are denoted by $\{(C_1^a, C_1^b), (C_2^a, C_2^b), \dots, (C_M^a, C_M^b)\}$ as shown in Fig. 1. Given an input image x , we use $p(y|x)$ to denote the prediction of the model, e.g. we have $p_m^a(y|x) = C_m^a(F(x))$ for the

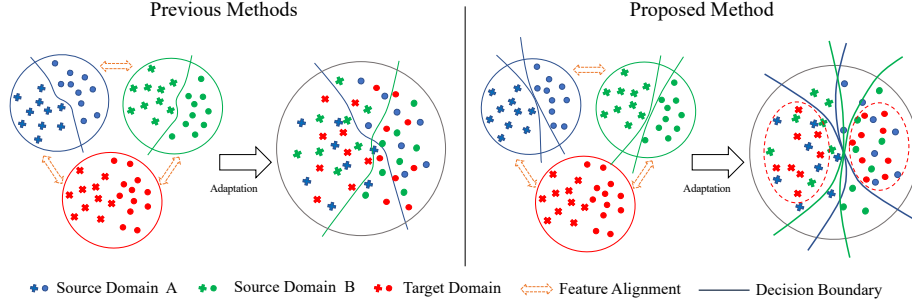


Figure 3: Illustration of feature alignment with existing MUDA methods and the proposed CRMA: Existing MUDA methods align features at domain level without considering class-specific decision boundaries and tend to misclassify target samples lying around the decision boundaries of source domains. With IntraDA and InterDA, the proposed CRMA aligns target features to the overlapped regions of different source domains, which enables more accurate target sample classification. Best viewed in color.

first classifier of the m -th source domain. The goal of the MUDA task is to maximize the performance on the target domain.

We design Intra-Domain Alignment (IntraDA), Inter-Domain Alignment (InterDA), and Adaptive Self-Training (AST) for MUDA. Inspired by the SUDA method MCD [17], IntraDA exploits a similar min-max optimization strategy that employs two classifiers to align source and target features. As illustrated in Fig. 2, the two classifiers are first trained to maximize their discrepancy to detect target samples that are misaligned with the source domain. The discrepancy is then minimized to guide the feature extractor to learn domain-invariant representations that can better classify those misaligned target samples. Different from adversarial methods that perform domain-level alignment, it performs class-level alignment with class-specific decision boundaries of the two classifiers, which can better handle target samples around the decision boundaries.

IntraDA aligns the target domain with each source domain individually but cannot handle MUDA well when multiple source domains of different distributions are present. We design InterDA to capture the synergy of multiple source domains while jointly aligning with the target domain. Specifically, In-

terDA computes inter-domain consistency that measures how the predictions of multiple domain-specific classifiers agree with each other. By maximizing this consistency with a similar mechanism adopted in IntraDA, InterDA encourages consistent target predictions from domain-specific classifiers. As illustrated in Fig. 3, IntraDA and InterDA move target features to the overlapped regions of different source domains, which can better classify target samples lying around the decision boundaries. As a comparison, existing adversarial MUDA methods align features without considering class-specific decision boundaries, which tend to misclassify those target samples.

Another feature of MUDA is that different source domains have different similarities to the target domain and should have different weights while performing the alignment. While some source domains have very different distributions from the target domain, feature alignment could even suffer from *negative transfer* that affects target performance negatively. The proposed Adaptive Self-Training (AST) generates pseudo labels (for target samples) by assigning authorities to domain-specific classifiers adaptively. The idea is to assign higher authorities to more confident classifiers which mitigates the negative effects of low-confidence classifiers effectively. In CRMA, we determine the classifier authorities by using the intra-domain consistency which provides a good measure of prediction confidence for different domain-specific classifiers. More details of the three designs are to be discussed in the ensuing subsections.

3.2. Intra-Domain Alignment

The shared feature extractor and domain-specific classifiers are first trained with source-domain samples to obtain discriminative features by using softmax cross entropy loss:

$$\min_{F,C} L_{src} = - \sum_{m=1}^M \mathbb{E}_{(x_m, y_m) \sim (X_m, Y_m)} \sum_{k=1}^K \mathbb{1}_{[k=y_m]} \log p(y|x_m) \quad (1)$$

where K denotes the number of classes. Subsequently, we compute the intra-domain consistency by summing up the discrepancy between the target predictions of the two domain-specific classifiers and perform intra-domain consistency

minimization by fixing the feature extractor F and training the domain-specific classifiers C . It adjusts the decision boundaries of the classifiers so that they could detect target samples misaligned with the source domains (i.e. where the classifier pairs disagree with each other). We follow the practice in [17] to perform this update with the source data classification loss. The objective can be described by:

$$\min_C L_{src} - L_{intra} \quad (2)$$

where

$$L_{intra} = \mathbb{E}_{x_t \sim X_t} \sum_{m=1}^M d(p_m^a(y|x_t), p_m^b(y|x_t)) \quad (3)$$

L_{intra} stands for the intra-domain consistency loss and d denotes the discrepancy in the form of L1 loss that measures the distance between two prediction probability vectors:

$$d(p, q) = \frac{1}{K} \sum_{k=1}^K |p_k - q_k| \quad (4)$$

Next, we conduct the consistency maximization step which works in an adversarial manner with the consistency minimization step by fixing the classifiers and updating the shared feature extractor. This aims to train the feature extractor to generate domain-invariant features for the target domain and each source domain. In practice, we combine this update with the inter-domain alignment to be discussed in the next subsection to simplify the training process.

3.3. Inter-Domain Alignment

While IntraDA aligns the target domain and individual source domains, InterDA aims to learn domain-invariant representations across all domains. It works in a similar fashion as IntraDA by maximizing the inter-domain consistency of the target predictions. The inter-domain consistency L_{inter} is calculated by summing up the discrepancy among the mean predictions of all

domain-specific classifier pairs:

$$L_{inter} = \mathbb{E}_{x_t \sim X_t} \sum_{m=1}^M \sum_{n=m+1}^M d(\hat{p}_m, \hat{p}_n) \quad (5)$$

where

$$\hat{p}_m = (p_m^a(y|x_t) + p_m^b(y|x_t))/2 \quad (6)$$

where \hat{p}_m stands for the mean prediction of the classifier pair corresponds to the m -th domain. We then combine local and inter-domain consistency losses and minimize the loss by fixing the classifiers and updating the feature extractor:

$$\min_F L_{intra} + \alpha L_{inter} \quad (7)$$

where α is the loss ratio for the inter-domain consistency loss, which is a hyper-parameter of the proposed network.

InterDA works under a similar principle as IntraDA that it guides the feature extractor to generate target domain features that align with all source domains by boosting the inter-domain consistency of the target predictions from different source domains. Note that we do not perform the inter-domain consistency minimization as done in IntraDA because the classifier pairs correspond to different source domains naturally form different decision boundaries as labeled data from different source domains are used for their training, which leads to inconsistency in the target predictions.

3.4. Adaptive Self-Training

On top of consistency-based feature alignment, we design a self-training mechanism to generate pseudo labels for target samples by weighting the target predictions of different domain-specific classifier pairs adaptively. We use the intra-domain consistency as an indicator of prediction confidence since better alignments between the source and target domains lead to a smaller discrepancy and thus, higher confidence in the target predictions. Specifically, the pseudo label $P(y|x_t)$ is generated as follows:

$$P(y|x_t) = \sum_{m=1}^M \frac{w_m}{\sum_{n=1}^M w_n} \hat{p}_m \quad (8)$$

where

$$w_m = 1/(L_{intra}^m + \lambda \bar{L}_{intra}^m) \quad (9)$$

where w_m is the weight of the m -th source domain which is computed based on the intra-domain consistency loss L_{intra}^m and the mean of L_{intra}^m over all seen training samples denoted by \bar{L}_{intra}^m . As L_{intra}^m captures the prediction confidence on the current training sample, \bar{L}_{intra}^m captures the mean prediction confidence which is affected by the similarity between the m -th source domain and the target domain. Here \bar{L}_{intra}^m acts as a regularization factor which mitigates large fluctuations of w_m and the modulation strength is controlled by λ .

After obtaining the pseudo label, we update the model using KL divergence loss [33]:

$$\begin{aligned} \min_{F,C} L_{AST} = \mathbb{E}_{x_t \sim X_t} \sum_{m=1}^M \beta (D_{KL}(p_m^a(y|x_t)||P(y|x_t)) \\ + D_{KL}(p_m^b(y|x_t)||P(y|x_t))) \end{aligned} \quad (10)$$

where

$$\beta = \min(\bar{L}_{intra}^m) \sum_{m=1}^M w_m \quad (11)$$

Parameter β is the weight of the adaptive self-training loss L_{AST} , which aims to suppress the impact of the less confident pseudo labels in the self-training process. It is computed by summing up w_m over M source domains and multiplying with the minimum of \bar{L}_{intra}^m among source domains. As w_m is a confidence indicator of individual target prediction, the sum of w_m captures the overall prediction confidence from all source domains on the current target sample. While \bar{L}_{intra}^m captures the mean confidence over all samples, $\min(\bar{L}_{intra}^m)$ is the mean prediction confidence by the most confident source domain which is a modulator for β . Under such designs, the generated pseudo labels with less overall confidence will be assigned with smaller weights while computing L_{AST} and thus have smaller impacts on the learning process.

Algorithm 1 Consistency-Regularized Self-Training for multi-source domain adaptation (CRMA)

Input: labeled source domains $\{\{X_1, Y_1\}, \{X_2, Y_2\}, \dots, \{X_M, Y_M\}\}$, an unlabeled target domain X_T . Feature extractor F and classifiers C .

Output: Trained feature extractor F' and classifiers C' .

```

1: for  $iteration = 1, 2, \dots$  do
2:   Sample  $\{x_m, y_m\}_{m=1}^M$  from the source domains and  $x_t$  from the target
   domain
3:   Update feature extractor  $F$  and classifiers  $C$  using Eq. 1 with source data

4:   Perform intra-domain consistency minimization by updating  $C$  with Eq.
   2
5:   Compute intra-domain consistency  $L_{intra}$  with Eq. 3
6:   Compute intra-domain consistency  $L_{inter}$  with Eq. 5
7:   Perform consistency maximization by updating  $F$  with Eq. 7
8:   Compute pseudo labels  $P(y|x)$  with Eq. 8
9:   Update  $F$  and  $C$  using  $P(y|x)$  with Eq. 10
10: end for
11: return  $F' = F, C' = C$ 

```

3.5. Network Training

Algorithm 1 summarizes the CRMA training process. In each training iteration, we randomly pick training samples from each source domain and the target domain and train a pair of domain-specific classifiers by using the sampled source samples (with labels). The intra-domain consistency is computed according to the target predictions of domain-specific classifier pairs, while the mean prediction is derived from the classifier pairs' predictions to compute the inter-domain consistency. Intra-domain consistency minimization is performed to allow classifiers to detect misaligned target samples and both local and inter-domain consistency are maximized to achieve domain-specific and domain-agnostic feature alignment. For AST, pseudo labels of target-domain samples are predicted

by taking the weighted average of the mean predictions based on intra-domain consistency and used to update the whole network. In evaluation, we average the probability vectors of all classifiers to generate the final prediction.

4. Experiments

4.1. Datasets

We compare our CRMA with state-of-the-art methods over three public datasets as listed:

Digits-5 contains images of digits from five different visual domains including handwritten images in MNIST [34] and USPS [35], combined images in MNIST-M [36], street images in SVHN [37], and synthetic images in SYN [36]. For fair comparisons, we follow the same training-testing split as in [32].

DomainNet [32] is a recent large-scale domain adaptation dataset. Images in DomainNet are collected from the Internet and categorized into six domains including clipart, infograph, painting, quickdraw, real, and sketch. The total number of images in this dataset is about 600,000 and each domain has 345 categories.

PACS [38] contains images from four domains which are art painting, cartoon, photo, and sketch. Each domain contains objects from 7 categories.

4.2. Experimental Setups

We compare CRMA with state-of-the-art MUDA methods including MDAN [16], DCTN [11], M³SDA [32], MDDA [13], and LtC-MSDA [14]. Being an emerging research area, MUDA has relatively small literature so we also compare with SUDA methods DAN [39], DANN [8], ADDA [9], and MCD [17] for more comprehensive evaluations. For the compared SUDA methods, both single-best and source-combined setups are evaluated, where the former adapts each source domain separately and reports the best model accuracy on the target domain while the latter combines all source domains as a single source domain for adaptation and evaluation.

Table 1: The training setups for the three studied datasets: The *Batch size* denotes the number of training samples in each mini-batch drawn from each domain during training.

Dataset	Backbone	Image size	Batch size	Learning rate	Epoch
Digits-5	Lenet [34]	32×32	128	1×10^{-3}	50
DomainNet	Resnet-101 [1]	224×224	16	1×10^{-3}	10
PACS	Resnet-18 [1]	224×224	16	1×10^{-3}	100

In the experiments, we use the same backbone networks as the comparing methods on each of the three datasets. Table 1 shows the network architectures and the training parameters for the experiments. Specifically, Lenet [34] consists of three convolutional layers and three fully connected layers, and the input channels for the three fully connected layers are 8192, 3072, and 2048, respectively. For Resnet-101 and Resnet-18 [1], we use two fully connected layers following the convolution blocks with input channels of (2048, 1000) and (512, 512), respectively. In all our experiments, we treat the convolutional layers as the feature extractor F and fully connected layers as the classifier C . The Lenet model is trained from scratch. For ResNet-18 and ResNet-101, we follow [14] and load the checkpoints pre-trained on ImageNet [40] as the feature extractor. Besides, we adopt the practice in [41] and assign a smaller learning rate to the pre-trained feature extractors during training, which is 1/10 of the base learning rate. To speed up the training process for the dataset DomainNet, we apply the cosine annealing scheduling [42] to adjust the learning rate. α is set to 0.5 and λ is set to 0.1 for all the experiments. For each experiment, we conduct five random runs and report the average performance.

4.3. Experimental Results

For each dataset evaluated, we take turns to put each domain as the target domain and the rest as the source domains.

Table 2 shows the experimental results on the Digits-5 dataset. We can see that CRMA achieves an average classification accuracy of 94.3%, which is 2.5%

Table 2: Comparing CRMA with the state-of-the-art on Digits-5 (in classification accuracy %).

Standards	Methods	MNIST-M	MNIST	USPS	SVHN	SYN	Avg
Single Best	Source Only	59.2±0.6	97.2±0.6	84.7±0.8	77.7±0.8	85.2±0.6	80.8
	DAN [39]	63.9±0.7	96.3±0.5	94.2±0.9	62.5±0.7	85.4±0.8	80.4
	DANN [8]	71.3±0.6	97.6±0.8	92.3±0.9	63.5±0.8	85.4±0.8	82.0
	ADDA [9]	71.6±0.5	97.9±0.8	92.8±0.7	75.5±0.5	86.5±0.6	84.8
Source Combine	Source Only	63.4±0.7	90.5±0.8	88.7±0.9	63.5±0.9	82.4±0.6	77.7
	DANN [8]	70.8±0.8	97.9±0.7	93.5±0.8	68.5±0.5	87.4±0.9	83.6
	ADDA [9]	72.3±0.7	97.9±0.6	93.1±0.8	75.0±0.8	86.7±0.6	85.0
	MCD [17]	72.5±0.7	96.2±0.8	95.3±0.7	78.9±0.8	87.5±0.7	86.1
Multi-Source	MDAN [16]	69.5±0.3	98.0±0.9	92.4±0.7	69.2±0.6	87.4±0.5	83.3
	DCTN [11]	70.5±1.2	96.2±0.8	92.8±0.3	77.6±0.4	86.8±0.8	84.8
	M ³ SDA [32]	72.8±1.1	98.4±0.7	96.1±0.8	81.3±0.9	89.6±0.6	87.7
	MDDA [13]	78.6±0.6	98.8±0.4	93.9±0.5	79.3±0.8	89.7±0.7	88.1
	LtC-MSDA [14]	85.6±0.8	99.0±0.4	98.3±0.4	83.2±0.6	93.0±0.5	91.8
	CRMA	94.5±0.4	99.0±0.1	98.0±0.3	85.6±1.0	94.6±0.1	94.3

higher than the state-of-the-art by LtC-MSDA [14]. In addition, CRMA obtains higher or comparable accuracy when all domains are used as the target domains. In particular, when transferring knowledge from other domains to MNIST-M, a significant improvement of 8.9% is achieved. From the experiments, we observe that the source domains have relatively balanced contributions when MNIST-M is the target domain, which indicates that each source domain shares fair similarity with the target. This implies that our method is especially beneficial in gathering useful information from different source domains, which also explains why MUDA often outperforms SUDA in a multi-source setup. CRMA also achieves an accuracy gain of 2.4% on SVHN, which is known as the most difficult target domain due to its different similarities to the source domains. The superior accuracy indicates that CRMA is capable of extracting useful knowledge in a noisy environment.

Table 3 shows the comparison over DomainNet which is a more challenging dataset due to its large number of categories and significant domain shift. CRMA achieves an average accuracy of 48.2% and performs best in the major-

Table 3: Comparing CRMA with the state-of-the-art on DomainNet (in classification accuracy %).

Standards	Methods	Clipart	Infograph	Painting	Quickdraw	Real	Sketch	Avg
Single Best	Source Only	39.6±0.6	8.2±0.8	33.9±0.6	11.8±0.7	41.6±0.8	23.1±0.7	26.4
	DANN [8]	37.9±0.7	11.4±0.9	33.9±0.6	13.7±0.6	41.5±0.7	28.6±0.6	27.8
	ADDA [9]	39.5±0.8	14.5±0.7	29.1±0.8	14.9±0.5	41.9±0.8	30.7±0.7	28.4
	MCD [17]	42.6±0.3	19.6±0.8	42.6±1.0	3.8±0.6	50.5±0.4	33.8±0.9	32.2
Source Combine	Source Only	47.6±0.5	13.0±0.4	38.1±0.5	13.3±0.4	51.9±0.9	33.7±0.5	32.9
	DANN [8]	45.5±0.6	13.1±0.7	37.0±0.7	13.2±0.8	48.9±0.7	31.8±0.6	32.6
	ADDA [9]	47.5±0.8	11.4±0.7	36.7±0.5	14.7±0.5	49.1±0.8	33.5±0.5	32.2
	MCD [17]	54.3±0.6	22.1±0.7	45.7±0.6	7.6±0.5	58.4±0.7	43.5±0.6	38.5
Multi-Source	MDAN [16]	52.4±0.6	21.3±0.8	46.9±0.4	8.6±0.6	54.9±0.6	46.5±0.7	38.4
	DCTN [11]	48.6±0.7	23.5±0.6	48.8±0.6	7.2±0.5	53.5±0.6	47.3±0.5	38.2
	M ³ SDA [32]	58.6±0.5	26.0±0.9	52.3±0.6	6.3±0.6	62.7±0.5	49.5±0.8	42.6
	MDDA [13]	59.4±0.6	23.8±0.8	53.2±0.6	12.5±0.6	61.8±0.5	48.6±0.8	43.2
	LtC-MSDA [14]	63.1±0.5	28.7 ±0.7	56.1 ±0.5	16.3±0.5	66.1±0.6	53.8±0.6	47.4
	CRMA	67.6 ±0.6	25.3±0.4	55.1±0.3	18.4 ±0.5	66.9 ±0.3	56.0 ±0.3	48.2

ity of the target domains. For the challenging target domain *quickdraw* where negative transfer is often observed, CRMA obtains an accuracy of 18.4% with a clear margin of 2.1%.

Table 4 shows the comparison on PACS dataset. It can be seen that CRMA achieves an average accuracy of 90.6% with a margin of 0.7% as compared with the state-of-the-art. The performance gain is smaller because this dataset is relatively small and the performance relies heavily on the pre-trained model. In addition, a majority of target domains are very similar to the source domains and the classification accuracy is close to saturation ($> 90\%$).

4.4. Ablation Study

In this subsection, we conduct ablation studies to analyze the effectiveness and contributions of different components in our proposed CRMA.

Specifically, CRMA consists of three major components: intra-domain alignment (IntraDA), inter-domain alignment (InterDA), and Adaptive Self-Training (AST). To investigate the contribution of these three components, we conduct a series of ablation studies over Digits-5 by applying relevant losses in network

Table 4: Comparing CRMA with the state-of-the-art on PACS (in classification accuracy %).
(A: art painting, C: cartoon, S: sketch, P: photo)

Standards	Methods	A	C	S	P	Avg
Single Best	Source Only	68.2	60.0	61.8	95.2	71.3
	ADDA [9]	75.9	80.7	66.4	93.0	79.0
	MCD [17]	81.2	84.5	65.9	96.9	82.1
Source Combine	Source Only	82.9	76	66.4	93.2	79.6
	ADDA [9]	87	83.9	73.7	94.7	84.8
	MCD [17]	88.2	85.2	61.0	97.2	82.9
Source Combine	MDAN [16]	83.5	86.7	71.8	94.5	84.7
	DCTN [11]	84.7	86.7	71.8	95.6	84.7
	M ³ SDA [32]	84.2	85.7	74.6	94.5	84.7
	MDDA [13]	86.7	86.2	77.6	93.9	86.1
	LtC-MSDA [14]	90.2	90.5	81.5	97.2	89.9
	CRMA	91.5	92.3	80.9	97.7	90.6

training. We also conduct experiments over the *Source-Only* condition to derive a baseline, where the model is trained by using the labeled source-domain data and applied to the target-domain data without domain adaptation. Table 5 shows experimental results. It is interesting to observe that the *Source-Only* results obtained with our proposed network architecture achieve an average accuracy of 82.1%, which is higher than both the single-best and source-combine baselines as shown in Table 2. This shows that adopting a network architecture of shared feature extractor and domain-specific classifiers leads to more effective network learning and generalization under a multi-source domain adaptation setup.

We can also observe that when each of the three components is incorporated, there is a clear performance gain as compared to the *Source-Only* baseline. Specifically, incorporating AST achieves the highest accuracy of 90.5% while incorporating IntraDA introduces the lowest gain, which is expected since IntraDA only addresses domain-specific feature alignment and leads to imbal-

Table 5: Ablation study of CRMA on Digits-5 (in classification accuracy %). (MM: MNIST-M, MT: MNIST, UP: USPS, SV: SVHN, SY: SYN)

IntraDA	InterDA	AST	MM	MT	UP	SV	SY	Avg
			67.9	98.0	95.4	69.8	79.4	82.1
✓			72.9	98.3	95.8	76.9	84.3	85.6
	✓		75.8	99.1	98.2	75.0	92.5	88.1
		✓	79.4	99.1	98.1	82.7	93.4	90.5
✓	✓		75.8	98.3	97.2	79.0	88.4	87.7
✓		✓	93.2	99.0	97.4	83.2	94.1	93.4
	✓	✓	82.5	99.1	98.4	84.1	94.2	91.7
✓	✓	✓	94.4	99.0	98.0	85.6	94.6	94.3

Table 6: Comparing AST with uniform ensemble on dataset Digits-5 (in classification accuracy %). (MM: MNIST-M, MT: MNIST, UP: USPS, SV: SVHN, SY: SYN)

Methods	MM	MT	UP	SV	SY	Avg
Uniform Ensemble	93.8	99.0	98.3	79.0	94.3	92.9
AST	94.5	99.0	98.0	85.6	94.6	94.3

anced performance across domain-specific classifiers on the target domain. In addition, IntraDA and AST are complementary that incorporating both produces an average accuracy of 93.4%, largely because the self-training effectively aggregates domain-specific knowledge and closes the performance gap among classifiers. On the other hand, IntraDA and InterDA do not demonstrate clear complementary effects. We observe from the experiments that although InterDA aligns the feature distributions globally, it does not resolve the imbalanced performance of IntraDA well. Furthermore, InterDA is less complementary to AST either as compared to IntraDA as both InterDA and AST address the global alignment problem. Finally as expected, the complete network model with all three components incorporated achieves the best classification accuracy.

We also conduct a sensitivity analysis for the hyper-parameter λ , a weighting factor in Eq. 9 that balances individual prediction confidence and the mean

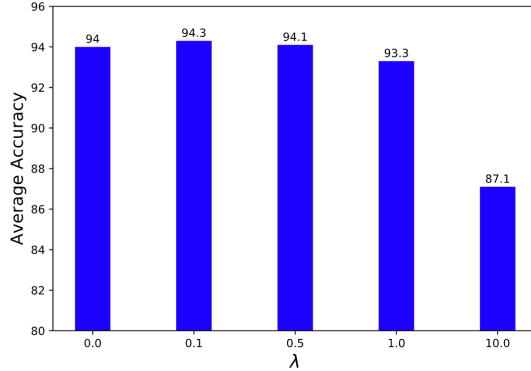


Figure 4: Ablation study of λ on Digits-5 dataset. The classification is stable while λ lies between 0 and 1 (best performance is obtained while $\lambda = 0.1$).

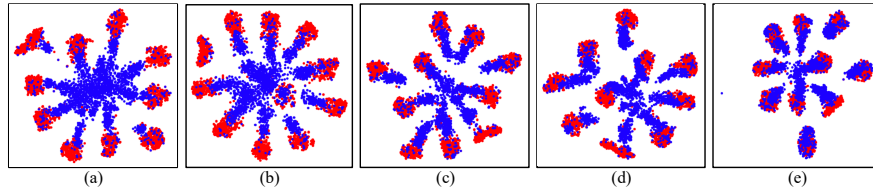


Figure 5: t-SNE visualization of feature representations of a source domain (MNIST) and a target domain (MNIST-M) in Digits-5: (a) Source Only with no adaptation, (b) Intra-Domain Alignment, (c) Inter-Domain Alignment, (d) Adaptive Self-Training, (e) IntraDA + InterDA + AST. Red/blue points represent source/target domain. Best viewed in color.

confidence. Fig. 4 shows the average classification accuracy when λ is set to different values (tested on Digits-5). It can be seen that the classification performance is stable while λ lies between 0 and 1 and the best accuracy is obtained at $\lambda = 0.1$.

4.5. Discussion

To examine the effectiveness of our adaptive pseudo label generation in AST, we replace our generated pseudo labels with the *Uniform Ensemble* of target predictions (where each classifier has an equal contribution) and benchmark over Digits-5. As Table 6 shows, AST outperforms the *Uniform Ensemble* by 1.4% in average classification accuracy. For easier target domains that share

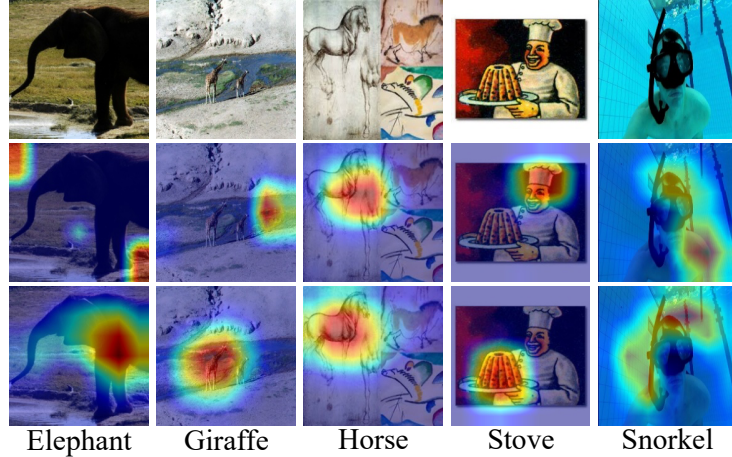


Figure 6: Grad-CAM visualization on sample images from PACS and DomainNet datasets: For the sample images in Row 1, Rows 2 and 3 show the corresponding class activation maps with no domain adaptation and with CRMA, respectively. The text below images indicates the class labels. Best viewed in color.

good similarity with the source domains, AST achieves similar accuracy as *Uniform Ensemble*. However, for difficult target domain SVHN, AST outperforms *Uniform Ensemble* by a large margin at 6.6%. Such experimental results further show that our consistency-based ensemble performs model selection effectively by assigning higher authorities (or weights) to the target predictions that are more suitable for adaptation.

We also study how different CRMA components affect the feature distributions via t-SNE visualization [43]. We extract visual features before the last fully connected layer and Fig. 5 shows the feature distributions for source domain MNIST (red-color points) and target domain MNIST-M (blue-color points). The feature visualization aligns well with the Ablation Study in Table 5. Specifically, the source and target domain features are initially not well aligned as in (a) but the alignment is improved clearly when IntraDA, InterDA, or AST is incorporated, respectively, as in (b), (c), and (d). When all three components are incorporated, we get the best feature alignment as in (e).

In addition, to demonstrate model interpretability, we apply the Grad-CAM

[44] algorithm to generate class activation maps that indicate important regions in the input that lead to predictions. As illustrated in Fig. 6, by comparing the heat maps in the second row (without domain adaptation) and the third row (with CRMA), we observe that CRMA could shift the model attention to more discriminative regions within the image for the desired classification task through domain adaptation.

5. Conclusion

This paper presents a concise yet effective method that exploits consistency-regularized self-training for multi-source unsupervised domain adaptation. For each source domain, we train a pair of domain-specific classifiers to perform intra-domain alignment based on the intra-domain consistency of target predictions. In addition, we compute the mean prediction of domain-specific classifier pairs to perform inter-domain alignment by maximizing the inter-domain consistency of all classifiers. As different source domains have different similarities with the target domain, we design an adaptive pseudo label generation technique that predicts target labels by weighted averaging the mean predictions of multiple source domains. Extensive experiments show that our method obtains superior accuracy consistently across all three widely studied datasets on multi-domain unsupervised domain adaptation.

The proposed method effectively addresses the multi-source domain adaptive classification problem with a small overhead on top of a base network model, and it could be applied to various classification or segmentation tasks when annotations are scarce or unavailable in the target domain. However, it relies on multiple classifiers and their decision boundaries for feature alignment, which limits its adaptability to more complicated tasks such as object detection. We will continue to study more generic domain adaptation techniques that can work with minimal task-specific designs in our future works.

References

- [1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
- [2] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587.
- [3] J. Huang, D. Guan, A. Xiao, S. Lu, Cross-view regularization for domain adaptive panoptic segmentation, arXiv preprint arXiv:2103.02584.
- [4] S. Ren, K. He, R. Girshick, J. Sun, Faster r-CNN: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2017) 1137–1149. doi:10.1109/tpami.2016.2577031.
- [5] G. Zhang, Z. Luo, K. Cui, S. Lu, Meta-detr: Few-shot object detection via unified image-level meta-learning, arXiv preprint arXiv:2103.11731.
- [6] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, T. Darrell, Deep domain confusion: Maximizing for domain invariance, arXiv preprint arXiv:1412.3474.
- [7] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, W. Zuo, Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2272–2281.
- [8] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International conference on machine learning, PMLR, 2015, pp. 1180–1189.
- [9] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7167–7176.

- [10] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, T. Mei, Transferrable prototypical networks for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2239–2247.
- [11] R. Xu, Z. Chen, W. Zuo, J. Yan, L. Lin, Deep cocktail network: Multi-source unsupervised domain adaptation with category shift, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3964–3973.
- [12] S. Zhao, B. Li, X. Yue, Y. Gu, P. Xu, R. Hu, H. Chai, K. Keutzer, Multi-source domain adaptation for semantic segmentation, in: Advances in Neural Information Processing Systems, 2019, pp. 7287–7300.
- [13] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, K. Keutzer, Multi-source distilling domain adaptation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 12975–12983.
- [14] H. Wang, M. Xu, B. Ni, W. Zhang, Learning to combine: Knowledge aggregation for multi-source domain adaptation, in: European Conference on Computer Vision, Springer, 2020, pp. 727–744.
- [15] H. Wang, W. Yang, Z. Lin, Y. Yu, Tmda: Task-specific multi-source domain adaptation via clustering embedded adversarial training, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 1372–1377.
- [16] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, G. J. Gordon, Adversarial multiple source domain adaptation, in: Advances in neural information processing systems, 2018, pp. 8559–8570.
- [17] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: Proceedings of the IEEE

Conference on Computer Vision and Pattern Recognition, 2018, pp. 3723–3732.

- [18] S. J. Pan, Q. Yang, A survey on transfer learning, *IEEE Transactions on knowledge and data engineering* 22 (10) (2009) 1345–1359.
- [19] C. Han, Y. Lei, Y. Xie, D. Zhou, M. Gong, Visual domain adaptation based on modified a- distance and sparse filtering, *Pattern Recognition* (2020) 107254doi:10.1016/j.patcog.2019.106996.
- [20] Y. Yao, Y. Zhang, X. Li, Y. Ye, Discriminative distribution alignment: A unified framework for heterogeneous domain adaptation, *Pattern Recognition* 101 (2020) 107165. doi:10.1016/j.patcog.2019.107165.
- [21] L. Zuo, M. Jing, J. Li, L. Zhu, K. Lu, Y. Yang, Challenging tough samples in unsupervised domain adaptation, *Pattern Recognition* 110 107540. doi:10.1016/j.patcog.2020.107540.
- [22] M. M. Rahman, C. Fookes, M. Baktashmotlagh, S. Sridharan, Correlation-aware adversarial domain adaptation and generalization, *Pattern Recognition* 100 (2020) 107124. doi:10.1016/j.patcog.2019.107124.
- [23] J. Huang, S. Lu, D. Guan, X. Zhang, Contextual-relation consistent domain adaptation for semantic segmentation, in: *European Conference on Computer Vision*, Springer, 2020, pp. 705–722.
- [24] J. Zhang, J. Huang, Z. Luo, G. Zhang, S. Lu, Da-detr: Domain adaptive detection transformer by hybrid attention, *arXiv preprint arXiv:2103.17084*.
- [25] J. Liang, R. He, Z. Sun, T. Tan, Exploring uncertainty in pseudo-label guided unsupervised domain adaptation, *Pattern Recognition* 96 (2019) 106996. doi:10.1016/j.patcog.2019.106996.
- [26] J. Yang, R. Yan, A. G. Hauptmann, Cross-domain video concept detection using adaptive svms, in: *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 188–197.

- [27] Q. Sun, R. Chattopadhyay, S. Panchanathan, J. Ye, A two-stage weighting framework for multi-source domain adaptation, in: Advances in neural information processing systems, 2011, pp. 505–513.
- [28] Z. Xu, S. Sun, Multi-source transfer learning with multi-view adaboost, in: International conference on neural information processing, Springer, 2012, pp. 332–339. doi:10.1007/978-3-642-34487-9_41.
- [29] H. Liu, M. Shao, Y. Fu, Structure-preserved multi-source domain adaptation, in: 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 1059–1064.
- [30] J. Hoffman, M. Mohri, N. Zhang, Algorithms and theory for multiple-source adaptation, in: Advances in Neural Information Processing Systems, 2018, pp. 8246–8256.
- [31] H. Zhao, S. Zhang, G. Wu, G. J. Gordon, et al., Multiple source domain adaptation with adversarial learning, In ICLR.
- [32] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, B. Wang, Moment matching for multi-source domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1406–1415.
- [33] S. Kullback, R. A. Leibler, On information and sufficiency, The annals of mathematical statistics 22 (1) (1951) 79–86.
- [34] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
- [35] J. J. Hull, A database for handwritten text recognition research, IEEE Transactions on pattern analysis and machine intelligence 16 (5) (1994) 550–554.
- [36] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, arXiv preprint arXiv:1409.7495.

- [37] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning.
- [38] D. Li, Y. Yang, Y.-Z. Song, T. M. Hospedales, Deeper, broader and artier domain generalization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 5542–5550.
- [39] M. Long, Y. Cao, J. Wang, M. I. Jordan, Learning transferable features with deep adaptation networks, arXiv preprint arXiv:1502.02791.
- [40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
- [41] Y. Zhu, F. Zhuang, D. Wang, Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5989–5996.
- [42] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983.
- [43] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.
- [44] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.