Coupled Training for Multi-Source Domain Adaptation

Ohad Amosy
Bar-Ilan University, Israel
amosyoh@biu.ac.il

Gal Chechik
Bar-Ilan University, Israel
NVIDIA Research, Israel
gal.chechik@biu.ac.il

Abstract

Unsupervised domain adaptation is often addressed by learning a joint representation of labeled samples from a source domain and unlabeled samples from a target domain. Unfortunately, hard sharing of representation may hurt adaptation because of negative transfer, where features that are useful for source domains are learned even if they hurt inference on the target domain. Here, we propose an alternative, soft sharing scheme. We train separate but weakly-coupled models for the source and the target data, while encouraging their predictions to agree. Training the two coupled models jointly effectively exploits the distribution over unlabeled target data and achieves high accuracy on the target. Specifically, we show analytically and empirically that the decision boundaries of the target model converge to low-density "valleys" of the target distribution. We evaluate our approach on four multi-source domain adaptation (MSDA) benchmarks, digits, amazon text reviews, Office-Caltech and images (DomainNet). We find that it consistently outperforms current MSDA SoTA, sometimes by a very large margin.

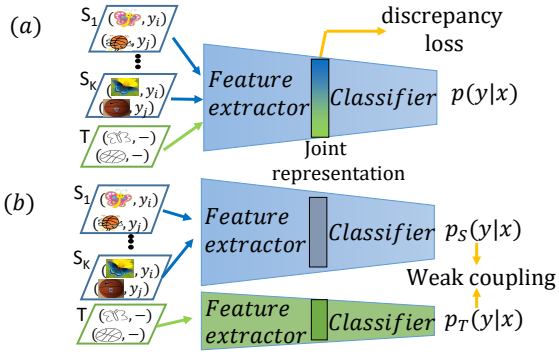

Figure 1. **MUST weak coupling**. (a) Mainstream DA approaches use joint representation of the source and the target domain. The feature extractor is trained to minimize a discrepancy loss (domain adversarial loss or other distance metrics between source and target distribution). The classifier is trained to classify samples from the joint representation using source domain labels. (b) MUST trained the teacher network (blue) on the source domains and the student network (green) on the target domain. Instead of enforcing joint representation, each network uses a different feature space. The models are coupled through their predictions: the student uses the predictions of the teacher as pseudo-labels and the teacher uses the predictions of the student as an extra regularization.

1. Introduction

Multi-source domain adaptation (MSDA) is a fundamental problem in ML with applications to vision [24], audio [20] and text [27]. In unsupervised MSDA, labeled samples are given from multiple source domains and we wish to make predictions on a target domain, from which only unlabeled samples are available [40]. For example, images may be taken under several known lighting conditions, medical data may be collected using different versions of a sensor and product reviews may be collected for different products. In all these cases, we wish to learn from all source domains.

The most widely used approaches to DA, learn a single model from source and target data, see review by [38]. The idea is that both labeled source samples and unlabeled target samples would steer the shared representation towards features that are beneficial to all domains. Unfortunately, hard sharing of representations often suffer from negative transfer [23, 28, 32, 36]. Features that are useful for source domain are emphasized over those useful for the target, and inference on the target data is harmed.

In this paper, we propose an alternative to hard sharing of representations and describe a soft-sharing scheme. Figure 1 illustrates this learning setup and the architecture which we name MUST, for *MUlti-Source Shared Training*. We train separate models for source data and target data while encouraging agreement across their *predictions* rather than their representations. Namely, instead of constraining all models to use the same feature representation, this approach encourages them to make similar predictions on the target domain.

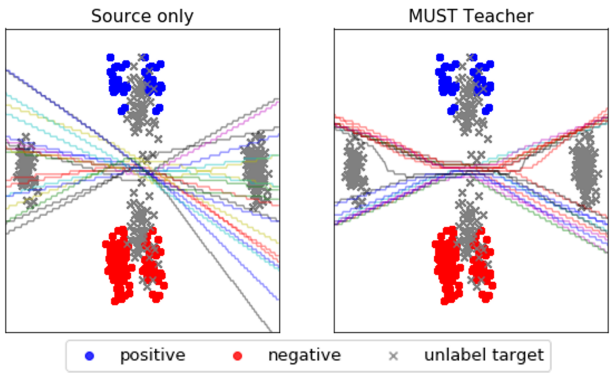

Figure 2. Decision boundaries of source-only model and MUST. Each colored line corresponds to a different initialization. Blue: positive source samples. Red: negative source samples. Gray: unlabeled target samples. The source-only model is trained only on source samples; it classifies perfectly the source data but ignores the target data. The MUST teacher learns to classify the source data and at the same time avoids dense areas of the target distribution as analyzed in lemma 4.1.

The key idea in MUST is to use the target data in supervised training, by learning to generate pseudo-labels that agree with the target distribution. **The target distribution, although unlabeled, contains valuable information** that can be used for classification. Specifically, if the target data is clustered, samples within a cluster often share the same label. As a result, a classifier that separates the data correctly would have its decision boundaries lie in low density regions. This property is known as the *clustering assumption* and it is widely used in semi-supervised methods [35, 9, 31]. We find that the weak coupling employed by MUST allows it to exploits target information. We analyze this weak coupling theoretically and empirically in section 4 and find that the decision boundaries of the student converge to low-density "valleys" of the target distribution (see Figure 2).

MUST can be viewed as a variant of teacher-student (TS) approaches [27, 3, 21, 20]. There, a student network is trained on target pseudo-labels, which are generated by a teacher network trained on the source. Unlike current TS, in our weakly-coupled version, the teacher and the student networks are trained jointly. Also unlike TS, in our approach, the teacher network is tuned such that its predictions on target data are consistent with those of the student network.

This paper makes the following novel contributions. (1) We describe a new training procedure for multi-source DA. (2) We analyze the coupled dynamics of the student and teacher models in our approach, showing empirically and analytically that it tends to converge to the low-density areas of the target distribution. (3) New SoTA on three MSDA classification benchmarks: Digits (MNIST-like), sentiment analysis (Amazon data) and visual object recognition (DomainNet). Compared with an estimated upper bound on classification error, MUST sometimes achieves dramatic reduction in average error rate, as high as 76% on digits dataset.

2. Related work

Single-source domain adaptation (SSDA) has been extensively studied. See [38, 10] recent literature surveys. SSDA can directly apply to multi-source DA, by combining all the source domains to one domain. However, this method leads to poor adaptation performance [24].

Multi-source DA: Compared to the vast amount of research done on single-source DA, multi-source DA (MSDA) is less explored. The theory of MSDA was studied by [1]. They suggested using divergence between source and target domains to find a theoretical bound for generalization error. [18] introduces a new divergence measure and [19] suggested presenting the target hypotheses as a weighted combination of source hypotheses. Inspired by DANN [7], an SSDA approach, [43] proposed multi-source domain adversarial networks (MDAN), an adversarial loss to find a representation indistinguishable between all source domains and the target domain and, at the same time, informative enough for the given task. [26] uses adversarial loss as well as ensemble of encoder and decoder networks to learn a shared feature space to the source and target domains. [39] adapts domain adversarial loss to transformers. [14] uses different batch norm per domain (adaBN), letting the model learn different batch statistics in different domains. Motivated by [19], Deep Cocktail Networks (DCTN) [40] used adversarial learning to minimize the discrepancy between the target and each of the multiple source domains. Multi-Domain Matching Network (MDMN) [13] increases domain similarities not only between the source and target domains but also within the source domain themselves based on a Wasserstein-like measure. Moment matching (M3SDA) [24] minimizes the first order moment-related distance between all source and target domains. Domain aggregation network (DARN) [37] learns to weight the source domains to find the optimal balance between increasing the effective sample size and excluding irrelevant data. Learn to combine (LtC) [34] uses a knowledge graph on the prototypes of various domains to realize the information propagation among semantically adjacent representations. Recently, model-agnostic approaches suggested to improve existing MSDA approaches. [12] uses meta-learning and [42] uses a curriculum agent to choose source training samples. Those approaches can be combined with any model to improve their accuracy. The current paper proposes a new adaptation algorithm. Thus, we compare our method with adaptation approaches.

Reverse Validation: Unlike supervised learning, hy-

perparameters in DA should not be tuned using cross validation, and for two reasons. First, no labels are available for the target domain. Second, using cross-validation on source data is not a good estimator of model performance on target data. To address this issue, [44] proposed using reverse validation (RV) for tuning hyperparameter without target labels. RV is estimated by first splitting the source (labeled) and target (unlabeled) data into training and validation sets. The source training set is used to train a classifier with any UDA method, and infer pseudo-labels over the target validation set. Those pseudo-labels are then used to train a second classifier. That second classifier uses the same UDA method, but with the pseudo-labeled target data as the source domain and the source data, without labels, as the target domain. The classifier is evaluated on the source validation data and its loss is the RV. The parameters that gain the lowest RV are selected. [38] pointed out that many studies incorrectly use target labels for hyperparameter tuning, breaking the very definition of UDA. Results of these studies can be interpreted as upper bounds on method performance perform. We stress that here, we used reverse validation for parameter tuning, adhering to a more realistic training scenario. For comparison, we also provide results achieved using a non-RV evaluation protocol.

Negative transfer: Although negative transfer does not rigorously defined, a widely accepted description of negative transfer is stated as "transferring knowledge from the source can have a negative impact on the target learner" [41]. [36] suggested ways to measure negative transfer, but it focuses on cases where there is at least a small amount of target label data. In section 5 we design an experiment to measure the way different methods deal with negative transfer.

Teacher-student approaches: In teacher-student (TS) approaches a teacher network is trained on the source domain. Then, a student network is trained on the target domain, using the predictions of the teacher networks as labels. Several authors showed that TS architectures are effective for DA. [3] uses pairs of samples from source and target domains, which are frame-by-frame synchronized for speech recognition tasks. The teacher is a trained model on the source domain, and remains constant. In the training process, the teacher and student make predictions on pairs of related source and target samples. The student is trained to minimize the KL divergence between the outputs. [21] uses the same idea as [3], but adds to the student an adversarial objective that encourages it to learn condition-invariant features. [20] also followed [3], but replaces the teacher and student models with an attention-based encoder-decoder. [27] uses one teacher per source domain. Teacher models are first trained in a supervised manner, then remain fixed during adaptation. The student model is then trained to imitate a weighted sum of the teacher predictions. [5] trains a single model (the student) on the source domain, while the weights of the teacher network are set as an exponential moving average of student weights. In contrast with previous TS approaches, we train the teacher and the student *jointly*, making the teacher take into account the (unlabeled) distribution of target samples $p(x)$. Section 6 shows that teacher pseudo-labels on target data are highly variable during learning, so freezing them according to one epoch (the last) leads to poor student performance. Our approach addresses this issue by training both networks iteratively, so the student network is trained on different predictions of the teacher each iteration. In addition, the weak-coupling makes the predictions of the teacher more consistent from one iteration to another.

3. Our approach

Our key idea is to train a network on the target data, and iteratively discovers target pseudo-labels that are consistent with the target distribution. Even though the target domain is unlabeled, it does contain useful information, which lies in the distribution of the data $p(x)$. This is because when data is clustered, samples within a cluster tend to share the same label. As a result, a classifier that separates the data correctly would have its decision boundaries located between the high-density clusters and passing in regions with low sample density. This property is known as the *clustering assumption* and it is widely used in semi-supervised methods [35, 9, 31].

Building on these ideas, MUST trains two separate models. One for source domains which learns from labeled data and another for the target domain which exploits the unlabeled distribution. The models are weakly couples through their predictions, so MUST converge to solutions that fit both the source domains labels and the target distribution. In section 4 we analyze the coupling effect on the target classifier, showing its decision boundaries converge to low-density "valleys" of the target distribution.

We now formally define the learning setup and our approach. Let $\{S^k\}_{i=1}^K$ be a collection of source domains, where the k^{th} domain has N_k labeled samples $\{(x_i^k, y_i^k)\}_{i=1}^{N_k}$. Similarly, T is a target domain, with N_T unlabeled samples $\{z_i\}_{i=1}^{N_T}$. The multi-source DA problem aims to find a hypothesis f, which minimizes the test target error $\epsilon(f) = E_{z\sim T}[p_T(y|z) - p(y|z, f)]$, where $p_T(y|z)$ denotes the conditional distribution of the target domain and $p(y|z, f)$ denotes the conditional distribution of the predicted label.

Our algorithm trains two networks. The teacher network is trained on labeled source samples and the student network is trained on target samples, using the predictions of the teacher as pseudo labels. These two networks are trained jointly, using two losses:

(1) The teacher learns to classify. Train a teacher clas-

Algorithm 1 MUST training procedure

Input: Source domains samples $\{\{(x_i^k, y_i^k)\}_{i=1}^{N_k}\}_{k=1}^{K}$, target domain samples $\{z_i\}_{i=1}^{N_T}$ and a hyperparameter λ
for $t = 1..steps$ **do**
  $src = random(K)$ //Choose random source
  X_{src}, Y_{src}=sample-batch$(\{(x_i^{src}, y_i^{src}\})_{i=1}^{N_{src}})$
  Calculate L_{source} using (1)
  X_{tgt} = sample-batch$(\{z_i\}_{i=1}^{N_T})$
  Calculate $L_{student}$ using (2)
  Update ϕ to minimize $L_{student}$
  Calculate $L_{teacher}$ using (3)
  Update θ to minimize $L_{teacher}$
end for

sifier f_θ using samples from the source domains, by minimizing

$$\mathcal{L}_{source} = \sum_{k=1}^{K} \frac{1}{N_k} \sum_{i=1}^{N_k} l_1(f_\theta(x_i^k), y_i^k), \tag{1}$$

where l_1 is a loss functions.

(2) The teacher trains the student: Use f_θ to give soft labels to the target domain and use them to train the student f_ϕ. For that, we minimize

$$\mathcal{L}_{student} = \frac{1}{N_T} \sum_{i=1}^{N_T} l_2(f_\phi(z_i), f_\theta(z_i)), \tag{2}$$

where l_2 is a loss functions. $L_{student}$ is a function of both the student network f_ϕ and the teacher network f_θ. We used its derivatives w.r.t. the student parameter ϕ to train the student network.

(3) Combining the losses: We linearly combine the teacher loss on the source domain and the student loss on the target domain to get the teacher loss. This time, $L_{student}$ derivatives calculated w.r.t. the teacher parameters θ. This can be viewed as a regularizer of the student network on the teacher, while training the teacher on the source domain. The total teacher loss becomes

$$\mathcal{L}_{teacher} = \mathcal{L}_{source} + \lambda \cdot \mathcal{L}_{student}, \tag{3}$$

where λ is a hyper parameter selected using reverse validation [44]. The training process is summarized in algorithm 1.

4. Analysis

In this section we analyze MUST solutions both analytically and experimentally. We show that MUST converges to solutions with a large margin on the target data.

4.1. Theoretical analysis

The objective of MUST includes a coupling term (2) that encourages the predictions of the teacher network to be similar to those of the student network (in the target data). We now analyze how this weak coupling affects the solutions of the two models.

As discussed above, the clustering assumption suggests that samples in the same cluster tend to have the same label. This suggests that decision boundaries should pass in low density areas of the target distribution.

We now analyze the solutions that MUST converges to.

Lemma 4.1. *Let the teacher network f_θ be a binary classifier parametrized by θ, whose last layer is a sigmoid $f_\theta = \sigma(g_\theta(x))$. Let f_ϕ be the student network. If $\forall z \in T : |\frac{\partial g_\theta(z)}{\partial \theta}| \leq A$ and $|g_\theta(z)| \geq \rho \geq 0$ than using cross entropy for the teacher loss and using L_2 for the student loss, the gradient update bounds by:*

$$\frac{\partial L_{teacher}}{\partial \theta} \leq E_{(x,y)\sim S}[y\frac{\frac{\partial f_\theta}{\partial \theta}(x)}{f_\theta(x)} + (1-y)\frac{-\frac{\partial f_\theta}{\partial \theta}(x)}{1-f_\theta(x)}] + 2\lambda E_{z\sim T}[(f_\theta(z) - f_\phi(z))]\frac{A}{e^\rho}. \tag{4}$$

Proof.

$$L_{teacher} = E_{(x,y)\sim S}[ylogf_\theta(x) + (1-y)log(1-f_\theta(x))] + \lambda E_{z\sim T}[l_2(f_\theta(z), f_\phi(z))]. \tag{5}$$

The gradient decent update rule for the teacher is:

$$\theta^{t+1} = \theta^t + \eta \frac{\partial L_{teacher}}{\partial \theta}, \tag{6}$$

where η is the learning rate.

For any θ, the loss derivative is:

$$\frac{\partial L_{teacher}}{\partial \theta} = E_{(x,y)\sim S}[y\frac{\frac{\partial f_\theta}{\partial \theta}(x)}{f_\theta(x)} + (1-y)\frac{-\frac{\partial f_\theta}{\partial \theta}(x)}{1-f_\theta(x)}] + \lambda E_{z\sim T}[2(f_\theta(z) - f_\phi(z))\frac{\partial f_\theta}{\partial \theta}(z)]. \tag{7}$$

Now, using $\forall z \in T : |\frac{\partial g_\theta(z)}{\partial \theta}| \leq A$ and $|g_\theta(z)| \geq \rho \geq 0$ gives:

$$\frac{\partial f_\theta(z)}{\partial \theta} = \frac{\frac{\partial g_\theta(z)}{\partial \theta}}{2 + e^{-g_\theta(z)} + e^{g_\theta(z)}} \leq \frac{\frac{\partial g_\theta(z)}{\partial \theta}}{e^{g_\theta(z)}} \leq \frac{A}{e^\rho}. \tag{8}$$

Plugging (8) into (7) gives $\frac{\partial L_{teacher}}{\partial \theta} \leq E_{(x,y)\sim S}[y\frac{\frac{\partial f_\theta}{\partial \theta}(x)}{f_\theta(x)}$

$$+(1-y)\frac{-\frac{\partial f_\theta}{\partial \theta}(x)}{1-f_\theta(x)}] + 2\lambda E_{z\sim T}[(f_\theta(z) - f_\phi(z))\frac{A}{e^\rho}].$$

Since $\frac{A}{e^{\rho}}$ is a constant it follows that:

$$\frac{\partial L_{teacher}}{\partial \theta} \leq E_{(x,y)\sim S}[y\frac{\frac{\partial f_\theta}{\partial \theta}(x)}{f_\theta(x)} + (1-y)\frac{-\frac{\partial f_\theta}{\partial \theta}(x)}{1-f_\theta(x)}] + 2\lambda E_{z\sim T}[(f_\theta(z) - f_\phi(z))]\frac{A}{e^{\rho}}. \quad (9)$$

□

This result gives an upper bound for changes in θ. The teacher converges when $\frac{\partial L_{teacher}}{\partial \theta}$ vanishes. If the teacher is in a local minimum with respect to the source data $E_{(x,y)\sim S}[y\frac{\frac{\partial f_\theta}{\partial \theta}(x)}{f_\theta(x)} + (1-y)\frac{-\frac{\partial f_\theta}{\partial \theta}(x)}{1-f_\theta(x)}]$ is 0. $E_{z\sim T}[(f_\theta(z) - f_\phi(z))]\frac{A}{e^{\rho}}$ decreases to zero in one of the following two options: (1) Fitting the student to the teacher predictions on the target data perfectly. This can easily happen when the source and target data have many common features, but for large domain shift this may not be the case. (2) Increasing ρ. $|g_\theta(z)|$ is the distance from the decision boundary, so increasing ρ is achieved by moving the decision boundaries away from the target samples. This way the classifier uses a larger margin for samples that are far from the source domain.

4.2. Empirical analysis

To get a better understanding of the classifier properties learned using MUST, we used an illustrative example. We generated 2D data from 2 classes and trained a 2-layer fully-connected neural network to perform binary classification.

Figure 2 compares source-only model, which is a model that trained only on the source domain and MUST teacher. The decision boundaries visualized for 20 different initializations. The decision boundaries of the MUST teacher avoid highly-dense regions of the target distribution (grey dots), as analyzed in lemma 4.1.

To further assess if this effect also occurs in real datasets, we trained a MUST model with the DomainNet dataset [24]. We estimated the distance of each sample from the decision boundary. Specifically, we perturbed each sample by epsilon in the direction of its adversarial perturbation and counted how many samples changed their labels as a function of epsilon. Figure 3 shows that for the source-only model, many more samples are close to the decision boundary, compared with the MUST model. This suggests that MUST decision boundaries are more distant on average from target samples, and is consistent with the hypothesis that they traverse low-density regions of the target density.

5. Experimental evaluation

We evaluate MUST using three real-world datasets. First, in a task of sentiment analysis by using the Amazon product reviews benchmark [2]. Second, in a task of image classification using four digit recognition datasets (MNIST, MNIST-M, SVHN and SynthDigits) and DomainNet dataset [24]. Code is available at https://github.com/amosy3/MUST.

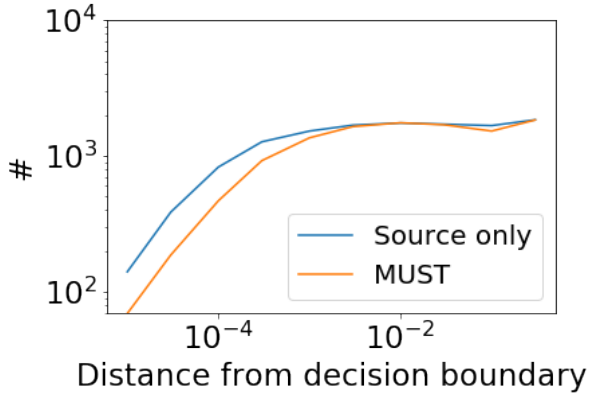

Figure 3. The number of target samples within a specific distance from decision boundary. For each target sample, we move an epsilon step in direction of adversarial perturbation and count the number of samples that change their labels. Calculated on DomainNet.

Implementation details: We use cross entropy as the loss function of Eq. (1). In Eq. (2), training with a cross entropy loss was unstable, because the loss gradients are calculated on both the estimated probability distribution and the true distribution. We found the training to be more stable using a L_2 loss in Eq. (2).

We followed [14] and use different batch norm layers for each domain. This way, the model can effectively use the data from all domains using one model. We also followed [5] and use confidence thresholding, to stabilize the training process. For each target sample, if the teacher softmax layer maximum is lower than the confidence threshold (denoted as C_{th}), the student will not use that sample for training.

5.1. Sentiment analysis

Sentiment analysis aims to decide if the sentiment of a given text is positive or negative. In the MSDA setup, domains are text from a different distribution. For the Amazon reviews dataset different domains are different products.

The data: The Amazon reviews dataset contains 27677 reviews on four kinds of products: Books, DVDs, Electronics and Kitchen appliances. Each review is labeled as positive or negative. For a fair comparison, we followed the experimental setup used by [43] with the code provided by the authors. We conduct four experiments: for each experiment, we pick one product as the target domain and use the rest as source domains.

Experimental setup: We used the same basic network structure as [43]. Since MUST uses a different batch norm

Method	Books	DVD	Elect	Kitn	Avg.
KD [25]	79.2	80.9	85.8	87.3	83.3
mSDA [4]	77.0	78.6	82.0	84.3	80.5
DANN [7]	79.1	80.6	85.3	85.6	82.6
SE [5]	77.0	79.1	83.7	84.9	81.2
KA [27]	80.1	80.9	83.1	86.5	82.5
M^3SDA [24]	79.4	80.8	85.5	86.5	83.1
MDAN [43]	80.0	81.7	84.8	86.8	83.3
MDMN [13]	80.1	81.6	85.6	87.1	83.6
DARN [37]	79.9	81.6	85.8	87.2	83.6
MUST (ours)	**80.5**	**81.9**	**86.3**	**87.9**	**84.2**
Labeled target	84.2	83.8	86.4	88.7	85.8
Error reduction (vs Tgt)	14%	14%	83%	46%	27%

Table 1. Accuracy for sentiment-analysis classification

per each domain, we added a batch-norm layer to the input layer. We used SGD optimizer with a learning rate of 0.001, a momentum of 0.9 for training and experimented with several hyperparameter configurations ($\lambda = 0.25, 0.5, 1.0$ and $C_{th} = 0.6, 0.9$), choosing between them using reverse validation [44].

Compared approaches:

Since our approach uses pseudo labels for the target domain, it may be considered as a kind of knowledge-distillation (KD) technique [25]. Unlike KD, we train the teacher and the student iteratively, so in each iteration, the student is trained on different predictions of the teacher, and the teacher is also influenced by the predictions of the student. This important difference leads the decision boundaries of the teacher to lie in low density regions. To show that difference empirically, we compare MUST with KD. In addition, we compare our results with the state-of-the-art results reported in [27] and [37]: **(1) mSDA**: [4] uses stacked denoising autoencoder to learn new higher-level representations. **(2) DANN**: [7] uses adversarial loss to create a representation that a domain classifier is unable to classify from which domain the feature representation originated. DANN is a single to single DA method and can not be directly applied in a multiple source domains setting. For the multi-source setup [37] merged all the source domains and use them as one large source domain. **(3) Knowledge Adaptation (KA)**: [27] uses multiple teachers, one for each domain and another general teacher that trained on all the sources combined, to train a student to imitate the weighted sum of the teacher's predictions. **(4) Moment matching (M3SDA)**: [24]. **(5) Adversarial MSDA (MDAN)**: [43]. **(6) Multi-Domain Matching Network (MDMN)**: [13]. **(7) Domain aggregation network (DARN)**: [37]. **(8) Target**: a baseline model that trained on the target labels.

Results: The accuracy of the various methods is summarized in Table 1. Clearly, MUST outperforms all other methods.

Method	MNIST	M-M	SVHN	SYN.	Avg.
KD [25]	97.7	74.1	81.1	92.6	86.4
DANN [7]	96.4	60.1	70.2	83.8	77.6
SE [5]	98.6	73.9	78.1	95.1	86.4
M^3SDA [24]	97.0	65.0	71.7	80.1	78.4
MDAN [43]	97.1	64.1	77.7	85.5	81.1
MDMN [13]	97.2	64.3	76.4	85.8	80.9
LtC [34]	98.3	63.1	79.4	92.5	83.3
DARN [37]	98.1	67.1	81.6	86.8	83.4
MUST (ours)	**98.9**	**83.8**	**86.0**	**96.1**	**91.2**
Labeled target	99.0	94.7	87.6	97.0	94.6
Error reduction (vs Tgt)	77%	60%	73%	91%	76%

Table 2. Accuracy for Digit recognition

5.2. Image classification

In the task of image classification, we used the digit recognition datasets and DomainNet dataset [24].

Experimental setup: For image classification, we trained a ResNet-152 with SGD using a learning rate of 0.001 and a momentum of 0.9. We grid-searched over hyperparameter configurations ($\lambda = 0.25, 0.5$ and $C_{th} = 0.25, 0.5, 0.95$) and selected the best configuration based on the average accuracy of the student on the source domain, which is similar to reverse validation.

Digit recognition

The data: The digits dataset is a union of several datasets: (1) MNIST [11]: low resolution black and white images of handwritten digits. (2) MNIST-M (M-M) [6]: consists of MNIST digits blended with random color patches. (3) SVHN [22]: contains low-resolution images of digits from google street view home number. (4) SynthDigits [6]: synthetic SVHN-like dataset digits. Those datasets are combined into one dataset where each dataset is considered as a different domain. It is worth noting that there is also a fifth dataset name USPS, which contains low-resolution MNIST-like images. Since this dataset contains a small number of samples, some protocols discard it. We followed the protocol used by [37] and [43].

Compared approaches: We compare our results with the current state-of-the-art results as reported in [37]. In addition, we compared with LtC [34] using the authors official code.

Results: As summarized in Table 2, MUST outperforms current state-of-the-art MSDA methods by a significant margin. The largest improvement is the adaptation to MNIST-M dataset and SYNTH dataset. This makes sense due to the fact that MUST focus on common features between the sources and the target. MNIST-M is based on MNIST and SYNTH is based on SVHN, so those datasets have more common features than typical datasets, which leads to big improvements.

Office-Caltech

The data: The Office-Caltech [8] dataset is extended from the standard Office31 [29] dataset. It consists of the same 10 object categories from 4 different domains: Amazon, Caltech, DSLR, and Webcam.

Compared approaches: We compare our results with the current state-of-the-art results as reported in [24].

Results: As summarized in Table 3. MUST outperforms current state-of-the-art MSDA methods in 3 out of 4 tasks.

METHOD	WEBCAM	DSLR	CALTECH	AMAZON	AVG.
DAN [15]	99.5	99.1	89.2	91.6	94.8
DCTN [40]	99.4	99.0	90.2	92.7	95.3
JAN [17]	99.4	**99.4**	91.2	91.8	95.5
MCD [30]	99.5	99.1	91.5	92.1	95.6
M^3SDA [24]	99.5	99.2	92.2	94.5	96.4
MUST (OURS)	**99.7**	99.0	**93.6**	**95.3**	**96.9**
(OURS)	$\pm$ 0.05	$\pm$ 0.4	$\pm$ 0.2	$\pm$ 0.6	

Table 3. Classification accuracy on Office-Caltech. Values are averages over 5 seeds; $\pm$ denotes STD over 5 seeds.

DomainNet

We next evaluate MUST in a problem of adaptation for visual object recognition.

The data: DomainNet [24] is a recent challenging dataset designed to evaluate multi-source domain adaptation. It is by far the largest DA dataset, containing six aligned domains (clipart, infograph, painting, quickdraw, real, and sketch) and about 6 million images distributed among 345 categories. This dataset is far more challenging than previous digit-based datasets. As shown by [24], even state-of-the-art methods fail to adapt well across domains in this dataset.

Compared approaches: We compared our results to a set of methods as reported by [24]. In addition, we compared with LtC [34] using the authors official code. For SSDA methods, all samples from the source domains are aggregated as if coming from a single source domain. The model is trained on the aggregated dataset and evaluated on the target domain.

We compared MUST to several SSDA approaches: **(0) No adaptation** Using only source domain samples. **(1) DAN** [15] applied MMD to layers embedded in a reproducing kernel Hilbert space, matching higher order statistics of the two distributions. **(2) RTN** [16] uses residual layers to bridge over components that do not transfer well between domains. **(3) JAN** [17] aligns the joint distributions of multiple layers across domains based on a joint maximum mean discrepancy. **(4) DANN** [7]. **(5) ADDA** [33] combines discriminative modeling and a GAN loss. **(6) SE** [5]. **(7) MCD** [30] finding two classifiers that maximize the discrepancy on the target sample and then generate features that minimize this discrepancy. In addition to SSDA baselines, there are MSDA baselines: **(8) DCTN** [40]. **(9)** $M^3SDA-\beta$ [24]. **(10) DARN**: [37].

	Models	Clip	Info	Paint	Quick	Real	Skt	Avg
Single-Source	Source Only	47.6	13.0	38.1	13.3	51.9	33.7	32.9
	KD [25]	55.6	13.8	36.2	13.7	43.9	44.5	34.6
	DAN [15]	45.4	12.8	36.2	**15.3**	48.6	34.0	32.1
	RTN [16]	44.2	12.6	35.3	14.6	48.4	31.7	31.1
	JAN [17]	49.0	11.1	35.4	12.1	45.8	32.3	29.6
	DANN [7]	45.5	13.1	37.0	13.2	48.9	31.8	32.6
	ADDA [33]	47.5	11.4	36.7	14.7	49.1	33.5	32.2
	SE [5]	24.7	3.9	12.7	7.1	22.8	9.1	16.1
	MCD [30]	54.3	22.1	45.7	7.6	58.4	43.5	38.5
Multi-Source	DCTN [40]	48.6	23.5	48.8	7.2	53.5	47.3	38.2
	M3SDA [24]	58.6	**26.0**	**52.3**	6.3	62.7	49.5	42.5
	DARN [37]	28.5	8.6	29.4	3.2	39.2	20.3	21.5
	LtC [34]	37.4	7.4	19.7	6.7	20.8	30.7	20.5
	MUST (ours)	**60.8**	20.5	48.2	12.2	**65.1**	**49.8**	**42.8**
	Target	71.6	36.7	68.6	69.6	81.8	65.8	65.7

Table 4. Image classification accuracy on DomainNet

Results: As summarized in Table 4. MUST outperforms current state-of-the-art MSDA methods in 4 out of 6 tasks. As was shown by [24], MSDA methods are consistently better than SSDA baselines. The only exception was Quickdraw. No adaptation baselines score is 13.3% while some SSDA baselines improve this score up to 15.3%. Surprisingly the MSDA methods achieve a lower score than no adaptation. This indicates that a negative transfer occurred when applying MSDA methods. MUST reduces the negative transfer and compare to other MSDA methods we almost doubled the accuracy in the Quickdraw task up to 12.2%, getting much closer to the no adaptation baselines. This is still below some SSDA baseline. That demonstrates the difficulty of avoiding negative transfer in the multi-source setting and shows that there is room for further improvement.

5.3. Negative transfer

Learning a joint representation of source and target domains may transfer knowledge from the source that can have a negative impact on the target classifier [41]. In MUST, the teacher is trained on the source domains, and the student is trained on the target domain. We expect this separation to help reduce negative transfer. Since the student network does not train on source samples, it will not learn source features that do not appear in the target. To study this effect, we need a way to measure negative transfer. [36] suggested ways to measure negative transfer, but it focuses on cases where there is at least a small amount of target label data. To overcome this, we created an experiment that helps to estimate the negative transfer on a real-world dataset.

The data: We created a dataset where some features are

Figure 4. Samples from the negative-transfer experiment. Source domain samples were obtained by modifying MNIST: A small patch in the upper left of each image was set to white. The size of the patch is the $n \times n$ pixels where n is the image label. The target domain is the original MNIST dataset.

Method	Accuracy on target
No adaptation	22.3
DANN [7]	11.3
SE [5]	12.4
MUST (ours)	**35.6**

Table 5. Accuracy on the negative-transfer dataset of Sec. 5.3

strongly correlated with the label in the source domain, but those features are absent in the target data. This allows us to quantify the negative transfer by measuring the decline in performance between the source and the target domain.

The data is illustrated in Figure 4. We used MNIST as the target domain. For the source domain, we created a variant of MNIST as follows: We modified a patch in the upper left part of each image, by setting its values to 255. The size of the patch was determined by the label of the image, specifically, it was $n \times n$ pixels where n is the image label.

Compared approaches: We compare MUST to 3 baslines: (1) **No adaptation** using only source domain samples. (2) **DANN** [7]. Since the dataset is SSDA we compared our approach to DANN which is the SSDA version of MDMN [13], MDAN [43] and DARN [37]. (3) **SE** [5].

Results: As shown in Table 5, MUST out-performs other DA methods, and is the only one that improves the performance comparing to the no adaptation baseline.

6. Additional analysis

6.1. Analysis of optimization dynamics

A closer look into the training process can help us understand the dynamic interplay of the two networks. See section C in the supplementary material for a detailed analysis.

6.2. Ablation experiments

We quantify the contribution of various components of MUST. **(1) Only-BN:** We trained a network without the weak coupling using per-domain batch-norm layers. **(2) Teacher:** The teacher model trained using MUST. **(3) MUST:** The full approach described in this paper, use the student as the final classifier. Table 6 summarizes the results for the sentiment analysis experiment, Table 7 for digit recognition and Table 8 for DomainNet.

Method	Books	DVD	Elect	Kitn	Avg.
Only-BN	75.5	78.2	82.3	84.2	80.1
Teacher	76.0	79.1	82.6	84.7	80.6
MUST	**80.5**	**81.9**	**86.3**	**87.9**	**84.2**

Table 6. MUST ablation on Amazon product reviews

Method	MNIST	M-M	SVHN	SYN	Avg.
Only-BN	98.2	72.6	76.3	94.5	85.4
Teacher	**98.9**	83.2	**86.0**	95.9	91.0
MUST	**98.9**	**83.8**	**86.0**	**96.1**	**91.2**

Table 7. MUST ablation on digit recognition

Models	Clip-art	Info-graph	Paint	Quick-draw	Real	Sketch	Avg.
Only-BN	47.9	13.5	39.9	9.3	55.9	36.6	33.8
Teacher	58.9	8.1	47.9	3.3	**65.2**	49.3	38.8
MUST	**60.8**	**20.5**	**48.2**	**12.2**	65.1	**49.8**	**42.8**

Table 8. MUST ablation on DomainNet

In most cases, the teacher improves the Only-BN score, demonstrating the influence of the student feedback on the teacher. In addition, the student is consistently better than the teacher, indicating the reduction of negative transfer. This effect is particularly noticeable on Quickdraw where the negative transfer for all MSDA methods was shown in section 5.2.

7. Conclusion

This paper describes a new MSDA method. Separate networks are trained while weakly coupling their predictions. We showed that this weak coupling moves the decision boundary to low-density regions in the target distribution. An experiment designed to measure the effect of negative transfer shows that its effect is reduced using our approach. We hope this experiment will encourage the DA community to measure negative transfer of new methods. Evaluated on three MSDA benchmarks, we find that MUST achieves new SoTA results.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[2] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128, 2006.

[3] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017.

[4] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

[5] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017.

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

[7] 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.

[8] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[9] Konstantinos Kamnitsas, Daniel Castro, Loic Le Folgoc, Ian Walker, Ryutaro Tanno, Daniel Rueckert, Ben Glocker, Antonio Criminisi, and Aditya Nori. Semi-supervised learning via compact latent space clustering. In *International Conference on Machine Learning*, pages 2459–2468. PMLR, 2018.

[10] Wouter Marco Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 382–403. Springer, 2020.

[13] Yitong Li, David E Carlson, et al. Extracting relationships by multi-domain matching. In *Advances in Neural Information Processing Systems*, pages 6798–6809, 2018.

[14] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

[15] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[16] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in neural information processing systems*, pages 136–144, 2016.

[17] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2208–2217. JMLR. org, 2017.

[18] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

[19] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in neural information processing systems*, pages 1041–1048, 2009.

[20] Zhong Meng, Jinyu Li, Yashesh Gaur, and Yifan Gong. Domain adaptation via teacher-student learning for end-to-end speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 268–275. IEEE, 2019.

[21] Zhong Meng, Jinyu Li, Yifan Gong, and Biing-Hwang Juang. Adversarial teacher-student learning for unsupervised domain adaptation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5949–5953. IEEE, 2018.

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

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

[24] 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 International Conference on Computer Vision*, pages 1406–1415, 2019.

[25] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151. PMLR, 2019.

[26] Sayan Rakshit, Biplab Banerjee, Gemma Roig, and Subhasis Chaudhuri. Unsupervised multi-source domain adaptation driven by deep adversarial ensemble learning. In *German Conference on Pattern Recognition*, pages 485–498. Springer, 2019.

[27] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. Knowledge adaptation: Teaching to adapt. *arXiv preprint arXiv:1702.02052*, 2017.

[28] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[29] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[30] 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.

[31] Philip Sellars, Angelica Aviles-Rivero, and Carola Bibiane Schönlieb. Two cycle learning: clustering based regularisation for deep semi-supervised classification. *arXiv preprint arXiv:2001.05317*, 2020.

[32] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceed-*

ings of the IEEE conference on computer vision and pattern recognition, pages 7167–7176, 2017.

[33] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[34] 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.

[35] Yunyun Wang, Songcan Chen, and Zhi-Hua Zhou. New semi-supervised classification method based on modified cluster assumption. *IEEE Transactions on Neural Networks and Learning Systems*, 23(5):689–702, 2012.

[36] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019.

[37] 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.

[38] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *arXiv preprint arXiv:1812.02849*, 2018.

[39] Dustin Wright and Isabelle Augenstein. Transformer based multi-source domain adaptation. *arXiv preprint arXiv:2009.07806*, 2020.

[40] 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.

[41] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1426–1435, 2019.

[42] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. *arXiv preprint arXiv:2007.01261*, 2020.

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

[44] Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 547–562. Springer, 2010.