Multi-Source domain adaptation via supervised contrastive learning and confident consistency regularization

Marin Scalbert[1, 2]
marin.scalbert@centralesupelec.fr
Maria Vakalopoulou[1]
maria.vakalopoulou@centralesupelec.fr
Florent Couzinié-Devy[2]
f.couzinie-devy@vitadx.com

[1] MICS
CentraleSupélec, Paris Saclay
Gif-sur-Yvette, France

[2] VitaDX International
Paris, France

Abstract

Multi-Source Unsupervised Domain Adaptation (multi-source UDA) aims to learn a model from several labeled source domains while performing well on a different target domain where only unlabeled data are available at training time. To align source and target features distributions, several recent works use source and target explicit statistics matching such as features moments or class centroids. Yet, these approaches do not guarantee class conditional distributions alignment across domains. In this work, we propose a new framework called Contrastive Multi-Source Domain Adaptation (CMSDA) for multi-source UDA that addresses this limitation. Discriminative features are learned from interpolated source examples via cross entropy minimization and from target examples via consistency regularization and hard pseudo-labeling. Simultaneously, interpolated source examples are leveraged to align source class conditional distributions through an interpolated version of the supervised contrastive loss. This alignment leads to more general and transferable features which further improves the generalization on the target domain. Extensive experiments have been carried out on three standard multi-source UDA datasets where our method reports state-of-the-art results. Our code is available at https://gitlab.com/vitadx/articles/cmsda.

1 Introduction

The performances of deep learning models are known to drop when training and testing data have different distributions. This phenomenon, known as *Domain Shift*, has led to the emergence of the Unsupervised Domain Adaptation (UDA) problem that has been intensively studied over the recent years [8, 14, 17, 18, 19, 30, 33, 35, 42, 49]. Single-source UDA aims to leverage labeled examples from a source domain and unlabeled examples from a target domain to learn a model that performs well on unseen target examples. In the most practical scenario, to collect as much labeled data as possible, several source domains are considered rather than a single one. In such case, the setting is referred to as multi-source UDA.

To solve the UDA problem, most of the methods learn discriminative features from labeled source data and exploit unlabeled target data to align source and target distributions. Source and target distributions alignment is performed so as to maintain the discriminative power of the model on the target domain. Plethora of UDA methods, in the single-source [17, 18, 19, 33, 35, 49] or multi-source settings [25, 45, 50] have tried to align source and target marginal distributions. However, these methods are susceptible to fail if source and target class conditional distributions are not aligned [7]. Alignment of source and target class conditional distributions can be achieved through adversarial based methods such as [7] but they tend to be cumbersome to train while the alignment of the domains can fail if pseudo labels on target examples are noisy. To align source and target class conditional distributions, [42] has proposed to match source and target class centroids. Nevertheless, in order to estimate accurately the true class centroids, batches should be carefully designed to contain enough examples per class while maintaining a well-tuned moving average of centroids. This method assumes also that a single centroid can represent the whole distribution in a class which is a wrong assumption in case of multimodal class distributions.

In this work, to address the problem of multi-source UDA and align efficiently source class conditional distributions, we introduce a new framework named Contrastive Multi-Source Domain Adaptation (CMSDA). CMSDA learns discriminative features on source examples via cross entropy minimization and aligns class conditional distributions of all source domains through supervised contrastive loss. Source class conditional distributions alignment leads to more general and transferable features for the target domain. In the same time, the model adjusts to the target domain via hard pseudo labeling and consistency regularization. To further enhance the robustness, the calibration of our model and enable deeper exploration of the input space, MixUp [48] is leveraged on source examples. Interpolating examples from different source domains can even be seen as way to mix source domains styles. Since MixUp is performed on source domains, interpolated versions of the cross entropy and supervised contrastive losses are used in the final objective. To sum up, our contributions are the following: **(1)** we design a novel tailored end-to-end architecture that maps the different domains to a common latent space, and efficiently transfers knowledge learned on source domains to the target domain using recent advances of supervised contrastive learning, semi-supervised learning and mixup training; **(2)** we show for the first time that supervised contrastive learning and its interpolated extension can be used in the context of domain adaptation for source class conditional distributions alignment leading to higher accuracy for target domains with large domain shift, **(3)** we report state of the art results on three standard multi-source UDA datasets.

2 Related works

Multi-source UDA. In the multi-source UDA setting, former methods such as MDAN [50] have tried to extend a single-source UDA method to the multi-source setting. In MDAN, each source and target marginal distributions are aligned via an adversarial based method [9]. In DCTN [45], an adversarial based method is proposed to align marginally each source with the target domain. Additionally, perplexity scores, measuring the probabilities that a target sample belongs to the different source domains, are used to weight predictions of different source classifiers. M^{3}SDA-β [25] uses a two-steps approach combining an ensemble of source classifiers. In the first step, the method aligns marginally each source with the target domain and also each pair of source domains by matching first order moments of features maps channels. In the second step, to enhance distributions alignment, the different source classifiers are trained following an adversarial method [27]. CMSS [46] exploits an original adversarial approach that selects dynamically the source domains and examples that are the most suitable for aligning source and target distributions. DAEL [52] combines a collaborative training of an ensemble of source expert classifiers with hard pseudo labeling and consistency regularization on the target domain. For source examples, robust features are learned by ensuring consistency between the expert source classifier and an ensemble of non-expert source classifiers. For unlabeled target examples, since no expert target classifier is available, consistency is ensured between the most confident source expert classifier and the ensemble of remaining classifiers. Our method shares some similarities with DAEL as it exploits similar semi-supervised learning techniques to learn on unlabeled target data. However, ours adds an additional constraint to align source class conditional distributions. Recent works are also focusing on improving UDA methods by using specific data augmentation. For instance, MixUp has been explored for single source UDA methods [40, 43] but the literature on methods exploiting MixUp for problems such as multi-source UDA and multi-source Domain Generalization is still very sparse [20, 38]. Our method exploits MixUp but also an interpolated version of the supervised contrastive loss to work with the soft labels produced by MixUp. Multi-source UDA methods have also explored other types of approaches [16, 37, 51, 54], different UDA settings [26] or other data modalities [12].

Contrastive learning. Recently, representation learning has known major breakthroughs due to the advance in contrastive learning [4, 5, 6, 11, 13, 22]. The main idea behind most of contrastive learning methods is that similar examples should share the same representation. For example, in SimCLR [5] the loss enforces pairs of augmented versions of the same image (positives) to have the same representation while having dissimilar representations from all other examples in the batch (negatives). Recently, the loss introduced in [5] has been extended to the supervised setting. In the supervised contrastive loss, examples belonging to the same class are pushed closer while examples from other classes are pushed apart. There have been some attempts to adapt contrastive learning in the case of single-source UDA [14]. However, supervised contrastive learning has yet to be explored on the multi-source UDA problem, even if it seems a natural way to align source class conditional distributions.

Semi-supervised learning. To exploit unlabeled examples, common semi-supervised approaches are based either on consistency regularization [2, 21, 31, 41] or pseudo labeling [1]. Consistency regularization learns on unlabeled data by relying on the assumption that the model should output similar predictions when perturbed versions of the same image are presented. Hard pseudo-labeling consists of using hard predictions on unlabeled examples as ground truth labels for these examples. Some semi-supervised methods such as FixMatch [29] use both approaches. In FixMatch, strongly and weakly augmented images are generated from the same unlabeled image. The network is then trained to ensure consistency between the prediction on the strongly augmented image and the hard pseudo label of the weakly augmented image. To handle potential false pseudo labels, only confident pseudo-labeled examples contribute to the loss. In our method, FixMatch is leveraged to learn on unlabeled target data.

3 CMSDA Framework

In the setting of multi-source UDA, we are given S different source domains $\{\mathcal{D}_1, \ldots, \mathcal{D}_S\}$ and one target domain $\mathcal{D}_T$. Each of the source domain $\mathcal{D}_i$ contains $n_{\mathcal{D}_i}$ labeled examples

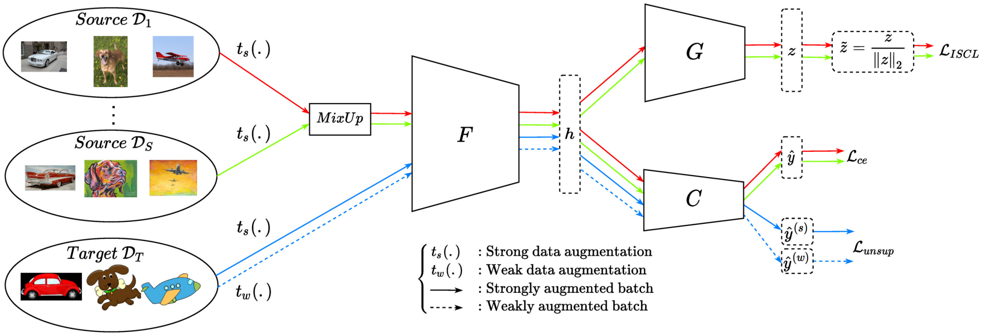


Figure 1: **CMSDA framework.** CMSDA contains 3 shared components: a features extractor F, a projection head G and a classification head C. Source examples are strongly augmented via $t_s(.)$ and interpolated using MixUp. Target examples are weakly and strongly augmented via $t_w(.)$ and $t_s(.)$. All examples are fed to F to produce the embeddings $\boldsymbol{h}$. In the top branch, source embeddings $\boldsymbol{h}$ are given to G to produce the representations $\tilde{\boldsymbol{z}}$ which are used to align source class conditional distributions via minimization of $\mathcal{L}_{ISCL}$. In the bottom branch, all embeddings $\boldsymbol{h}$ are fed to C to produce probability vectors $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{y}}^{(s)}$ and $\hat{\boldsymbol{y}}^{(w)}$. These probability vectors are used to learn discriminative features via minimization of $\mathcal{L}_{ce}$ and $\mathcal{L}_{unsup}$.

$\{(\boldsymbol{x_j}, \boldsymbol{y_j}) \mid 1 \leq j \leq n_{\mathcal{D}_i}\}$ while target domain contains $n_{\mathcal{D}_T}$ unlabeled examples $\{\boldsymbol{x_j} \mid 1 \leq j \leq n_{\mathcal{D}_T}\}$. The goal of multi-source UDA is to learn a robust model from the S labeled source domains $\mathcal{D}_1, \ldots, \mathcal{D}_S$ and the target domain $\mathcal{D}_T$ so that it generalizes well on unseen target domain examples.

3.1 Model architecture

Our model architecture (Figure 1) is shared for both source and target domains. It is composed of three different parts:

Features extractor. The features extractor F is a convolutional neural network. It takes an input image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times 3}$ and returns a features vector $\boldsymbol{h} \in \mathbb{R}^{d_1}$.

Projection head. The projection head G takes as input the representation $\boldsymbol{h}$ and outputs a lower dimensional representation $\boldsymbol{z} \in \mathbb{R}^{d_2}$ with $d_1 > d_2$. Similarly to [15], G is a multi-layer perceptron consisting in two fully connected layers. The first layer preserves the dimension while the second reduces the dimensions from d_1 to d_2. Previous self-supervised contrastive learning methods [4, 6, 11, 13] indicate that the use of a batch normalization layer after the first fully connected layer has shown to generate more powerful representations. Following these findings, we include a batch normalization after the first fully connected layer of G.

Classification head. The classification head C is responsible for the final predictions. In previous self-supervised [5] or supervised [15] contrastive learning methods, the projection head is usually removed after the training and a linear classifier is fine-tuned on top of the frozen representation $\boldsymbol{h}$. Indeed, in practice, $\boldsymbol{h}$ provides better representations than $\boldsymbol{z}$ for the final classification task [5]. Therefore, C is a single fully connected layer taking as input the features extractor representations $\boldsymbol{h}$ and outputing a probability vector $\hat{\boldsymbol{y}} \in [0,1]^K$ where K indicates the number of classes.

3.2 Optimization Strategy

Source domains interpolation with MixUp. MixUp [48] performs data augmentation by creating new examples $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ as convex combinations of random pairs of examples and their corresponding labels $(\boldsymbol{x_a}, \boldsymbol{y_a})$ and $(\boldsymbol{x_b}, \boldsymbol{y_b})$:

$$\begin{cases} \tilde{\boldsymbol{x}} = \lambda \boldsymbol{x_a} + (1-\lambda)\boldsymbol{x_b} \\ \tilde{\boldsymbol{y}} = \lambda \boldsymbol{y_a} + (1-\lambda)\boldsymbol{y_b} \end{cases} \tag{1}$$

where $\lambda \sim Beta(\alpha, \alpha)$. In CMSDA, MixUp is applied on source examples right after strong data augmentation $t_s(.)$.

Overall objective. The overall objective includes three different losses: the interpolated cross entropy loss $\mathcal{L}_{ce}$, the interpolated supervised contrastive loss $\mathcal{L}_{ISCL}$ and the unsupervised FixMatch loss $\mathcal{L}_{unsup}$. The final objective minimized by the model can be written:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{ISCL} + \lambda_2 \mathcal{L}_{unsup} \tag{2}$$

λ_1 and λ_2 are hyperparameters weighting the contributions of the losses $\mathcal{L}_{ISCL}$ and $\mathcal{L}_{unsup}$.

Interpolated cross-entropy. In order to leverage the interpolated labeled source examples obtained after MixUp, our framework minimizes an interpolated cross entropy loss denoted as $\mathcal{L}_{ce}$. For a single interpolated source example $(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}, \tilde{\boldsymbol{y}}_{\boldsymbol{i}}) = (\lambda \boldsymbol{x_a} + (1-\lambda)\boldsymbol{x_b}, \lambda \boldsymbol{y_a} + (1-\lambda)\boldsymbol{y_b})$ with $\hat{\boldsymbol{y}}$ the prediction on the example $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}$, the interpolated cross entropy per sample denoted $\mathcal{L}_i^{ce}$ can be written:

$$\mathcal{L}_i^{ce} = H(\lambda \boldsymbol{y_a} + (1-\lambda)\boldsymbol{y_b}, \hat{\boldsymbol{y}}) \tag{3}$$

$H(\boldsymbol{p}, \boldsymbol{q})$ denotes the cross entropy between a reference distribution $\boldsymbol{p}$ and an approximated distribution $\boldsymbol{q}$. $\mathcal{L}_{ce}$ is then computed by averaging the loss per sample on N_S interpolated source examples.

Interpolated supervised contrastive loss. To align source class conditional distributions, supervised contrastive loss (SCL) seems a simple and straightforward option. When minimizing SCL, representations of examples belonging to the same class are pulled together while representations of examples belonging to other classes are pushed away. In the case of multiple source domains, SCL would force learned features to be domain invariant. Given an example $(\boldsymbol{x_i}, \boldsymbol{y_i})$ with normalized projection head representation $\tilde{\boldsymbol{z}}_{\boldsymbol{i}}$, the per sample SCL is defined by:

$$\mathcal{L}^{SCL}(\tilde{\boldsymbol{z}}_{\boldsymbol{i}}, \boldsymbol{y}_i) = -\frac{1}{|P(i, \boldsymbol{y}_i)|} \sum_{p \in P(i, \boldsymbol{y}_i)} \log \left(\frac{e^{\frac{\tilde{z}_i \cdot \tilde{z}_p}{T}}}{\sum_{j \in A(i)} e^{\frac{\tilde{z}_i \cdot \tilde{z}_j}{T}}} \right) \text{ where: } \begin{cases} A(i) = \{1, \ldots, N_S\} \setminus \{i\} \\ P(i, \boldsymbol{y}_i) = \{j \in A(i) \mid \boldsymbol{y_j} = \boldsymbol{y_i}\} \end{cases} \tag{4}$$

Here, T corresponds to a temperature hyperparameter, $A(i)$ stands for the anchors set (indexes of examples different than i) and $P(i, \boldsymbol{y}_i)$ the positives set (indexes of other examples whose label is equal to $\boldsymbol{y}_i$).

However, by using MixUp on source examples, examples with soft labels are produced whereas SCL requires hard labels. Indeed, SCL needs hard labels so that examples with same labels can be identified and be pulled together. To circumvent the soft labels problem raised by MixUp, we apply an interpolated version of SCL (ISCL) introduced in [23] and minimize it on N_S interpolated source examples.

Given an interpolated source example $(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}, \tilde{\boldsymbol{y}}_{\boldsymbol{i}}) = (\lambda \boldsymbol{x}_{\boldsymbol{a}} + (1-\lambda)\boldsymbol{x}_{\boldsymbol{a}}, \lambda \boldsymbol{y}_{\boldsymbol{a}} + (1-\lambda)\boldsymbol{y}_{\boldsymbol{b}})$ with $\tilde{\boldsymbol{z}}_{\boldsymbol{i}}$ the normalized projection head representation of the example $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}$, the per sample ISCL denoted $\mathcal{L}_i^{ISCL}$ is defined by:

$$\mathcal{L}_i^{ISCL} = \lambda \mathcal{L}^{SCL}(\tilde{\boldsymbol{z}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{a}}) + (1-\lambda)\mathcal{L}^{SCL}(\tilde{\boldsymbol{z}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{b}}) \tag{5}$$

$\mathcal{L}^{SCL}(\tilde{\boldsymbol{z}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{a}})$ ($\mathcal{L}^{SCL}(\tilde{\boldsymbol{z}}_{\boldsymbol{i}}, \boldsymbol{y}_{\boldsymbol{b}})$) stands for the per sample SCL for the example $\boldsymbol{x}_{\boldsymbol{i}}$ with label $\boldsymbol{y}_{\boldsymbol{a}}$ ($\boldsymbol{y}_{\boldsymbol{b}}$). The definition of $P(i, \boldsymbol{y}_{\boldsymbol{i}})$ in Equation 4 implies that the examples in $A(i)$ have hard labels. Therefore, as in [23], for each interpolated example in $A(i)$, we consider as hard label the dominant label which is the one associated to the highest mixing coefficients $(\lambda, 1-\lambda)$. Finally, $\mathcal{L}_{ISCL}$ is computed by averaging the loss per sample on N_S interpolated source examples.

Consistency regularization and hard-pseudo labeling. In our framework, given N_T unlabeled examples drawn from the target domain, we apply a weak augmentation $t_w(.)$ and a strong augmentation $t_s(.)$ on each example to obtain a weakly augmented example $\boldsymbol{x}^{(w)}$ and strongly augmented example $\boldsymbol{x}^{(s)}$. $\boldsymbol{x}^{(w)}$ and $\boldsymbol{x}^{(s)}$ are fed to F and C to obtain respectively the predictions $\hat{\boldsymbol{y}}^{(w)}$ and $\hat{\boldsymbol{y}}^{(s)}$. The hard prediction on the weakly augmented example denoted $\arg\max \hat{\boldsymbol{y}}^{(w)}$ is used as a hard pseudo label[1] while we ensure consistency between the predictions on the strongly example $\hat{\boldsymbol{y}}^{(s)}$ and the hard pseudo label of the weakly augmented example $\arg\max \hat{\boldsymbol{y}}^{(w)}$. As described in Section 2, to discard potential false pseudo labels, only pseudo-labeled weakly augmented examples with a maximum predicted probability above some fixed probability threshold τ contribute to the loss. This corresponds to minimizing the unsupervised loss term of [29] defined by:

$$\mathcal{L}_{unsup} = \frac{1}{N_T} \sum_{i=1}^{N_T} \mathbb{1}_{\{\max \hat{\boldsymbol{y}}^{(w)} > \tau\}} H\left(\arg\max \hat{\boldsymbol{y}}^{(w)}, \hat{\boldsymbol{y}}^{(s)}\right) \tag{6}$$

4 Experiments

4.1 Evaluation

We evaluate and compare our method on three standard multi-source UDA datasets: DomainNet, MiniDomainNet and Office-Home. For each dataset and target domain, two standard baselines commonly used in the context of multi-source UDA [25, 45, 46, 52] have been added: *Source-only* and *Oracle*. *Source-only* represents a model trained only on source examples with standard cross-entropy whereas *Oracle* represents a model trained with labeled target examples. Performances on MiniDomainNet and DomainNet are averaged over three runs with different random seeds. The performances of our method along with the compared multi-source UDA methods for the datasets DomainNet, MiniDomainNet and Office-Home are respectively reported on Table 1, Table 2 and Table 3. For easier interpretation, first and second best methods are respectively highlighted in **bold red** and *italic blue*. Datasets information and implementation details can be found in the supplementary material.

[1] Similar to [29], for simplicity, we assume that $\arg\max$ applied on a K dimensional probability vector gives a valid K dimensional one-hot vector.

DomainNet. Our method achieves the best performance with 50.42% average accuracy which corresponds to $+1.72\%$ gain over the previous state of the art. Overall, our method reports the best performances on 4 out of 6 target domains and the second best performance on one of the two other target domains. On the *Quickdraw* domain, our framework achieves the best performance by a large margin $(+2.40\%)$ compared to the second best method (SImpAl). On this challenging target domain, only SImpaL, CMSS, LtC-MSDA and our method are not subject to negative transfer [24] (lower performances than *Source-only* baseline). This indicates that our method is able to work even for complex target domains.

MiniDomainNet. Our method achieves the best overall accuracy with 61.90%, the best accuracy on 3 out of 4 target domains and the second best on the last domain.

Office-Home. Our method achieves the best overall accuracy with 76.60%. More specifically, CMSDA obtains the best/second best accuracy on 2 out of 4 target domains and competitive results on the two other target domains.

	Target domain						
Methods	*Clp*	*Inf*	*Pnt*	*Qdr*	*Rel*	*Skt*	*Avg*
Source-only [52]	47.60 ± 0.52	13.00 ± 0.41	38.10 ± 0.45	13.30 ± 0.39	51.90 ± 0.85	33.70 ± 0.54	32.90
Oracle [52]	69.30 ± 0.37	34.50 ± 0.42	66.30 ± 0.67	66.80 ± 0.51	80.10 ± 0.59	60.70 ± 0.48	63.00
DANN [10]	45.50 ± 0.59	13.10 ± 0.41	37.00 ± 0.69	13.20 ± 0.77	48.90 ± 0.65	31.80 ± 0.62	32.60
DCTN [45]	48.60 ± 0.73	23.50 ± 0.59	48.80 ± 0.63	7.20 ± 0.46	53.50 ± 0.56	47.30 ± 0.47	38.20
MCD [27]	54.30 ± 0.64	22.10 ± 0.70	45.70 ± 0.63	7.60 ± 0.49	58.40 ± 0.65	43.50 ± 0.57	38.50
M^{3}SDA-β [25]	58.60 ± 0.53	26.00 ± 0.89	52.30 ± 0.55	6.30 ± 0.58	62.27 ± 0.51	49.50 ± 0.76	42.60
CMSS [46]	64.20 ± 0.18	*28.00 ± 0.20*	53.60 ± 0.39	16.00 ± 0.12	63.40 ± 0.21	53.80 ± 0.35	46.50
LtC−MSDA [37]	63.10 ± 0.50	**28.70 ± 0.70**	56.10 ± 0.50	16.30 ± 0.50	66.10 ± 0.60	53.80 ± 0.60	47.40
SImpAl$_{101}$ [34]	66.40 ± 0.80	26.50 ± 0.50	56.60 ± 0.70	*18.90 ± 0.80*	*68.00 ± 0.50*	55.50 ± 0.30	48.60
DAEL [52]	*70.80 ± 0.14*	26.50 ± 0.13	*57.40 ± 0.28*	12.20 ± 0.70	65.00 ± 0.23	**60.60 ± 0.25**	*48.70*
Ours	**70.95 ± 0.23**	26.58 ± 0.34	**57.56 ± 0.08**	**21.30 ± 0.11**	**68.12 ± 0.22**	*59.48 ± 0.07*	**50.42**

Table 1: Accuracy (%) on DomainNet (*Clp*: *Clipart*, *Inf*: *Infograph*, *Pnt*: *Painting*, *Qdr*: *Quickdraw*, *Rel*: *Real*, *Skt*: *Sketch*, *Avg*: *Average*).

	Target domain				
Methods	*Clp*	*Pnt*	*Rel*	*Skt*	*Avg*
Source-only [52]	63.44 ± 0.76	49.92 ± 0.71	61.54 ± 0.08	44.12 ± 0.31	54.76
Oracle [52]	72.59 ± 0.30	60.53 ± 0.74	80.47 ± 0.34	63.44 ± 0.15	69.26
DANN [10]	65.55 ± 0.34	46.27 ± 0.71	58.68 ± 0.64	47.88 ± 0.54	54.60
DCTN [45]	62.06 ± 0.60	48.79 ± 0.52	58.85 ± 0.55	48.25 ± 0.32	54.49
MCD [27]	62.91 ± 0.67	45.77 ± 0.45	57.57 ± 0.33	45.88 ± 0.67	53.03
M^{3}SDA-β [25]	64.18 ± 0.27	49.05 ± 0.16	57.70 ± 0.24	49.21 ± 0.34	55.03
MME [28]	68.09 ± 0.16	47.14 ± 0.32	63.33 ± 0.16	43.50 ± 0.47	55.52
DAEL [52]	*69.95 ± 0.52*	**55.13 ± 0.78**	*66.11 ± 0.14*	*55.72 ± 0.79*	*61.73*
Ours	**71.38 ± 0.65**	*53.76 ± 0.71*	**66.23 ± 0.08**	**56.24 ± 0.67**	**61.90**

Table 2: Accuracy (%) on MiniDomainNet (*Clp*: *Clipart*, *Pnt*: *Painting*, *Rel*: *Real*, *Skt*: *Sketch*, *Avg*: *Average*).

	Target domain				
Methods	*Art*	*Clp*	*Pct*	*Rel*	*Avg*
Source-only [39]	58.02	57.29	74.26	77.98	66.89
Oracle [39]	71.19	79.16	90.66	85.60	81.65
M^{3}SDA-β [25]	64.05	62.79	76.21	78.63	70.42
SImpAl$_{50}$ [34]	70.80	56.30	80.20	81.50	72.20
MFSAN [54]	**72.10**	62.00	80.30	81.80	74.10
MDAN [50]	68.14	67.04	81.03	82.79	74.75
MDMN [16]	68.67	*67.75*	81.37	*83.32*	75.28
DARN [39]	*70.00*	**68.42**	*82.75*	**83.88**	*76.26*
Ours	*71.49*	67.72	**84.19**	82.99	**76.60**

Table 3: Accuracy (%) on Office-Home (*Art*: *Art*, *Clp*: *Clipart*, *Pct*: *Product*, *Rel*: *Real-World*, *Avg*: *Average*).

Source class conditional distributions alignment. To assess the efficiency of $\mathcal{L}_{ISCL}$ on source class conditional distributions alignment, CMSDA has been trained separately with $\mathcal{L}_{ce}$ and $\mathcal{L}_{ce}+\lambda_1\mathcal{L}_{ISCL}$ for each (sources, target) possible combination. Then, the Calinski-Harabasz index (CH-index) [3], a clustering quality metric, has been computed on the source examples embedding $\boldsymbol{h}$. Using CH-index, we could identify and evaluate if the class conditional distributions from the different source domains are well aligned. For each (sources, target) combination, the CH-indexes with and without $\mathcal{L}_{ISCL}$ are reported on Figure 2a. As expected, when $\mathcal{L}_{ISCL}$ is used in the final objective, the CH-index increases systematically. This suggests that $\mathcal{L}_{ISCL}$ aligns efficiently source class conditional distributions while keeping discriminative features.

4.2 Ablation study

Target domain	*Clp*	*Pnt*	*Rel*	*Skt*
$\mathcal{L}_{ce}$	3.76	4.08	3.66	3.97
$\mathcal{L}_{ce}+\lambda_1\mathcal{L}_{ISCL}$	**4.02**	**4.47**	**3.92**	**4.26**

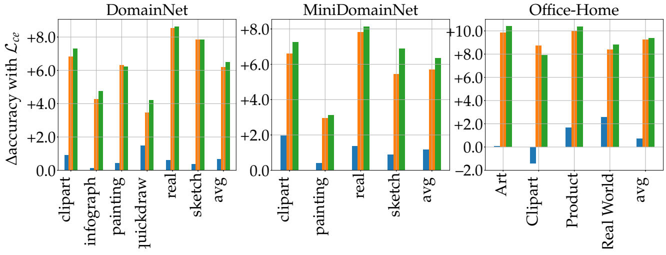
(a)

	Target domain				
Input	*Clp*	*Pnt*	*Rel*	*Skt*	*Avg*
$\boldsymbol{z}$	69.36	53.03	66.03	56.04	61.11
$\boldsymbol{h}$	**71.38**	**53.76**	**66.23**	**56.24**	**61.90**

(b)

	Target domain				
Mixup	*Clp*	*Pnt*	*Rel*	*Skt*	*Avg*
w/o Mixup	70.32	51.75	64.92	51.11	59.52
w Mixup	**71.38**	**53.76**	**66.23**	**56.24**	**61.90**

(c)



(d)

Figure 2: CH indexes for MiniDomainNet considering only the source embeddings $\boldsymbol{h}$ with or without $\mathcal{L}_{ISCL}$ (a), per domain accuracy on MiniDomainNet when classification is performed with $\boldsymbol{h}$ or $\boldsymbol{z}$ (b), with or without MixUp and interpolated losses (c), Δaccuracy compared to standard $\mathcal{L}_{ce}$ for different combination of losses (d).

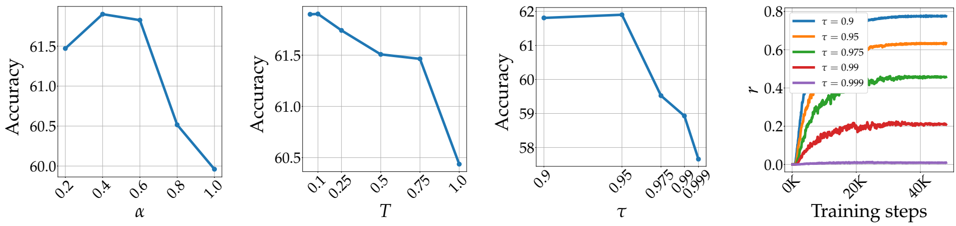
Classification on $\boldsymbol{h}$ or $\boldsymbol{z}$. Similarly to self-supervised methods such as [5], we have investigated the behavior of our model when the classification head takes as input the features extractor representation $\boldsymbol{h}$ or the projection head representation $\boldsymbol{z}$. For each target domain of MiniDomainNet, the framework has been trained by either feeding $\boldsymbol{h}$ or $\boldsymbol{z}$ to the classification head. The accuracies obtained on the different target domains of MiniDomainNet are reported in Figure 2b. Using the representation $\boldsymbol{z}$ instead of $\boldsymbol{h}$ leads most of the time to a small drop of performance. Nevertheless, performances remain competitive compared to other methods in Table 2. For the *Clipart* domain, using $\boldsymbol{z}$ results in a 2% accuracy drop. These performances discrepancies are in adequacy with the findings of [5] arguing that $\boldsymbol{h}$ usually provides better representations for the final classification task. Therefore, even if performances are quite comparable, we recommend feeding $\boldsymbol{h}$ to the classification head C.

MixUp and interpolated losses. To assess the contributions of MixUp and interpolated losses on CMSDA performances, we have trained two different versions of the framework. In the first version, MixUp is removed while standard versions of the cross entropy and supervised contrastive losses are used. Conversely, in the second version, MixUp is applied on source examples and interpolated versions of the cross entropy and the supervised contrastive losses are used. The accuracies for these two versions and for each target domain of MiniDomainNet are reported on Figure 2c. MixUp combined with the interpolated losses reports better performance with a 2% overall accuracy gain. We believe that MixUp applied on examples from different source domains can be seen as a way to mix source domains style and serves as an efficient data augmentation to learn domain invariant features. Additionally, it is known that MixUp improves model calibration and robustness on out-of-distribution data [32]. Therefore, it might enable cleaner pseudo-labels for the target examples.

Loss ablation. To highlight each loss influence on the performances, CMSDA has been trained with different combinations of the losses $\mathcal{L}_{ce}$, $\mathcal{L}_{ISCL}$ and $\mathcal{L}_{unsup}$. In this experiment, hyperparameters remain unchanged and results are averaged over three runs. For each dataset and each losses combination, we report on Figure 2d the gain/loss in terms of accuracy compared to minimizing only $\mathcal{L}_{ce}$. When $\mathcal{L}_{ISCL}$ is combined with $\mathcal{L}_{ce}$ (blue bars), it has in average a positive impact (0.67%, 1.23% and 0.72% respectively for DomainNet (DN), MiniDomainNet (MDN) and OfficeHome (OH)). $\mathcal{L}_{ISCL}$ brings significant gains for target domains with large domain shift (DN-*quickdraw*: $+1.5\%$, MDN-*sketch*:$+1.16\%$ or OH-

Real World: +2.55%). Even if in some rare cases, such as OH-*Clipart*, $\mathcal{L}_{ISCL}$ leads to a small loss of accuracy, its overall contribution is beneficial. When $\mathcal{L}_{unsup}$ is added to $\mathcal{L}_{ce}$ (orange bars), performances are systematically improved. In general, $\mathcal{L}_{unsup}$ contribution is higher than $\mathcal{L}_{ISCL}$. This is consistent with the usual gap in performance observed between methods that exploit target data (multi-source UDA) and the ones that do not (multi-source Domain Generalization). Additionally, when $\mathcal{L}_{ISCL}$ is combined to $\mathcal{L}_{ce}$ and $\mathcal{L}_{unsup}$ (green bars), the performances are often enhanced, especially on the most challenging target domains (DN-*quickdraw*: +0.73%, DN-*infograph*: +0.75%, MDN-*sketch*: +1.44%). The accuracy gains of $\mathcal{L}_{ISCL}$ and $\mathcal{L}_{unsup}$ seem to be additive suggesting that their contributions are independent. All these observations prove the usefulness and the independent contribution of each loss.

4.3 Sensitivity to hyperparameters



(*a*) (*b*) (*c*) (*d*)

Figure 3: Averaged accuracy (%) on MiniDomainNet with respect to α (*a*); T (*b*); τ (*c*). Ratio r of target examples contributing to $\mathcal{L}_{unsup}$ with respect to τ (*d*).

In this section, we assess the behavior of the model with respect to its hyperparameters α, T, and τ. Experiments about λ_1, λ_2 and other source examples mixing strategies (CutMix [47]) are included in the supplementary material.

MixUp hyperparameter α. In Mixup [48], the interpolation parameter $\lambda \in [0,1]$ is drawn such that $\lambda \sim Beta(\alpha, \alpha)$. α controls the interpolation strength. Too low α usually lead to weak regularization while too high α lead to too strong regularization resulting in underfitting and under-confident models [32, 48]. We have investigated how α impacts the performances by training the framework for each target domain of MiniDomainNet with different α. The average accuracy over the target domains with respect to α is reported on Figure 3a. Performances are quite stable for $\alpha \in [0.2, 0.6]$ but start decreasing for $\alpha > 0.6$, validating the observations made in [32, 48]. $\alpha = 0.4$ leads to the best accuracy. We suggest this value as a starting point for other datasets.

Temperature T. The temperature hyperparameter T is known to have a crucial role in self-supervised/supervised contrastive learning [5, 15]. Setting T properly can result in a non negligible gain of performances [15]. According to [36], selecting the optimal T is a compromise between uniformity and tolerance. Uniformity corresponds to the capacity of the representations $\tilde{z}$ to be uniformly distributed over the sphere while tolerance describes how close the representations are for examples in the same class. Uniformity has known to be important to learn separable features however high uniformity induces a decrease of tolerance [36]. $T \to 0$ tends to encourage uniformity whereas $T \to +\infty$ promotes tolerance. To evaluate the effect of T on the performances, we have trained our model for each target domain of MiniDomainNet with different T. The average accuracy over the target domains with respect to T is reported on Figure 3b. Our framework seems to benefit from lower temperatures. Indeed, for $T \in [0.05, 0.1]$, the performances are quite stable and reach the maximum at $T = 0.1$. For $T > 0.1$, the accuracy begins to decrease slowly until $T = 0.75$ and then drops. Overall, it suggests that uniformity is more important than tolerance however too low temperatures ($T < 0.05$) might also harm the model performances. We suggest to use a temperature $T = 0.1$ for further experiments on different datasets.

FixMatch probability threshold τ. In semi-supervised learning, including false pseudo-labeled examples during training can drastically hurt the performances [1]. τ addresses this problem by discarding non confident pseudo-labeled examples from $\mathcal{L}_{unsup}$. More specifically, examples whose maximum predicted probability is below τ do not contribute to $\mathcal{L}_{unsup}$. To investigate how τ affects the number of discarded target examples in $\mathcal{L}_{unsup}$ during training, we have plotted for different τ the ratio r of target examples contributing to $\mathcal{L}_{unsup}$:

$$r = \frac{\sum_{i=1}^{N_T} \mathbb{1}_{\{\max \hat{\boldsymbol{y}}_{\boldsymbol{i}}^{(w)} > \tau\}}}{N_T} \tag{7}$$

This experiment is performed on the target domain *Clipart* of MiniDomainNet and reported on Figure 3d. It reveals that for any value τ, as the training progresses, the model gets more and more confident predictions resulting in an increase of r. A second observation is that when τ is set too low ($\tau = 0.9$), r is in average high at the end of training ($r \sim 77\%$) whereas *Oracle* reaches only 72.59% accuracy. This suggests that false pseudo-labeled target examples contributes to $\mathcal{L}_{unsup}$. On the contrary, when τ is set too high ($\tau = 0.999$), r is in average very low at the end of training ($r \simeq 0.02$). This indicates that a lot of correct pseudo-labeled target examples have been discarded. To evaluate the effect of τ on the performances, CMSDA has been trained with different τ values for each target domain of MiniDomainNet. The average accuracy over target domains with respect to τ is reported in Figure 3c. Performances are stable when choosing $\tau \in [0.9, 0.95]$. $\tau = 0.95$ leads to the highest accuracy. For $\tau < 0.95$, false pseudo labels contribute to $\mathcal{L}_{unsup}$ resulting in a small drop of accuracy. For $\tau > 0.95$, as τ increases, more and more correct pseudo-labeled examples are discarded and performances start to drop.

5 Conclusion

In this work, we have introduced a new framework combining recent advances in supervised contrastive learning and semi-supervised learning to address the problem of multi-source UDA. Our framework, through supervised contrastive learning, is able to align source class conditional distributions resulting in more robust and universal features for the target domain. Simultaneously, the model adjusts to the target domain via hard pseudo labeling and consistency regularization on target examples. Our framework has been evaluated on three datasets commonly used for multi-source UDA and has reported superior results over previous state-of-the-art methods with robust performances even on complex domains where negative transfer can occur.

In future research, we plan to explore the use of supervised contrastive learning on both source and pseudo-labeled target examples so as to align source and target conditional distributions all together. Additionally, we believe that data augmentation strategies usually designed for domain generalization (MixStyle [53], Fourier Based Augmentation [44]) and conditional normalizations could provide interesting directions for our future work.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

[7] Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1416–1425, 2019.

[8] Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9944–9953, 2019.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[12] Jiang Guo, Darsh J Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. *arXiv preprint arXiv:1809.02256*, 2018.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[14] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019.

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[16] Yitong Li, Michael Murias, Samantha Major, Geraldine Dawson, and David E Carlson. Extracting relationships by multi-domain matching. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6799–6810, 2018.

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[18] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *arXiv preprint arXiv:1602.04433*, 2016.

[19] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[20] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 466–483. Springer, 2020.

[21] Aamir Mustafa and Rafal K. Mantiuk. Transformation Consistency Regularization-A Semi-Supervised Paradigm for Image-to-Image Translation. In *Computer Vision – ECCV 2020*, pages 599–615, July 2020.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[23] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. *arXiv preprint arXiv:2012.04462*, 2020.

[24] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[25] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[26] Sayan Rakshit, Dipesh Tamboli, Pragati Shuddhodhan Meshram, Biplab Banerjee, Gemma Roig, and Subhasis Chaudhuri. Multi-source open-set deep adversarial domain adaptation. In *European Conference on Computer Vision*, pages 735–750. Springer, 2020.

[27] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[28] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[29] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[30] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision - ECCV 2016 Workshops*, pages 443–450. Springer, 2016.

[31] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[32] Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *arXiv preprint arXiv:1905.11001*, 2019.

[33] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[34] Naveen Venkat, Jogendra Nath Kundu, Durgesh Kumar Singh, Ambareesh Revanur, and R Venkatesh Babu. Your classifier can secretly suffice multi-source domain adaptation. *arXiv preprint arXiv:2103.11169*, 2021.

[35] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[36] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. *arXiv preprint arXiv:2012.09740*, 2020.

[37] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *European Conference on Computer Vision*, pages 727–744. Springer, 2020.

[38] Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3622–3626. IEEE, 2020.

[39] Junfeng Wen, Russell Greiner, and Dale Schuurmans. Domain aggregation networks for multi-source domain adaptation. In *International Conference on Machine Learning*, pages 10214–10224. PMLR, 2020.

[40] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. Dual mixup regularized learning for adversarial domain adaptation. In *European Conference on Computer Vision*, pages 540–555. Springer, 2020.

[41] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc., 2020.

[42] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *International conference on machine learning*, pages 5423–5432. PMLR, 2018.

[43] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020.

[44] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. *arXiv preprint arXiv:2105.11120*, 2021.

[45] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

[46] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In *Computer Vision - ECCV 2020*, 2020.

[47] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[48] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.

[49] Xu Zhang, Felix Xinnan Yu, Shih-Fu Chang, and Shengjin Wang. Deep transfer network: Unsupervised domain adaptation. *arXiv preprint arXiv:1503.00591*, 2015.

[50] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.

[51] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12975–12983, 2020.

[52] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *arXiv preprint arXiv:2003.07325*, 2020.

[53] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021.

[54] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5989–5996, 2019.