Robust Domain Adaptation: Representations, Weights and Inductive Bias*

Victor Bouvier[1,2](✉), Philippe Very**[3], Clément Chastagnol*[4], Myriam Tami[1], and Céline Hudelot[1]

[1] Universit Paris-Saclay, CentraleSuplec, Mathmatiques et Informatique pour la Complexit et les Systmes, 91190, Gif-sur-Yvette, France
firstname.name@centralesupelec.fr
[2] Sidetrade, 114 Rue Gallieni, 92100, Boulogne-Billancourt, France,
vbouvier@sidetrade.com
[3] Lend-Rx, 24 Rue Saint Dominique, 75007, Paris, France,
philippe.very@lend-rxtech.com
[4] Alan, 117 Quai de Valmy, 75010 Paris, France,
clement.chastagnol@alan.eu

Abstract. Unsupervised Domain Adaptation (UDA) has attracted a lot of attention in the last ten years. The emergence of Domain Invariant Representations (IR) has improved drastically the transferability of representations from a labelled source domain to a new and unlabelled target domain. However, a potential pitfall of this approach, namely the presence of *label shift*, has been brought to light. Some works address this issue with a relaxed version of domain invariance obtained by weighting samples, a strategy often referred to as Importance Sampling. From our point of view, the theoretical aspects of how Importance Sampling and Invariant Representations interact in UDA have not been studied in depth. In the present work, we present a bound of the target risk which incorporates both weights and invariant representations. Our theoretical analysis highlights the role of inductive bias in aligning distributions across domains. We illustrate it on standard benchmarks by proposing a new learning procedure for UDA. We observed empirically that weak inductive bias makes adaptation more robust. The elaboration of stronger inductive bias is a promising direction for new UDA algorithms.

Keywords: Unsupervised Domain Adaptation · Importance Sampling · Invariant Representations · Inductive Bias

1 Introduction

Deploying machine learning models in the real world often requires the ability to generalize to *unseen samples i.e.* samples significantly different from those

* Published at ECML-PKDD2020, Best (Student) Paper Award (https://ecmlpkdd2020.net/programme/awards/).
** Work done when author was at Sidetrade

seen during learning. Despite impressive performances on a variety of tasks, deep learning models do not always meet these requirements [3,14]. For this reason, *out-of-distribution generalization* is recognized as a major challenge for the reliability of machine learning systems [1,2]. Domain Adaptation (DA) [30,28] is a well-studied approach to bridge the gap between train and test distributions. In DA, we refer to train and test distributions as *source* and *target* respectively noted $p_S(x, y)$ and $p_T(x, y)$ where x are inputs and y are labels. The objective of DA can be defined as learning a good classifier on a poorly sampled target domain by leveraging samples from a source domain. Unsupervised Domain Adaptation (UDA) assumes that only unlabelled data from the target domain is available during training. In this context, a natural assumption, named *Covariate shift* [33,19], consists in assuming that the mapping from the inputs to the labels is conserved across domains, *i.e.* $p_T(y|x) = p_S(y|x)$. In this context, *Importance Sampling* (IS) performs adaptation by weighting the contribution of sample x in the loss by $w(x) = p_T(x)/p_S(x)$ [30]. Although IS seems natural when unlabelled data from the target domain is available, the covariate shift assumption is not sufficient to guarantee successful adaptation [5]. Moreover, for high dimensional data [12] such as texts or images, the shift between $p_S(x)$ and $p_T(x)$ results from non-overlapping supports leading to unbounded weights [20].

In this particular context, representations can help to reconcile non-overlapping supports [5]. This seminal idea, and the corresponding theoretical bound of the target risk from [5], has led to a wide variety of deep learning approaches [13,23,24] which aim to learn a so-called *domain invariant representation*:

$$p_S(z) \approx p_T(z) \tag{1}$$

where $z := \varphi(x)$ for a given non-linear representation φ. These assume that the *transferability* of representations, defined as the combined error of an ideal classifier, remains low during learning. Unfortunately, this quantity involves target labels and is thus intractable. More importantly, looking for strict invariant representations, $p_S(z) = p_T(z)$, hurts the transferability of representations [20,22,36,40]. In particular, there is a fundamental trade-off between learning invariant representations and preserving transferability in presence of label shift ($p_T(y) \neq p_S(y)$) [40]. To mitigate this trade-off, some recent works suggest to relax domain invariance by weighting samples [8,36,37,9]. This strategy differs with (1) by aligning a *weighted source* distribution with the target distribution:

$$w(z)p_S(z) \approx p_T(z) \tag{2}$$

for some weights $w(z)$. We now have two tools, w and φ, which need to be calibrated to obtain distribution alignment. Which one should be promoted? How weights preserve good transferability of representations?

While most prior works focus on the invariance error for achieving adaptation [13,23,24], this paper focuses on the transferability of representations. We show that weights allow to design an interpretable generalization bound where transferability and invariance errors are uncoupled. In addition, we discuss the role of inductive design for both the classifier and the weights in addressing the lack of labelled data in the target domain. Our contributions are the following:

1. We introduce a new bound of the target risk which incorporates both weights and domain invariant representations. Two new terms are introduced. The first is an *invariance error* which promotes alignment between a weighted source distribution of representations and the target distribution of representations. The second, named *transferability error*, involves labelling functions from both source and target domains.
2. We highlight the role of **inductive bias** for approximating the transferability error. First, we establish connections between our bound and popular approaches for UDA which use target predicted labels during adaptation, in particular Conditional Domain Adaptation [24] and Minimal Entropy [15]. Second, we show that the inductive design of weights has an impact on representation invariance.
3. We derive a new learning procedure for UDA. The particularity of this procedure is to only minimize the transferability error while controlling representation invariance with weights. Since the transferability error involves target labels, we use the predicted labels during learning.
4. We provide an empirical illustration of our framework on two DA benchmarks (**Digits** and **Office31** datasets). We stress-test our learning scheme by modifying strongly the label distribution in the source domain. While methods based on invariant representations deteriorate considerably in this context, our procedure remains robust.

2 Preliminaries

We introduce the *source* distribution *i.e.* data where the model is trained with supervision and the *target* distribution *i.e.* data where the model is tested or applied. Formally, for two random variables (X, Y) on a given space $\mathcal{X} \times \mathcal{Y}$, we introduce two distributions: the source distribution $p_S(x, y)$ and the target distribution $p_T(x, y)$. Here, labels are one-hot encoded *i.e.* $y \in [0, 1]^C$ such that $\sum_c y_c = 1$ where C is the number of classes. The distributional shift situation is then characterized by $p_S(x, y) \neq p_T(x, y)$ [30]. In the rest of the paper, we use the index notation S and T to differentiate source and target terms. We define the hypothesis class $\mathcal{H}$ as a subset of functions from $\mathcal{X}$ to $\mathcal{Y}$ which is the composition of a representation class Φ and a classifier class $\mathcal{G}$, *i.e.* $\mathcal{H} = \mathcal{G} \circ \Phi$. For the ease of reading, given a classifier $g \in \mathcal{G}$ and a representation $\varphi \in \Phi$, we note $g\varphi := g \circ \varphi$. Furthermore, in the definition $z := \varphi(x)$, we refer indifferently to z, φ, $Z := \varphi(X)$ as the *representation*. For two given h and $h' \in \mathcal{H}$ and ℓ the L^2 loss $\ell(y, y') = ||y - y'||^2$, the risk in domain $D \in \{S, T\}$ is noted:

$$\varepsilon_D(h) := \mathbb{E}_D[\ell(h(X), Y)] \tag{3}$$

and $\varepsilon_D(h, h') := \mathbb{E}_D[\ell(h(X), h'(X))]$. In the seminal works [5,27], a theoretical limit of the target risk when using a representation φ has been derived:

Bound 1 (Ben David et al.) *Let* $d_{\mathcal{G}}(\varphi) = \sup_{g,g' \in \mathcal{G}} |\varepsilon_S(g\varphi, g'\varphi) - \varepsilon_T(g\varphi, g'\varphi)|$ *and* $\lambda_{\mathcal{G}}(\varphi) = \inf_{g \in \mathcal{G}} \{\varepsilon_S(g\varphi) + \varepsilon_T(g\varphi)\}$, $\forall g \in \mathcal{G}, \forall \varphi \in \Phi$*:*

$$\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi) + d_{\mathcal{G}}(\varphi) + \lambda_{\mathcal{G}}(\varphi) \tag{4}$$

This generalization bound ensures that the target risk $\varepsilon_T(g\varphi)$ is bounded by the sum of the source risk $\varepsilon_S(g\varphi)$, the disagreement risk between two classifiers from representations $d_{\mathcal{G}}(\varphi)$, and a third term, $\lambda_{\mathcal{G}}(\varphi)$, which quantifies the ability to perform well in both domains from representations. The latter is referred to as the *adaptability* error of representations. It is intractable in practice since it involves labels from the target distribution. Promoting distribution invariance of representations, *i.e.* $p_S(z)$ close to $p_T(z)$, results on a low $d_{\mathcal{G}}(\varphi)$. More precisely:

$$d_{\mathcal{G}}(\varphi) \leq 2 \sup_{d \in \mathcal{D}} |p_S(d(z) = 1) - p_T(d(z) = 0)| \tag{5}$$

where $\mathcal{D}$ is the so-called set of *discriminators* or *critics* which verifies $\mathcal{D} \supset \{g \oplus g' : (g, g') \in \mathcal{G}^2\}$ where $\oplus$ is the XOR function [13]. Since the domain invariance term $d_{\mathcal{G}}(\varphi)$ is expressed as a supremal value on classifiers, it is suitable for domain adversarial learning with critic functions. Conversely, the adaptability error $\lambda_{\mathcal{G}}(\varphi)$ is expressed as an infremal value. This 'sup / inf' duality induces an unexpected trade-off when learning domain invariant representations:

Proposition 1 (Invariance hurts adaptability [20,40]). *Let ψ be a representation which is a richer feature extractor than φ: $\mathcal{G} \circ \varphi \subset \mathcal{G} \circ \psi$. Then,*

$$d_{\mathcal{G}}(\varphi) \leq d_{\mathcal{G}}(\psi) \text{ while } \lambda_{\mathcal{G}}(\psi) \leq \lambda_{\mathcal{G}}(\varphi) \tag{6}$$

As a result of proposition 1, the benefit of representation invariance must be higher than the loss of adaptability, which is impossible to guarantee in practice.

3 Theory

To overcome the limitation raised in proposition 1, we expose a new bound of the target risk which embeds a new trade-off between invariance and transferability (3.1). We show this new bound remains inconsistent with the presence of label shift (3.2) and we expose the role of weights to address this problem (3.3).

3.1 A new trade-off between Invariance and Transferability

Core assumptions. Our strategy is to express both the transferability and invariance as a supremum using Integral Probability Measure (IPM) computed on a critic class. We thus introduce a class of critics suitable for our analysis. Let $\mathcal{F}$ from $\mathcal{Z} \to [-1, 1]$ and $\mathcal{F}_C$ from $\mathcal{Z} \to [-1, 1]^C$ with the following properties:

- (A1) $\mathcal{F}$ and $\mathcal{F}_C$ are symmetric (*i.e.* $\forall f \in \mathcal{F}, -f \in \mathcal{F}$) and convex.
- (A2) $\mathcal{G} \subset \mathcal{F}_C$ and $\{\mathbf{f} \cdot \mathbf{f}' \ ; \ \mathbf{f}, \mathbf{f}' \in \mathcal{F}_C\} \subset \mathcal{F}$.
- (A3) $\forall \varphi \in \Phi$, $\mathbf{f}_D(z) \mapsto \mathbb{E}_D[Y|\varphi(X) = z] \in \mathcal{F}_C$. [5]
- (A4) For two distributions p and q on $\mathcal{Z}$, $p = q$ if and only if:

$$\text{IPM}(p, q; \mathcal{F}) := \sup_{f \in \mathcal{F}} \{\mathbb{E}_p[f(Z)] - \mathbb{E}_q[f(Z)]\} = 0 \tag{7}$$

[5] See Appendix A.1 for more details on this assumption.

The assumption (A1) ensures that rather comparing two given $\mathbf{f}$ and $\mathbf{f}'$, it is enough to study the error of some $\mathbf{f}'' = \frac{1}{2}(\mathbf{f} - \mathbf{f}')$ from $\mathcal{F}_C$. This brings back a supremum on $\mathcal{F}_C^2$ to a supremum on $\mathcal{F}_C$. The assumption (A2), combined with (A1), ensures that an error $\ell(\mathbf{f}, \mathbf{f}')$ can be expressed as a critic function $f \in \mathcal{F}$ such that $f = \ell(\mathbf{f}, \mathbf{f}')$. The assumption (A3) ensures that $\mathcal{F}_C$ is rich enough to contain label function from representations. Here, $\mathbf{f}_D(z) = \mathbb{E}_D[Y|Z = z]$ is a vector of probabilities on classes: $f_D(z)_c = p_D(Y = c|Z = z)$. The last assumption (A4) ensures that the introduced IPM is a distance. Classical tools verify these assumptions *e.g.* continuous functions; here $\text{IPM}(p, q; \mathcal{F})$ is the *Maximum Mean Discrepancy* [16] and one can reasonably believe that $\mathbf{f}_S$ and $\mathbf{f}_T$ are continuous.

Invariance and transferability as IPMs. We introduce here two important tools that will guide our analysis:

- $\text{INV}(\varphi)$, named *invariance error*, that aims at capturing the difference between source and target distribution of representations, corresponding to:

$$\text{INV}(\varphi) := \sup_{f \in \mathcal{F}} \{\mathbb{E}_T[f(Z)] - \mathbb{E}_S[f(Z)]\} \tag{8}$$

- $\text{TSF}(\varphi)$, named *transferability error*, that catches if the coupling between Z and Y shifts across domains. For that, we use our class of functions $\mathcal{F}_C$ and we compute the IPM of $Y \cdot \mathbf{f}(Z)$, where $\mathbf{f} \in \mathcal{F}_C$ and $Y \cdot \mathbf{f}(Z)$ is the scalar product[6], between the source and the target domains:

$$\text{TSF}(\varphi) := \sup_{\mathbf{f} \in \mathcal{F}_C} \{\mathbb{E}_T[Y \cdot \mathbf{f}(Z)] - \mathbb{E}_S[Y \cdot \mathbf{f}(Z)]\} \tag{9}$$

A new bound of the target risk. Using $\text{INV}(\varphi)$ and $\text{TSF}(\varphi)$, we can provide a new bound of the target risk:

Bound 2 *$\forall g \in \mathcal{G}$ and $\forall \varphi \in \Phi$:*

$$\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi) + 6 \cdot \text{INV}(\varphi) + 2 \cdot \text{TSF}(\varphi) + \varepsilon_T(\mathbf{f}_T\varphi) \tag{10}$$

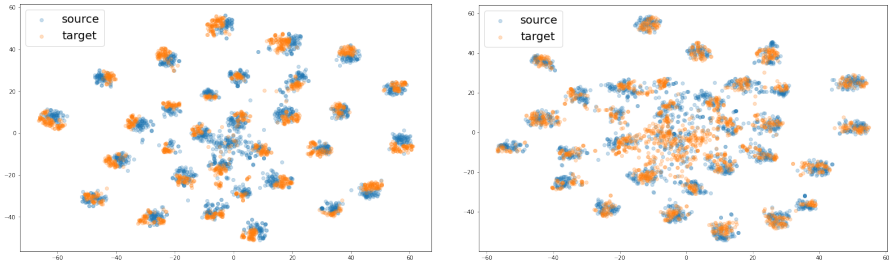
The proof is in Appendix A.1. In contrast with bound 1 (Eq. 6), here two IPMs are involved to compare representations ($\text{INV}(\varphi)$ and $\text{TSF}(\varphi)$). A new term, $\varepsilon_T(\mathbf{f}_T\varphi)$, reflects the level of noise when fitting labels from representations. All the trade-off between invariance and transferability is embodied in this term:

Proposition 2. *Let ψ a representation which is a richer feature extractor than φ: $\mathcal{F} \circ \varphi \subset \mathcal{F} \circ \psi$ and $\mathcal{F}_C \circ \varphi \subset \mathcal{F}_C \circ \psi$. φ is more domain invariant than ψ:*

$$\text{INV}(\varphi) \leq \text{INV}(\psi) \text{ while } \varepsilon_T(\mathbf{f}_T^{\psi}\psi) \leq \varepsilon_T(\mathbf{f}_T^{\varphi}\varphi) \tag{11}$$

where $\mathbf{f}_T^{\varphi}(z) = \mathbb{E}_T[Y|\varphi(X) = z]$ and $\mathbf{f}_T^{\psi}(z) = \mathbb{E}_T[Y|\psi(X) = z]$. Proof in A.2.

[6] the scalar product between Y and $\mathbf{f}(Z)$ emerges from the choice of the L^2 loss.



(a) $\lambda_{\mathcal{G}}(\varphi)$ adaptability in bound 1 from [5]. Inside class clusters, source and target representations are separated.

(b) TSF(φ) transferability from bound 2 (contribution). Inside class clusters, source and target representations are not distinguishable

Fig. 1. t-SNE [26] visualisation of representations when trained to minimize (a) adaptability error $\lambda_{\mathcal{G}}(\varphi)$ from [5], (b) transferability error TSF(φ) introduced in the present work. The task used is A→W of the **Office31** dataset. *Labels in the target domain are used during learning in this specific experiment.* For both visualisations of representations, we observe well-separated clusters associated to the label classification task. Inside those clusters, we observe a separation between source and target representations for $\lambda_{\mathcal{G}}(\varphi)$. That means that representations embed domain information and thus are not invariant. On the contrary, source and target representations are much more overlapping inside of each cluster with TSF(φ), illustrating that this new term is not conflictual with invariance.

Bounding the target risk using IPMs has two advantages. First, it allows to better control the invariance / transferability trade-off since $\varepsilon_T(\mathbf{f}_T\varphi) \leq \lambda_{\mathcal{G}}(\varphi)$. This is paid at the cost of $4{\cdot}\text{INV}(\varphi) \geq d_{\mathcal{G}}(\varphi)$ (see Proposition 7 in Appendix A.1). Second, $\varepsilon_T(\mathbf{f}_T\varphi)$ is source free and indicates whether there is enough information in representations for learning the task in the target domain at first. This means that TSF(φ) is only dedicated to control if aligned representations have the same labels across domains. To illustrate the interest of our new transferability error, we provide visualisation of representations (Fig. 1) when trained to minimize the adaptability error $\lambda_{\mathcal{G}}(\varphi)$ from bound 1 and the transferability error TSF(φ) from bound 2.

3.2 A detailed view on the property of tightness

An interesting property of the bound, named tightness, is the case when INV$(\varphi) =$ 0 and TSF$(\varphi) = 0$ simultaneously. The condition of tightness of the bound provides rich information on the properties of representations.

Proposition 3. INV$(\varphi) =$ TSF$(\varphi) = 0$ *if and only if* $p_S(y, z) = p_T(y, z)$.

The proof is given in Appendix A.3. Two important points should be noted:

1. INV$(\varphi) = 0$ ensures that $p_S(z) = p_T(z)$, using (A4). Similarly, TSF$(\varphi) = 0$ leads to $p_S(y, z) = p_T(y, z)$. Since $p_S(y, z) = p_T(y, z)$ implies $p_S(z) = p_T(z)$, INV(φ) does not bring more substantial information about representations

distribution than $\text{TSF}(\varphi)$. More precisely, one can show that $\text{TSF}(\varphi) \geq \text{INV}(\varphi)$ noting that $Y \cdot \mathbf{f}(Z) = f(z)$ when $\mathbf{f}(z) = (f(z), ..., f(z))$ for $f \in \mathcal{F}$.
2. Second, the equality $p_S(y, z) = p_T(y, z)$ also implies that $p_S(y) = p_T(y)$. Therefore, in the context of label shift (when $p_S(y) \neq p_T(y)$), the transferability error cannot be null. This is a big hurdle since it is clearly established that most real world UDA tasks exhibit some label shift. This bound highlights the fact that representation invariance alone can not address UDA in complex settings such as the label shift one.

3.3 Reconciling Weights and Invariant Representations.

Based on the interesting observations from [20,40] and following the line of study that proposed to relax invariance using weights [9,38,37,36], we propose to adapt the bound by incorporating weights. More precisely, we study the effect of modifying the source distribution $p_S(z)$ to a *weighted source* distribution $w(z)p_S(z)$ where w is a positive function which verifies $\mathbb{E}_S[w(Z)] = 1$. By replacing $p_S(z)$ by $w(z)p_S(z)$ (distribution referred as $w \cdot S$) in bound 2, we obtain a new bound of the target risk incorporating both weights and representations:

Bound 3 $\forall g \in \mathcal{G}, \forall w : \mathcal{Z} \to \mathbb{R}^+$ *such that* $\mathbb{E}_S[w(z)] = 1$*:*

$$\varepsilon_T(g\varphi) \leq \varepsilon_{w \cdot S}(g\varphi) + 6 \cdot \text{INV}(w, \varphi) + 2 \cdot \text{TSF}(w, \varphi) + \varepsilon_T(\mathbf{f}_T\varphi)$$

where $\text{INV}(w, \varphi) := \sup_{f \in \mathcal{F}} \{\mathbb{E}_T[f(Z)] - \mathbb{E}_S[w(Z)f(Z)]\}$ *and* $\text{TSF}(w, \varphi) := \sup_{\mathbf{f} \in \mathcal{F}_C} \{\mathbb{E}_T[Y \cdot \mathbf{f}(Z)] - \mathbb{E}_S[w(Z)Y \cdot \mathbf{f}(Z)]\}$.

As for the previous bound 2, the property of tightness, *i.e.* when invariance and transferability are null simultaneously, leads to interesting observations:

Proposition 4. $\text{INV}(w, \varphi) = \text{TSF}(w, \varphi) = 0$ *if and only if* $w(z) = \frac{p_T(z)}{p_S(z)}$ *and* $\mathbb{E}_T[Y|Z = z] = \mathbb{E}_S[Y|Z = z]$*. The proof is given in Appendix A.4.*

This proposition means that the nullity of invariance error, *i.e.* $\text{INV}(w, \varphi) = 0$, implies distribution alignment, *i.e.* $w(z)p_S(z) = p_T(z)$. This is of strong interest since both representations and weights are involved for achieving domain invariance. The nullity of the transferability error, *i.e.* $\text{TSF}(w, \varphi) = 0$, implies that labelling functions, $\mathbf{f} : z \mapsto \mathbb{E}[Y|Z = z]$, are conserved across domains. Furthermore, the equality $\mathbb{E}_T[Y|Z] = \mathbb{E}_S[Y|Z]$ interestingly resonates with a recent line of work called *Invariant Risk Minimization* (IRM) [2]. Incorporating weights in the bound thus brings two benefits:

1. First, it raises the inconsistency issue of invariant representations in presence of label shift, as mentioned in section 3. Indeed, tightness is not conflicting with label shift.
2. $\text{TSF}(w, \varphi)$ and $\text{INV}(w, \varphi)$ have two disctinct roles: the former promotes domain invariance of representations while the latter controls whether aligned representations share the same labels across domains.

4 The role of Inductive Bias

Inductive Bias refers to the set of assumptions which improves generalization of a model trained on an empirical distribution. For instance, a specific neural network architecture or a well-suited regularization are prototypes of inductive biases. First, we provide a theoretical analysis of the role of inductive bias for addressing the lack of labelling data in the target domain (4.1), which is the most challenging part of *Unsupervised* Domain Adaptation. Second, we describe the effect of weights to induce invariance property on representations (4.2).

4.1 Inductive design of a classifier

General Formulation. Our strategy consists in approximating target labels error through a classifier $\tilde{g} \in \mathcal{G}$. We refer to the latter as the inductive design of the classifier. Our proposition follows the intuitive idea which states that the best source classifier, $g_S := \arg\min_{g \in \mathcal{G}} \varepsilon_S(g\varphi)$, is not necessarily the best target classifier *i.e.* $g_S \neq \arg\min_{g \in \mathcal{G}} \varepsilon_T(g\varphi)$. For instance, a well-suited regularization in the target domain, noted $\Omega_T(g)$ may improve performance, *i.e.* setting $\tilde{g} := \arg\min_{g \in \mathcal{G}} \varepsilon_S(g\varphi) + \lambda \cdot \Omega_T(g)$ may lead to $\varepsilon_T(\tilde{g}\varphi) \leq \varepsilon_T(g_S\varphi)$. We formalize this idea through the following definition:

Definition 5 (Inductive design of a classifier). *We say that there is an inductive design of a classifier at level $0 < \beta \leq 1$ if for any representations φ, noting $g_S = \arg\min_{g \in \mathcal{G}} \varepsilon_S(g\varphi)$, we can determine $\tilde{g}$ such that:*

$$\varepsilon_T(\tilde{g}\varphi) \leq \beta \varepsilon_T(g_S\varphi) \tag{12}$$

We say the inductive design is $\beta-$strong when $\beta < 1$ and weak when $\beta = 1$.

In this definition, β does not depend of φ, which is a strong assumption, and embodies the strength of the inductive design. The closer to 1 is β, the less improvement we can expect using the inductive classifier $\tilde{g}$. We now study the impact of the inductive design of a classifier in our previous bound 3. Thus, we introduce the approximated transferability error:

$$\widehat{\mathrm{TSF}}(w, \varphi, \tilde{g}) = \sup_{\mathbf{f} \in \mathcal{F}_C} \{\mathbb{E}_T[\tilde{g}(Z) \cdot \mathbf{f}(Z)] - \mathbb{E}_S[w(Z)Y \cdot \mathbf{f}(Z)]\} \tag{13}$$

leading to a bound of the target risk where transferability is target labels free:

Bound 4 (Inductive Bias and Guarantee) *Let $\varphi \in \Phi$ and $w : \mathcal{Z} \to \mathbb{R}^+$ such that $\mathbb{E}_S[w(z)] = 1$ and a $\beta-$strong inductive classifier $\tilde{g}$ and $\rho := \frac{\beta}{1-\beta}$ then:*

$$\varepsilon_T(\tilde{g}\varphi) \leq \rho \left(\varepsilon_{w \cdot S}(g_{w \cdot S}\varphi) + 6 \cdot \mathrm{INV}(w, \varphi) + 2 \cdot \widehat{\mathrm{TSF}}(w, \varphi, \tilde{g}) + \varepsilon_T(\mathbf{f}_T\varphi) \right) \tag{14}$$

The proof is given in Appendix A.5. Here, the target labels are only involved in $\varepsilon_T(\mathbf{f}_T\varphi)$ which reflects the level of noise when fitting labels from representations. Therefore, transferability is now free of target labels. This is an important result since the difficulty of UDA lies in the lack of labelled data in the target domain. It is also interesting to note that the weaker the inductive bias ($\beta \to 1$), the higher the bound and vice versa.

The role of predicted labels. Predicted labels play an important role in UDA. In light of the inductive classifier, this means that $\tilde{g}$ is simply set as $g_{w\cdot S}$. This is a weak inductive design ($\beta = 1$), thus, theoretical guarantee from bound 4 is not applicable. However, there is empirical evidence that showed that predicted labels help in UDA [15,24]. It suggests that this inductive design may find some strength in the finite sample regime. A better understanding of this phenomenon is left for future work (See Appendix B). In the rest of the paper, we study this weak inductive bias by establishing connections between $\widehat{\mathrm{TSF}}(w, \varphi, g_S)$ and popular approaches of the literature.

Connections with Conditional Domain Adaptation Network. CDAN [24] aims to align the joint distribution $(\hat{Y}, Z)$ across domains, where $\hat{Y} = g_S\varphi(X)$ are estimated labels. It is performed by exposing the tensor product between $\hat{Y}$ and Z to a discriminator. It leads to substantial empirical improvements compared to *Domain Adversarial Neural Networks* (DANN) [13]. We can observe that it is a similar objective to $\widehat{\mathrm{TSF}}(w, \varphi, g_S)$ in the particular case where $w(z) = 1$.

Connections with Minimal Entropy. MinEnt [15] states that an adapted classifier is confident in prediction on target samples. It suggests the regularization: $\Omega_T(g) := H(\hat{Y}|Z) = \mathbb{E}_{Z\sim p_T}[-g(Z)\cdot \log g(Z)]$ where H is the entropy. If labels are smooth enough (*i.e.* it exists α such that $\frac{\alpha}{C-1} \leq \mathbb{E}_S[\hat{Y}|Z] \leq 1-\alpha$), MinEnt is a lower bound of transferability: $\widehat{\mathrm{TSF}}(w, \varphi, g_s) \geq \eta\left(H_T(g_S\varphi) - \mathrm{CE}_{w\cdot S}(Y, g_S\varphi)\right)$ for some $\eta > 0$ and $\mathrm{CE}_{w\cdot S}(g_S\varphi, Y)$ is the cross-entropy between $g_S\varphi$ and Y on $w(z)p_S(z)$ (see Appendix A.6).

4.2 Inductive design of weights

While the bounds introduced in the present work involve weights in the representation space, there is an abundant literature that builds weights in order to relax the domain invariance of representations [8,36,37,9]. We study the effect of inductive design of w on representations. To conduct the analysis, we consider there is a non-linear transformation ψ from $\mathcal{Z}$ to $\mathcal{Z}'$ and we assume that weights are computed in $\mathcal{Z}'$, *i.e.* w is a function of $z' := \psi(z) \in \mathcal{Z}'$. We refer to this as *inductive design of weights*. For instance, in the particular case where $\psi = g_S$, weights are designed as $w(\hat{y}) = p_T(\hat{Y} = \hat{y})/p_S(\hat{Y} = \hat{y})$ [9] where $\hat{Y} = g\varphi(X)$. In [24], *entropy conditioning* is introduced by designing weights $w(z') \propto 1 + e^{-z'}$ where $z' = -\frac{1}{C}\sum_{1\leq c\leq C} g_{S,c}\log(g_{S,c})$ is the predictions entropy. The inductive design of weights imposes invariance property on representations:

Proposition 6 (Inductive design of w and invariance). *Let $\psi : \mathcal{Z} \to \mathcal{Z}'$ such that $\mathcal{F}\circ\psi \subset \mathcal{F}$ and $\mathcal{F}_C\circ\psi \subset \mathcal{F}_C$. Let $w : \mathcal{Z}' \to \mathbb{R}^+$ such that $\mathbb{E}_S[w(Z')] = 1$ and we note $Z' := \psi(Z)$. Then, $\mathrm{INV}(w, \varphi) = \mathrm{TSF}(w, \varphi) = 0$ if and only if:*

$$w(z') = \frac{p_T(z')}{p_S(z')} \quad \textit{and} \quad p_S(z|z') = p_T(z|z') \tag{15}$$

while both $\mathbf{f}_S^{\varphi} = \mathbf{f}_T^{\varphi}$ and $\mathbf{f}_S^{\psi} = \mathbf{f}_T^{\psi}$. The proof is given in Appendix A.7.

This proposition shows that the design of w has a significant impact on the property of domain invariance of representations. Furthermore, both labelling functions are conserved. In the rest of the paper we focus on weighting in the representation space which consists in:

$$w(z) = \frac{p_T(z)}{p_S(z)} \tag{16}$$

Since it does not leverage any transformations of representations ψ, we refer to this approach as a weak inductive design of weights. It is worth noting this inductive design controls naturally the invariance error *i.e.* $\mathrm{INV}(w, \varphi) = 0$.

5 Towards Robust Domain Adaptation

In this section, we expose a new learning procedure which relies on weak inductive design of both weights and the classifier. This procedure focuses on the transferability error since the inductive design of weights naturally controls the invariance error. Our learning procedure is then a bi-level optimization problem, named RUDA (Robust UDA):

$$\begin{cases} \varphi^\star = \arg\min\limits_{\varphi \in \Phi} \ \varepsilon_{w(\varphi)\cdot S}(g_{w\cdot S}\varphi) + \lambda \cdot \widehat{\mathrm{TSF}}(w, \varphi, g_{w\cdot S}) \\ \quad \text{such that } w(\varphi) = \arg\min\limits_{w} \mathrm{INV}(w, \varphi) \end{cases} \tag{RUDA}$$

where $\lambda > 0$ is a trade-off parameter. Two discriminators are involved here. The former is a domain discriminator d trained to map 1 for source representations and 0 for target representations by minimizing a domain adversarial loss:

$$\mathcal{L}_{\mathrm{INV}}(\theta_d | \theta_\varphi) = \frac{1}{n_S} \sum_{i=1}^{n_S} -\log(d(z_{S,i})) + \frac{1}{n_T} \sum_{i=1}^{n_T} -\log(1 - d(z_{T,i})) \tag{17}$$

where θ_d and θ_φ are respectively the parameters of d and φ, and n_S and n_T are respectively the number of samples in the source and target domains. Setting weights $w_d(z) := (1 - d(z))/d(z)$ ensures that $\mathrm{INV}(w, \varphi)$ is minimal (See Appendix C.2). The latter, noted $\mathbf{d}$, maps representations to the label space $[0, 1]^C$ in order to obtain a proxy of the transferability error expressed as a domain adversarial objective (See Appendix C.1):

$$\begin{aligned} \mathcal{L}_{\mathrm{TSF}}(\theta_\varphi, \theta_{\mathbf{d}} | \theta_d, \theta_g) = \inf_{\mathbf{d}} \Bigg\{ & \frac{1}{n_S} \sum_{i=1}^{n_S} -w_d(z_{S,i}) g(z_{S,i}) \cdot \log(\mathbf{d}(z_{S,i})) \\ & + \frac{1}{n_T} \sum_{i=1}^{n_T} -g(z_{T,i}) \cdot \log(1 - \mathbf{d}(z_{T,i})) \Bigg\} \end{aligned} \tag{18}$$

where $\theta_{\mathbf{d}}$ and θ_g are respectively parameters of $\mathbf{d}$ and g. Furthermore, we use the cross-entropy loss in the source weighted domain for learning θ_g:

$$\mathcal{L}_c(\theta_g, \theta_\varphi | \theta_d) = \frac{1}{n_S} \sum_{i=1}^{n_S} -w_d(z_{S,i}) y_{S,i} \cdot \log(g(z_{S,i})) \tag{19}$$

Finally, the optimization is then expressed as follows:

$$\begin{cases} \theta_\varphi^\star = \arg\min_{\theta_\varphi} \mathcal{L}_c(\theta_g, \theta_\varphi | \theta_d) + \lambda \cdot \mathcal{L}_{\text{TSF}}(\theta_\varphi, \theta_{\mathbf{d}} | \theta_d, \theta_g) \\ \theta_g = \arg\min_{\theta_g} \mathcal{L}_c(\theta_g, \theta_\varphi | \theta_d) \\ \theta_d = \arg\min_{\theta_d} \mathcal{L}_{\text{INV}}(\theta_d | \theta_\varphi) \end{cases} \tag{20}$$

Losses are minimized by stochastic gradient descent (SGD) where in practice inf_d and $\text{inf}_{\mathbf{d}}$ are gradient reversal layers [13]. The trade-off parameter λ is pushed from 0 to 1 during training. We provide an implementation in `Pytorch` [29] based on [24]. The algorithm procedure is described in Appendix C.5.

6 Experiments

6.1 Setup

Datasets. We investigate two digits datasets: **MNIST** and **USPS** transfer tasks MNIST to USPS (M→U) and USPS to MNIST (U→M). We used standard train / test split for training and evaluation. **Office-31** is a dataset of images containing objects spread among 31 classes captured from different domains: **Amazon**, **DSLR** camera and a **Webcam** camera. **DSLR** and **Webcam** are very similar domains but images differ by their exposition and their quality.

Label shifted datasets. We stress-test our approach by investigating more challenging settings where the label distribution shifts strongly across domains. For the **Digits** dataset, we explore a wide variety of shifts by keeping only 5%, 10%, 15% and 20% of digits between 0 and 5 of the original dataset (refered as $\% \times [0 \sim 5]$). We have investigated the tasks U→M and M→U. For the **Office-31** dataset, we explore the shift where the object spread in classes 16 to 31 are duplicated 5 times (refered as $5 \times [16 \sim 31]$). Shifting distribution in the source domain rather than the target domain allows to better appreciate the drop in performances in the target domain compared to the case where the source domain is not shifted.

Comparison with the state-of-the-art. For all tasks, we report results from DANN [13] and CDAN [24]. To study the effect of weights, we name our method RUDA when weights are set to 1, and RUDA_w when weights are used. For the non-shifted datasets, we report a weighted version of CDAN (entropy conditioning CDAN+E [24]). For the label shifted datasets, we report IWAN [38], a weighted DANN where weights are learned from a second discriminator, and CDAN_w a weighted CDAN where weihghts are added in the same setting than RUDA_w.

Training details. Models are trained during 20.000 iterations of SGD. We report end of training accuracy in the target domain averaged on five random seeds. The model for the **Office-31** dataset uses a pretrained ResNet-50 [18]. We used the same hyper-parameters than [24] which were selected by importance weighted cross-validation [35]. The trade-off parameters λ is smoothly pushed from 0 to

Table 1. Accuracy (%) on the **Office-31** dataset.

	Method	A→W	W→A	A→D	D→A	D→W	W→D	Avg
Standard	ResNet-50	68.4 ± 0.2	60.7 ± 0.3	68.9 ± 0.2	62.5 ± 0.3	96.7 ± 0.1	99.3 ± 0.1	76.1
	DANN	82.0 ± 0.4	67.4 ± 0.5	79.7 ± 0.4	68.2 ± 0.4	96.9 ±0.2	99.1 ±0.1	82.2
	CDAN	93.1 ± 0.2	68.0 ± 0.4	89.8 ± 0.3	70.1 ± 0.4	98.2 ± 0.2	100. ± 0.0	86.6
	CDAN+E	94.1 ± 0.1	69.3 ± 0.4	**92.9 ± 0.2**	**71.0 ± 0.3**	**98.6 ± 0.1**	**100. ± 0.0**	**87.7**
	RUDA	**94.3 ± 0.3**	**70.7 ± 0.3**	92.1 ± 0.3	70.7 ± 0.1	98.5 ± 0.1	100. ± 0.0	87.6
	RUDA$_w$	92.0 ± 0.3	67.9 ± 0.3	91.1 ± 0.3	70.2 ± 0.2	98.6 ± 0.1	100. ± 0.0	86.6
$5 \times [16 \sim 31]$	ResNet-50	72.4 ± 0.7	59.5 ± 0.1	79.0 ± 0.1	61.6 ± 0.3	97.8 ± 0.1	99.3 ± 0.1	78.3
	DANN	67.5 ± 0.1	52.1 ± 0.8	69.7 ± 0.0	51.5 ± 0.1	89.9 ±0.1	75.9 ±0.2	67.8
	CDAN	82.5 ± 0.4	62.9 ± 0.6	81.4 ± 0.5	65.5 ± 0.5	98.5 ± 0.3	99.8 ± 0.0	81.6
	RUDA	85.4 ± 0.8	66.7 ± 0.5	81.3 ± 0.3	64.0 ± 0.5	98.4 ± 0.2	99.5 ± 0.1	82.1
	IWAN	72.4 ± 0.4	54.8 ± 0.8	75.0 ± 0.3	54.8 ± 1.3	97.0 ±0.0	95.8 ±0.6	75.0
	CDAN$_w$	81.5 ±0.5	64.5 ± 0.4	80.7 ± 1.0	65 ± 0.8	**98.7 ± 0.2**	99.9 ± 0.1	81.8
	RUDA$_w$	**87.4 ± 0.2**	**68.3 ± 0.3**	**82.9** ± 0.4	**68.8 ± 0.2**	**98.7** ± 0.1	**100.** ± 0.0	**83.8**

1 as detailed in [24]. To prevent from noisy weighting in early learning, we used weight relaxation: based on the sigmoid output of discriminator $d(z) = \sigma(\tilde{d}(z))$, we used $d_\tau(z) = \sigma(\tilde{d}(z/\tau))$ and weights $w(z) = (1 - d_\tau(z))/d_\tau(z)$. τ is decreased to 1 during training: $\tau = \tau_{\min} + 2(\tau_{\max} - \tau_{\min})/(1 + \exp(-\alpha p))$ where $\tau_{\max} = 5, \tau_{\min} = 1$, $p \in [0, 1]$ is the training progress. In all experiments, α is set to 5 (except for $5\% \times [0 \sim 5]$ where $\alpha = 15$, see Appendix C.3 for more details).

6.2 Results

Unshifted datasets. On both **Office-31** (Table 1) and **Digits** (Table 2), RUDA performs similarly than CDAN. Simply performing the scalar product allows to achieve results obtained by multi-linear conditioning [24]. This presents a second advantage: when domains exhibit a large number of classes, *e.g.* in **Office-Home** (See Appendix), our approach does not need to leverage a random layer. It is interesting to observe that we achieve performances close to CDAN+E on **Office-31** while we do not use entropy conditioning. However, we observe a substantial drop in performance when adding weights, but still get results comparable with CDAN in **Office-31**. This is a deceiptive result since those datasets naturally exhibit label shift; one can expect to improve the baselines using weights. We did not observe this phenomenon on standard benchmarks.

Label shifted datasets. We stress-tested our approach by applying strong label shifts to the datasets. First, we observe a drop in performance for all methods

Table 2. Accuracy (%) on the **Digits** dataset.

Method	U→ M						M→U						
Shift of $[0 \sim 5]$	5%	10%	15%	20%	100%	Avg	5%	10%	15%	20%	100%	Avg	Avg
DANN	41.7	51.0	59.6	69.0	94.5	63.2	34.5	51.0	59.6	63.6	90.7	59.9	63.2
CDAN	50.7	62.2	82.9	82.8	**96.9**	75.1	32.0	69.7	78.9	81.3	**93.9**	71.2	73.2
RUDA	44.4	58.4	80.0	84.0	95.5	72.5	34.9	59.0	76.1	78.8	93.3	68.4	70.5
IWAN	73.7	74.4	78.4	77.5	95.7	79.9	72.2	82.0	84.3	86.0	92.0	83.3	81.6
CDAN$_w$	68.3	78.8	84.9	**88.4**	96.6	83.4	69.4	80.0	83.5	87.8	93.7	82.9	83.2
RUDA$_w$	**78.7**	**82.8**	**86.0**	86.9	93.9	**85.7**	**78.7**	**87.9**	**88.2**	**89.3**	92.5	**87.3**	**86.5**

based on invariant representations compared with the situation without label shift. This is consistent with works that warn the pitfall of domain invariant representations in presence of label shift [20,40]. RUDA and CDAN perform similarly even in this setting. It is interesting to note that the weights improve significantly RUDA results (+1.7% on **Office-31** and +16.0% on **Digits** both in average) while CDAN seems less impacted by them (+0.2% on **Office-31** and +10.0% on **Digits** both in average).

Should we use weights? To observe a significant benefit of weights, we had to explore situations with strong label shift *e.g.* 5% and 10% $\times$ $[0 \sim 5]$ for the **Digits** dataset. Apart from this cases, weights bring small gain (*e.g.* + 1.7% on **Office-31** for RUDA) or even degrade marginally adaptation. Understanding why RUDA and CDAN are able to address small label shift, without weights, is of great interest for the development of more robust UDA.

7 Related work

This paper makes several contributions, both in terms of theory and algorithm. Concerning theory, our bound provides a risk suitable for domain adversarial learning with weighting strategies. Existing theories for non-overlapping supports [4,27] and importance sampling [11,30] do not explore the role of representations neither the aspect of adversarial learning. In [5], analysis of representation is conducted and connections with our work is discussed in the paper. The work [20] is close to ours and introduces a distance which measures support overlap between source and target distributions under covariate shift. Our analysis does not rely on such assumption, its range of application is broader.

Concerning algorithms, the covariate shift adaptation has been well-studied in the literature [19,17,35]. Importance sampling to address label shift has also been investigated [34], notably with kernel mean matching [39] and Optimal Transport [31]. Recently, a scheme for estimating labels distribution ratio with consistency guarantee has been proposed [21]. Learning domain invariant representations has also been investigated in the fold of [13,23] and mainly differs by the metric chosen for comparing distribution of representations. For instance, metrics are domain adversarial (Jensen divergence) [13,24], IPM based such as MMD [23,25] or Wasserstein [6,32]. Our work provides a new theoretical support for these methods since our analysis is valid for any IPM.

Using both weights and representations is also an active topic, namely for Partial Domain Adaptation (PADA) [9], when target classes are strict subset of the source classes, or Universal Domain Adaptation [37], when new classes may appear in the target domain. [9] uses an heuristic based on predicted labels for re-weighting representations. However, it assumes they have a good classifier at first in order to obtain cycle consistent weights. [38] uses a second discriminator for learning weights, which is similar to [8]. Applying our framework to Partial DA and Universal DA is an interesting future direction. Our work shares strong connections with [10] (authors were not aware of this work during the elaboration

of this paper) which uses consistent estimation of true labels distribution from [21]. We suggest a very similar empirical evaluation and we also investigate the effect of weights on CDAN loss [24] with a different weighting scheme since our approach computes weights in the representation space. All these works rely on an assumption at some level, *e.g. Generalized Label Shift* in [10], when designing weighting strategies. Our discussion on the role of inductive design of weights may provide a new theoretical support for these approaches.

8 Conclusion

The present work introduces a new bound of the target risk which unifies weights and representations in UDA. We conduct a theoretical analysis of the role of inductive bias when designing both weights and the classifier. In light of this analysis, we propose a new learning procedure which leverages two weak inductive biases, respectively on weights and the classifier. To the best of our knowledge, this procedure is original while being close to straightforward hybridization of existing methods. We illustrate its effectiveness on two benchmarks. The empirical analysis shows that weak inductive bias can make adaptation more robust even when stressed by strong label shift between source and target domains. This work leaves room for in-depth study of stronger inductive bias by providing both theoretical and empirical foundations.

Acknowledgements

Victor Bouvier is funded by Sidetrade and ANRT (France) through a CIFRE collaboration with CentraleSuplec. Authors thank the anonymous reviewers for their insightful comments for improving the quality of the paper. This work was performed using HPC resources from the Msocentre computing center of CentraleSuplec and cole Normale Suprieure Paris-Saclay supported by CNRS and Rgion le-de-France (http://mesocentre.centralesupelec.fr/).

References

1. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565 (2016)
2. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant risk minimization. arXiv preprint arXiv:1907.02893 (2019)
3. Beery, S., Van Horn, G., Perona, P.: Recognition in terra incognita. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 456–473 (2018)
4. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine learning **79**(1-2), 151–175 (2010)
5. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in neural information processing systems. pp. 137–144 (2007)

6. Bhushan Damodaran, B., Kellenberger, B., Flamary, R., Tuia, D., Courty, N.: Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 447–463 (2018)
7. Bottou, L., Arjovsky, M., Lopez-Paz, D., Oquab, M.: Geometrical insights for implicit generative modeling. In: Braverman Readings in Machine Learning. Key Ideas from Inception to Current State, pp. 229–268. Springer (2018)
8. Cao, Y., Long, M., Wang, J.: Unsupervised domain adaptation with distribution matching machines. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
9. Cao, Z., Ma, L., Long, M., Wang, J.: Partial adversarial domain adaptation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–150 (2018)
10. Combes, R.T.d., Zhao, H., Wang, Y.X., Gordon, G.: Domain adaptation with conditional distribution matching and generalized label shift. arXiv preprint arXiv:2003.04475 (2020)
11. Cortes, C., Mansour, Y., Mohri, M.: Learning bounds for importance weighting. In: Advances in neural information processing systems. pp. 442–450 (2010)
12. D'Amour, A., Ding, P., Feller, A., Lei, L., Sekhon, J.: Overlap in observational studies with high-dimensional covariates. arXiv preprint arXiv:1711.02582 (2017)
13. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning. pp. 1180–1189 (2015)
14. Geva, M., Goldberg, Y., Berant, J.: Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 1161–1166 (2019)
15. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Advances in neural information processing systems. pp. 529–536 (2005)
16. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. Journal of Machine Learning Research **13**(Mar), 723–773 (2012)
17. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: Covariate shift by kernel mean matching. Dataset shift in machine learning **3**(4), 5 (2009)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
19. Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: Advances in neural information processing systems. pp. 601–608 (2007)
20. Johansson, F., Sontag, D., Ranganath, R.: Support and invertibility in domain-invariant representations. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 527–536 (2019)
21. Lipton, Z., Wang, Y.X., Smola, A.: Detecting and correcting for label shift with black box predictors. In: International Conference on Machine Learning. pp. 3122–3130 (2018)
22. Liu, H., Long, M., Wang, J., Jordan, M.: Transferable adversarial training: A general approach to adapting deep classifiers. In: International Conference on Machine Learning. pp. 4013–4022 (2019)

23. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37. pp. 97–105. JMLR. org (2015)
24. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: Advances in Neural Information Processing Systems. pp. 1640–1650 (2018)
25. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 2208–2217. JMLR. org (2017)
26. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
27. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation: Learning bounds and algorithms. In: 22nd Conference on Learning Theory, COLT 2009 (2009)
28. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2009)
29. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8024–8035 (2019)
30. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning. The MIT Press (2009)
31. Redko, I., Courty, N., Flamary, R., Tuia, D.: Optimal transport for multi-source domain adaptation under target shift. arXiv preprint arXiv:1803.04899 (2018)
32. Shen, J., Qu, Y., Zhang, W., Yu, Y.: Wasserstein distance guided representation learning for domain adaptation. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
33. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference **90**(2), 227–244 (2000)
34. Storkey, A.: When training and test sets are different: characterizing learning transfer. Dataset shift in machine learning pp. 3–28 (2009)
35. Sugiyama, M., Krauledat, M., MÃžller, K.R.: Covariate shift adaptation by importance weighted cross validation. Journal of Machine Learning Research **8**(May), 985–1005 (2007)
36. Wu, Y., Winston, E., Kaushik, D., Lipton, Z.: Domain adaptation with asymmetrically-relaxed distribution alignment. In: International Conference on Machine Learning. pp. 6872–6881 (2019)
37. You, K., Long, M., Cao, Z., Wang, J., Jordan, M.I.: Universal domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2720–2729 (2019)
38. Zhang, J., Ding, Z., Li, W., Ogunbona, P.: Importance weighted adversarial nets for partial domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8156–8164 (2018)
39. Zhang, K., Schölkopf, B., Muandet, K., Wang, Z.: Domain adaptation under target and conditional shift. In: International Conference on Machine Learning. pp. 819–827 (2013)
40. Zhao, H., Des Combes, R.T., Zhang, K., Gordon, G.: On learning invariant representations for domain adaptation. In: International Conference on Machine Learning. pp. 7523–7532 (2019)

A Proofs

We provide full proof of bounds and propositions presented in the paper.

A.1 Proof of bound 2

We give a proof of bound 2 which states:

$$\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi) + 6 \cdot \mathrm{INV}(\varphi) + 2 \cdot \mathrm{TSF}(\varphi) + \varepsilon_T(\mathbf{f}_T\varphi) \tag{21}$$

First, we prove the following lemma:

Bound 5 (Revisit of theorem 1) *$\forall g \in \mathcal{G}$:*

$$\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi) + d_{\mathcal{F}_C}(\varphi) + \varepsilon_T(\mathbf{f}_S\varphi, \mathbf{f}_T\varphi) + \varepsilon_T(\mathbf{f}_T\varphi) \tag{22}$$

Proof. This is simply obtained using triangular inequalites:

$$\begin{aligned}\varepsilon_T(g\varphi) &\leq \varepsilon_T(\mathbf{f}_T\varphi) + \varepsilon_T(g\varphi, \mathbf{f}_T\varphi) \\ &\leq \varepsilon_T(\mathbf{f}_T\varphi) + \varepsilon_T(g\varphi, \mathbf{f}_S\varphi) + \varepsilon_T(\mathbf{f}_S\varphi, \mathbf{f}_T\varphi)\end{aligned}$$

Now using (A3) ($\mathbf{f}_S \in \mathcal{F}_C$) :

$$|\varepsilon_T(g\varphi, \mathbf{f}_S\varphi) - \varepsilon_S(g\varphi, \mathbf{f}_S\varphi)| \leq \sup_{\mathbf{f} \in \mathcal{F}_C} |\varepsilon_T(g\varphi, \mathbf{f}\varphi) - \varepsilon_S(g\varphi, \mathbf{f}\varphi)| = d_{\mathcal{F}_C}(\varphi) \tag{23}$$

which shows that: $\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi, \mathbf{f}_S) + d_{\mathcal{F}_C}(\varphi) + \varepsilon_T(\mathbf{f}_S\varphi, \mathbf{f}_T\varphi) + \varepsilon_T(\mathbf{f}_T\varphi)$ and we use the property of conditional expectation $\varepsilon_S(g\varphi, \mathbf{f}_S\varphi) \leq \varepsilon_S(g\varphi)$. □

Second, we bound $d_{\mathcal{F}_C}(\varphi)$.

Proposition 7. $d_{\mathcal{F}_C}(\varphi) \leq 4 \cdot \mathrm{INV}(\varphi)$.

Proof. We remind that $d_{\mathcal{F}_C}(\varphi) = \sup_{\mathbf{f},\mathbf{f}' \in \mathcal{F}_C} |\mathbb{E}_S[||\mathbf{f}\varphi(X) - \mathbf{f}'\varphi(X)||^2] - \mathbb{E}_T[||\mathbf{f}\varphi(X) - \mathbf{f}'\varphi(X)||^2]|$. Since (A1) ensures $\mathbf{f}' \in \mathcal{F}_C$, $-\mathbf{f}' \in \mathcal{F}_C$, then $\frac{1}{2}(\mathbf{f} - \mathbf{f}') = \mathbf{f}'' \in \mathcal{F}_C$ and finally $d_{\mathcal{F}_C}(\varphi) \leq 4 \sup_{\mathbf{f}'' \in \mathcal{F}_C} |\mathbb{E}_S[||\mathbf{f}''\varphi||^2] - \mathbb{E}_T[||\mathbf{f}''\varphi||^2]|$. Furthermore, (A2) ensures that $\{||\mathbf{f}''\varphi||^2\} \subset \{f\varphi, f \in \mathcal{F}\}$ which leads finally to the announced result. □

Third, we bound $\varepsilon_T(\mathbf{f}_S\varphi, \mathbf{f}_T\varphi)$.

Proposition 8. $\varepsilon_T(\mathbf{f}_S\varphi, \mathbf{f}_T\varphi) \leq 2 \cdot \mathrm{INV}(\varphi) + 2 \cdot \mathrm{TSF}(\varphi)$.

Proof. We note $\Delta = \mathbf{f}_T - \mathbf{f}_S$ and we omit φ for the ease of reading

$$\begin{aligned}\varepsilon_T(\mathbf{f}_S, \mathbf{f}_T) &= \mathbb{E}_T[||\Delta||^2] \\ &= \mathbb{E}_T[\mathbf{f}_T \cdot \Delta] - \mathbb{E}_T[\mathbf{f}_S \cdot \Delta] \\ &= (\mathbb{E}_T[\mathbf{f}_T \cdot \Delta] - \mathbb{E}_S[\mathbf{f}_S \cdot \Delta]) + (\mathbb{E}_S[\mathbf{f}_S \cdot \Delta] - \mathbb{E}_T[\mathbf{f}_S \cdot \Delta])\end{aligned}$$

Since $\mathbf{f}_T$ does not intervene in $\mathbb{E}_S[\mathbf{f}_S \cdot \Delta] - \mathbb{E}_T[\mathbf{f}_S \cdot \Delta]$, we show this term behaves similarly than $\mathrm{INV}(\varphi)$. First,

$$\begin{aligned}
\mathbb{E}_S[\mathbf{f}_S \cdot \Delta] - \mathbb{E}_T[\mathbf{f}_S \cdot \Delta] &\leq 2 \sup_{\mathbf{f} \in \mathcal{F}_C} \mathbb{E}_S[\mathbf{f}_S \cdot \mathbf{f}] - \mathbb{E}_T[\mathbf{f}_S \cdot \mathbf{f}] && \text{(Using (A1))} \\
&\leq 2 \sup_{\mathbf{f}, \mathbf{f}' \in \mathcal{F}_C} \mathbb{E}_S[\mathbf{f}' \cdot \mathbf{f}] - \mathbb{E}_T[\mathbf{f}' \cdot \mathbf{f}] && \text{(Using (A3))} \\
&\leq 2 \sup_{f \in \mathcal{F}} \mathbb{E}_S[f] - \mathbb{E}_T[f] && \text{(Using (A2))} \\
&= 2 \cdot \mathrm{INV}(\varphi) && (24)
\end{aligned}$$

Second,

$$\mathbb{E}_T[\mathbf{f}_T \cdot \Delta] - \mathbb{E}_S[\mathbf{f}_S \cdot \Delta] \leq 2 \sup \mathbb{E}_T[\mathbf{f}_T \cdot \mathbf{f}] - \mathbb{E}_S[\mathbf{f}_S \cdot \mathbf{f}] = 2 \cdot \mathrm{TSF}(\varphi) \quad \text{(Using (A1))}$$

which finishes the proof. □

Note that the fact $\mathbf{f}_S, \mathbf{f}_T \in \mathcal{F}_C$ is not of the utmost importance since we can bound:

$$\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi, \hat{\mathbf{f}}_S) + d_{\mathcal{F}_C}(\varphi) + \varepsilon_T(\hat{\mathbf{f}}_S, \hat{\mathbf{f}}_T) + \varepsilon_T(\hat{\mathbf{f}}_T) \tag{25}$$

where $\hat{\mathbf{f}}_D = \arg\min_{\mathbf{f} \in \mathcal{F}_C} \varepsilon_D(\mathbf{f})$. The only change emerges in the transferability error which becomes:

$$\mathrm{TSF}(w, \varphi) = \sup_{\mathbf{f} \in \mathcal{F}_C} \mathbb{E}_T[\hat{\mathbf{f}}_T\varphi \cdot \mathbf{f}\varphi] - \mathbb{E}_S[\hat{\mathbf{f}}_S\varphi \cdot \mathbf{f}\varphi] \tag{26}$$

A.2 Proof of the new invariance transferability trade-off

Proposition 9. *Let ψ a representation which is a richer feature extractor than φ: $\mathcal{F} \circ \varphi \subset \mathcal{F} \circ \psi$ and $\mathcal{F}_C \circ \varphi \subset \mathcal{F}_C \circ \psi$. Then, φ is more domain invariant than ψ:*

$$\mathrm{INV}(\varphi) \leq \mathrm{INV}(\psi) \text{ while } \varepsilon_T(f_T^\psi \psi) \leq \varepsilon_T(f_T^\varphi \varphi) \tag{27}$$

where $f_T^\varphi(z) = \mathbb{E}_T[Y|\varphi(X) = z]$ and $f_T^\psi(z) = \mathbb{E}_T[Y|\psi(X) = z]$.

Proof. First, $\mathrm{INV}(\varphi) \leq \mathrm{INV}(\psi)$ a simple property of the supremum. The definition of the conditional expectation leads to $\varepsilon_T(\mathbf{f}_T^\psi \psi) = \inf_{f \in \mathcal{F}_m} \varepsilon_T(f\psi)$ where $\mathcal{F}_m$ is the set of measurable functions. Since (A3) ensures that $\mathbf{f}_T^\psi \in \mathcal{F}_C$ then $\varepsilon_T(\mathbf{f}_T^\psi \psi) = \inf_{\mathbf{f} \in \mathcal{F}_C} \varepsilon_T(\mathbf{f}\psi)$. The rest is simply the use of the property of infremum. □

A.3 Proof of the tightness of bound 2

Proposition 10. $\mathrm{INV}(\varphi) + \mathrm{TSF}(\varphi) = 0$ *if and only if* $p_S(y, z) = p_T(y, z)$.

Proof. First, $\text{INV}(\varphi) = 0$ implies $p_T(z) = p_S(z)$ which is a direct application of (A4). Now $\text{TSF}(\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[\mathbf{f}_S(Z)\cdot\mathbf{f}(Z)] - \mathbb{E}_T[\mathbf{f}_T(Z)\cdot\mathbf{f}(Z)] = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[\mathbf{f}_S(Z)\cdot\mathbf{f}(Z)] - \mathbb{E}_S[\mathbf{f}_T(Z)\cdot\mathbf{f}(Z)] = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[(\mathbf{f}_S - \mathbf{f}_T)(Z)\cdot\mathbf{f}(Z)]$. For the particular choice of $\mathbf{f} = \frac{1}{2}(\mathbf{f}_S - \mathbf{f}_T)$ leads to $\mathbb{E}_S[||\mathbf{f}_S - \mathbf{f}_T||^2]$ then $\mathbf{f}_s = \mathbf{f}_T$, p_S almost surely. All combined leads to $p_S(y,z) = p_T(y,z)$. The converse is trivial. Note that $\text{TSF}(\varphi) = 0$ is enough to show $p_S(z) = p_T(z)$ by choosing $\mathbf{f}(z) = (f(z), ..., f(z))$ (C times $f(z)$) and $Y\cdot\mathbf{f}(Z) = f(Z)$ then $\text{TSF}(\varphi) \geq \sup_{f\in\mathcal{F}} \mathbb{E}_S[f(Z)] - \mathbb{E}_T[f(Z)]$. □

A.4 Proof of the tightness of bound 3

Proposition 11. $\text{INV}(w,\varphi) + \text{TSF}(w,\varphi) = 0$ *if and only if* $w(z) = \frac{p_T(z)}{p_S(z)}$ *and* $\mathbb{E}_T[Y|Z=z] = \mathbb{E}_S[Y|Z=z]$.

Proof. First, $\text{INV}(w,\varphi) = 0$ implies $p_T(z) = w(z)p_S(z)$ then which is a direct application of (A4). Now $\text{TSF}(w,\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(z)\mathbf{f}_S(Z)\cdot\mathbf{f}(Z)] - \mathbb{E}_T[\mathbf{f}_T(Z)\cdot\mathbf{f}(Z)] = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(z)\mathbf{f}_S(Z)\cdot\mathbf{f}(Z)] - \mathbb{E}_S[w(z)\mathbf{f}_T(Z)\cdot\mathbf{f}(Z)] = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[(\mathbf{f}_S - \mathbf{f}_T)(Z)\cdot\mathbf{f}(Z)]$. For the particular choice of $\mathbf{f} = \frac{1}{2}(\mathbf{f}_S - \mathbf{f}_T)$ leads to $\mathbb{E}_S[||\mathbf{f}_S - \mathbf{f}_T||^2]$ then $\mathbf{f}_s = \mathbf{f}_T$, p_T almost surely. The converse is trivial. □

A.5 Proof of bound 4

Bound 6 (Inductive Bias and Guarantee) *Let* $\varphi \in \Phi$ *and* $w : \mathcal{Z} \to \mathbb{R}^+$ *such that* $\mathbb{E}_S[w(z)] = 1$ *and a* $\beta-$*strong inductive classifier* $\tilde{g}$*, then:*

$$\varepsilon_T(\tilde{g}\varphi) \leq \frac{\beta}{1-\beta}\left(\varepsilon_{w\cdot S}(g_{w\cdot S}\varphi) + 6\cdot\text{INV}(w,\varphi) + 2\cdot\widehat{\text{TSF}}(w,\varphi,\tilde{g}) + \varepsilon_T(\mathbf{f}_T\varphi)\right)$$

Proof. We prove the bound in the case where $w = 1$, the general case is then straightforward. First, we reuse bound 5 with a new triangular inequality involving the inductive classifier $\tilde{g}$:

$$\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi) + d_{\mathcal{F}_C}(\varphi) + \varepsilon_T(\mathbf{f}_S\varphi, \tilde{g}\varphi) + \varepsilon_T(\tilde{g}\varphi, \mathbf{f}_T\varphi) + \varepsilon_T(\mathbf{f}_T\varphi) \tag{28}$$

where $\varepsilon_T(\tilde{g}\varphi, \mathbf{f}_T\varphi) \leq \varepsilon_T(\tilde{g}\varphi)$. Now, following previous proofs, we can show that:

$$\varepsilon_T(\mathbf{f}_S\varphi, \tilde{g}\varphi) \leq 2\cdot\widehat{\text{TSF}}(\varphi,\tilde{g}) + 2\cdot\text{INV}(\varphi) \tag{29}$$

Then,

$$\varepsilon_T(g\varphi) \leq \varepsilon_S(g\varphi) + 6\cdot\text{INV}(\varphi) + 2\cdot\widehat{\text{TSF}}(\varphi,\tilde{g}) + \varepsilon_T(\tilde{g}\varphi) + \varepsilon_T(\mathbf{f}_T\varphi) \tag{30}$$

This bound is true for any g and in particular for the best source classifier we have:

$$\varepsilon_T(g_S\varphi) \leq \varepsilon_S(g_S\varphi) + 6\cdot\text{INV}(w,\varphi) + 2\cdot\widehat{\text{TSF}}(w,\varphi,\tilde{g}) + \varepsilon_T(\tilde{g}\varphi) + \varepsilon_T(\mathbf{f}_T\varphi) \tag{31}$$

then the assumption of $\beta-$strong inductive bias is $\varepsilon_T(\tilde{g}\varphi) \leq \beta\varepsilon_T(g_S\varphi)$ which leads to

$$\varepsilon_T(g_S\varphi) \leq \varepsilon_S(g_S\varphi) + 6 \cdot \text{INV}(w, \varphi) + 2 \cdot \widehat{\text{TSF}}(w, \varphi, \tilde{g}) + \beta\varepsilon_T(g_S\varphi) + \varepsilon_T(\mathbf{f}_T\varphi) \quad (32)$$

Now we have respectively $\varepsilon_T(g_S\varphi)$ and $\beta\varepsilon_T(g_S\varphi)$ at left and right of the inequality. Since $1 - \beta > 0$, we have:

$$\varepsilon_T(g_S\varphi) \leq \frac{1}{1-\beta}\left(\varepsilon_S(g_S\varphi) + 6 \cdot \text{INV}(w, \varphi) + 2 \cdot \widehat{\text{TSF}}(w, \varphi, \tilde{g}) + \varepsilon_T(\mathbf{f}_T\varphi)\right) \quad (33)$$

And finally:

$$\varepsilon_T(\tilde{g}\varphi) \leq \beta\varepsilon_T(g_S\varphi) \leq \frac{\beta}{1-\beta}\left(\varepsilon_S(g_S\varphi) + 6 \cdot \text{INV}(w, \varphi) + 2 \cdot \widehat{\text{TSF}}(w, \varphi, \tilde{g}) + \varepsilon_T(\mathbf{f}_T\varphi)\right) \quad (34)$$

fnishing the proof. □

A.6 MinEnt [15] is a lower bound of transferability

Proof. We consider a label smooth classifier $g \in \mathcal{G}$ *i.e.* there is $0 < \alpha < 1$ such that:

$$\frac{\alpha}{C-1} \leq g(z) \leq 1 - \alpha \quad (35)$$

and we note $Y = g\varphi(X)$. One can show that:

$$\log\left(\frac{\alpha}{C-1}\right) \leq \log(g(z)) \leq \log(1-\alpha) \quad (36)$$

and finally:

$$1 \geq \frac{1}{\log(\frac{\alpha}{C-1})}\log(g(z)) \geq \frac{1}{\log(\frac{\alpha}{C-1})}\log(1-\alpha) \geq 0 \quad (37)$$

We choose as particular $\mathbf{f}$, $\mathbf{f}(z) = -\eta\log(g(z))$ with $\eta = -\log(\frac{\alpha}{|\mathcal{Y}|-1})^{-1} > 0$. The coefficient η ensures that $\mathbf{f}(z) \in [0, 1]$ to make sure $\mathbf{f} \in \mathcal{F}_C$. We have the following inequalities:

$$\begin{aligned}\widehat{\text{TSF}}(w, \varphi, g) &\geq \eta \cdot (\mathbb{E}_T[-g(Z) \cdot \log(g(Z))] - \mathbb{E}_{w \cdot S}[-Y \log(g(Z))]) \\ &\geq \eta \cdot \left(H_T(\hat{Y}|Z) - \text{CE}_{w \cdot S}(Y, g(Z))\right)\end{aligned}$$

Interestingly, the cross-entropy is involved. Then, when using $\text{CE}_{w \cdot S}(Y, g(Z))$ as a proxy of $\varepsilon_{w \cdot S}(g\varphi)$, we can observe the following lower bound:

$$\text{CE}_{w \cdot S}(Y, g(Z)) + \widehat{\text{TSF}}(w, \varphi, g) \geq (1-\eta) \cdot \text{CE}_{w \cdot S}(Y, g(Z)) + \eta \cdot H_T(\hat{Y}|Z) \quad (38)$$

which is a trade-off between minimizing the cross-entropy in the source domain while maintaining a low entropy in prediction in the target domain.

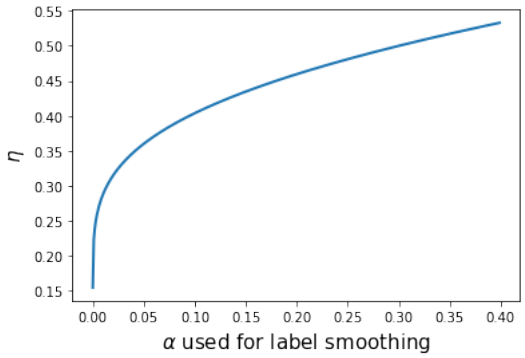


Fig. 2. We set $C = 31$ which is the number of classes in **Office31**. Label smoothing α leads naturally to a coefficient η which acts as a trade-off between cross-entropy minimization in the source domain and confidence in predictions in the target domain. This result follows a particular choice of the critic function in the transferability error introduced in this paper.

A.7 Proof of the inductive design of weights

Proposition 12 (Inductive design of w and invariance). *Let $\psi : \mathcal{Z} \to \mathcal{Z}'$ such that $\mathcal{F} \circ \psi \subset \mathcal{F}$ and $\mathcal{F}_C \circ \psi \subset \mathcal{F}_C$. Let $w : \mathcal{Z}' \to \mathbb{R}^+$ such that $\mathbb{E}_S[w(Z')] = 1$ and we note $Z' := \psi(Z)$. Then,* $\mathrm{INV}(w, \varphi) = \mathrm{TSF}(w, \varphi) = 0$ *if and only if:*

$$w(z') = \frac{p_T(z')}{p_S(z')} \quad \textit{and} \quad p_S(z|z') = p_T(z|z') \tag{39}$$

while both $\mathbf{f}_S^\varphi = \mathbf{f}_T^\varphi$ and $\mathbf{f}_S^\psi = \mathbf{f}_T^\psi$.

Proof. First,

$$\begin{aligned}
\mathrm{INV}(w, \varphi) &= \sup_{f \in \mathcal{F}} \mathbb{E}_S[w(Z')f(Z)] - \mathbb{E}_T[f(z)] && (40)\\
&\geq \sup_{f \in \mathcal{F}} \mathbb{E}_S[w(Z')f \circ \psi(Z)] - \mathbb{E}_T[f \circ \psi(z)] && (\mathcal{F} \circ \psi \subset \mathcal{F})\\
&= \sup_{f \in \mathcal{F}} \mathbb{E}_S[w(Z')f(Z')] - \mathbb{E}_T[f(z')] = 0 && (Z' = \psi(Z))
\end{aligned}$$

which leads to $w(z')p_S(z') = p_S(z')$ which is $w(z') = p_T(z')/p_S(z')$. Second, $\mathrm{INV}(w, \varphi) = 0$ also implies that $w(z')p_S(z) = p_T(z)$:

$$w(z') = \frac{p_T(z)}{p_S(z)} = \frac{p_T(z|z')}{p_S(z|z')}\frac{p_T(z')}{p_S(z')} = \frac{p_T(z|z')}{p_S(z|z')}w(z') \tag{41}$$

then $p_T(z|z') = p_S(z|z')$. Finally,

$$\begin{aligned}
\mathrm{TSF}(w,\varphi) &= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')Y\cdot\mathbf{f}(Z)] - \mathbb{E}_T[Y\cdot\mathbf{f}(Z)] && (42)\\
&= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\mathbb{E}_{Z|Z'\sim p_S}[Y\cdot\mathbf{f}(Z)]\right] - \mathbb{E}_{Z'\sim p_T}\left[\mathbb{E}_{Z|Z'\sim p_T}[Y\cdot\mathbf{f}(Z)]\right] && (43)\\
&= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\mathbb{E}_{Z|Z'\sim p_S}[Y\cdot\mathbf{f}(Z)]\right] - \mathbb{E}_{Z'\sim p_T} w(Z')\left[\mathbb{E}_{Z|Z'\sim p_S}[Y\cdot\mathbf{f}(Z)]\right] \\
&\qquad\qquad (w(z')p_S(z') = p_T(z')) && (44)\\
&= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[Y\cdot\mathbf{f}(Z)] - \mathbb{E}_{Z|Z'\sim p_T}[Y\cdot\mathbf{f}(Z)]\right)\right] \\
&= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{f}_S(Z)\cdot\mathbf{f}(Z) - \mathbf{f}_T(Z)\cdot\mathbf{f}(Z)]\right)\right] \\
&\qquad\qquad (p_S(z|z') = p_T(z|z')) \\
&= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{f}_S(Z)\cdot\mathbf{f}(Z) - \mathbf{f}_T(Z)\cdot\mathbf{f}(Z)]\right)\right] \\
&\qquad\qquad (p_S(z|z') = p_T(z|z')) \\
&\geq 2\mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2]\right)\right] && (45)\\
&\geq 2\mathbb{E}_{Z'\sim p_T}\left[\left(\mathbb{E}_{Z|Z'\sim p_T}[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2]\right)\right] && (46)\\
&\geq 2\mathbb{E}_{Z'\sim p_T}\left[\left(\mathbb{E}_{Z|Z'\sim p_T}[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2]\right)\right] && (47)\\
&\geq 2\mathbb{E}_{Z\sim p_T}\left[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2\right] && (48)
\end{aligned}$$

Which leads to $\mathbf{f}_S(z) = \mathbf{f}_T(z)$, $p_T(z)$ almost surely, then $\mathbb{E}_T[Y|Z] = \mathbb{E}_S[Y|Z]$ for $Z \sim p_T$. Now we finish by observing that:

$$\begin{aligned}
\mathrm{TSF}(w,\varphi) &= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')Y\cdot\mathbf{f}(Z)] - \mathbb{E}_T[Y\cdot\mathbf{f}(Z)] && (49)\\
&\geq \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')Y\cdot\mathbf{f}\circ\psi(Z)] - \mathbb{E}_T[Y\cdot\mathbf{f}\circ\psi(Z)] && (50)\\
&\geq \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')Y\cdot\mathbf{f}(Z')] - \mathbb{E}_T[Y\cdot\mathbf{f}(Z')] && (51)
\end{aligned}$$

which leads to $\mathbb{E}_S[Y|Z'] = \mathbb{E}_T[Y|Z']$ for $Z' \sim p_T$. The converse is trivial. □

B CDAN, DANN and TSF: An open dicussion.

In CDAN [24], authors claims to align conditional $Z|\hat{Y}$, by exposing the multilinear mapping of $\hat{Y}$ by Z, hence its name of Conditional Domain Adversarial Network. Here, we show this claim can be theoretically misleading:

Proposition 13. *If $\mathbb{E}[\hat{Y}|Z]$ is conserved across domains, i.e. g is conserved, and $\mathcal{D}$ and $\mathcal{D}_\otimes$ are infinite capacity set of discriminators, this holds:*

$$\mathrm{DANN}(\varphi) = \mathrm{CDAN}(\varphi) \tag{52}$$

Proof. First, let $d_\otimes \in \mathcal{D}_\otimes$. Then, for any $(\hat{y}, z) \sim p_S$ (similarly $\sim p_T$), $d(\hat{y} \otimes z) = d(g(z) \otimes z)$ since $\hat{y} = g(z) = \mathbb{E}[\hat{Y}|Z = z]$ is conserved across domains. Then $\tilde{d} : z \mapsto d_\otimes(g(z) \otimes z)$ is a mapping from $\mathcal{Z}$ to $[0, 1]$. Since $\mathcal{D}$ is the set of infinite capacity discriminators, $\tilde{d} \in \mathcal{D}$. This shows $\text{CDAN}(\varphi) \leq \text{DANN}(\varphi)$. Now we introduce $T : \mathcal{Y} \otimes \mathcal{Z} \to \mathcal{Z}$ such that $T(y \otimes z) = \sum_{1 \leq c \leq |\mathcal{Y}|} y_c (y \otimes z)_{cr:(c+1)r} = z$ where $r = \dim(Z)$. The ability to reconstruct z from $\hat{y} \otimes z$ results from $\sum_c y_c = 1$. This shows that $\mathcal{D}_\otimes \circ T = \mathcal{D}$ and finally $\text{CDAN}(\varphi) \geq \text{DANN}(\varphi)$ finishing the proof.

This proposition follows two key assumptions. The first is to assume that we are in context of infinite capacity discriminators of both $\mathcal{Z}$ and $\mathcal{Y} \otimes \mathcal{Z}$. This assumption seems reasonable in practice since discriminators are multi-layer perceptrons. The second is to assume that $\mathbb{E}[\hat{Y}|Z]$ is conserved across domains. Pragmatically, the same classifier is used in both source and target domains which is verified in practice. Despite the empirical success of CDAN, there is no theoretical evidence of the superiority of CDAN with respect to DANN for UDA. However, our discussion on the role of inductive design of classifiers is an attempt to explain the empirical superiority of such strategies.

C More training details

C.1 From IPM to Domain Adversarial Objective

While our analysis holds for IPM, we recall the connections with $\mathsf{f}-$divergence, where domain adversarial loss is a particular instance, for comparing distributions. This connection is motivated by the furnished literature on adversarial learning, based on domain discriminator, for UDA. This section is then an informal attempt to transport our theoretical analysis, which holds for IPM, to $\mathsf{f}-$divergence. Given f a function defined on $\mathbb{R}^+$, continuous and convex, the $\mathsf{f}-$divergence between two distributions p and q: $\mathbb{E}_p[\mathsf{f}(p/q)]$, is null if and only if $p = q$. Interestingly, $\mathsf{f}-$divergence admits a 'IPM style' expression $\mathbb{E}_p[\mathsf{f}(p/q)] = \sup_f \mathbb{E}_p[f] - \mathbb{E}_q[\mathsf{f}^\star(f)]$ where $\mathsf{f}^\star$ is the convex conjugate of f. It is worth noting it is not a IPM expression since the critic is composed by $\mathsf{f}^\star$ in the right expectation. The domain adversarial loss [13] is a particular instance of $\mathsf{f}-$divergence (see [7] for a complete description in the context of generative modelling). Then, we informally transports our analysis on IPM distance to domain adversarial loss. More precisely, we define:

$$\text{INV}_{\text{adv}}(w, \varphi) := \log(2) - \sup_{d \in \mathcal{D}} \mathbb{E}_S[w(Z) \log(d(Z))] + \mathbb{E}_T[\log(1 - d(Z))] \tag{53}$$

$$\text{TSF}_{\text{adv}}(w, \varphi) := \log(2) - \sup_{\mathbf{d} \in \mathcal{D}_\mathcal{Y}} \mathbb{E}_S[w(Z) Y \cdot \log(\mathbf{d}(Z))] + \mathbb{E}_T[Y \cdot \log(1 - \mathbf{d}(Z))] \tag{54}$$

where $\mathcal{D}$ is the well-established domain discriminator from $\mathcal{Z}$ to $[0, 1]$, and $\mathcal{D}_\mathcal{Y}$ is the set of *label domain discriminator* from $\mathcal{Z}$ to $[0, 1]^C$.

C.2 Controlling invariance error with relaxed weights

In this section, we show that even if representations are not learned in order to achieve domain invariance, the design of weights allows to control the invariance error during learning. More precisely $w^\star(\varphi) = \arg\min_w \mathrm{INV}(w, \varphi)$ has a closed form when given a domain discriminator d *i.e.* the following function from the representation space $\mathcal{Z}$ to $[0, 1]$:

$$d(z) := \frac{p_S(z)}{p_S(z) + p_T(z)} \tag{55}$$

Here, setting $w^\star(z) := (1 - d(z))/d(z) = p_T(z)/p_S(z)$ leads to $w(z)p_S(z) = p_T(z)$ and finally $\mathrm{INV}(w^\star(\varphi), \varphi) = 0$. At early stage of learning, the domain discriminator d has a weak predictive power to discriminate domains. Using exactly the closed form $w^\star(z)$ may degrade the estimation of the transferability error. Then, we suggest to build relaxed weights $\tilde{w}_d$ which are pushed to $w^\star$ during training. This is done using temperature relaxation in the sigmoid output of the domain discriminator:

$$w_d^\tau(z) := \frac{1 - \sigma\left(\tilde{d}(z)/\tau\right)}{\sigma\left(\tilde{d}(z)/\tau\right)} \tag{56}$$

where $d(z) = \sigma(\tilde{d}(z))$; when $\tau \to 1$, $w_d(z, \tau) \to w^\star(z)$.

C.3 Ablation study of the weight relaxation parameter α

α is the rate of convergence of relaxed weights to optimal weights. We investigate its role on the task U→M. Increasing α degrades adaptation, excepts in the harder case ($5\% \times [0 \sim 5]$). Weighting early during training degrades representations alignment. Conversely, in the case $5\% \times [0 \sim 5]$, weights need to be introduced early to not learn a wrong alignment. In practice $\alpha = 5$ works well (except for $5\% \times [0 \sim 5]$ in **Digits**).

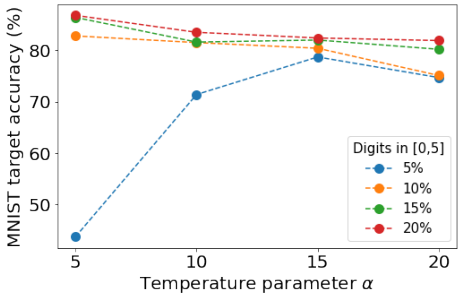


Fig. 3. Effect of α.

C.4 Additional results on Office-Home dataset

Table 3. Accuracy (%) on **Office-Home** based on ReseNet-50.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDAN+E	50.7	**70.6**	**76.0**	**57.6**	**70.0**	**70.0**	**57.4**	**50.9**	**77.3**	**70.9**	56.7	**81.6**	**65.8**
RUDA	**52.0**	67.1	74.4	56.8	69.5	69.8	57.3	**50.9**	77.2	70.5	**57.1**	81.2	64.9

C.5 Detailed procedure

The code is available at https://github.com/vbouvier/ruda.

Algorithm 1 Procedure for Robust Unsupervised Domain Adaptation

Input: Source samples $(x_{S,i}, y_{S,i})_i$, Target samples $(x_{T,i}, y_{T,i})_i$, $(\tau_t)_t$ such that $\tau_t \to 1$, learning rates $(\eta_t)_t$, trade-off $(\alpha_t)_t$ such that $\alpha_t \to 1$, batch-size b

$\theta_g, \theta_\varphi, \theta_d, \theta_{\mathbf{d}}$ random initialization.
$t \leftarrow 0$
while stopping criterion **do**
  $\mathcal{B}_S \sim (x_i^s)$, $\mathcal{B}_T \sim (x_j^t)$ of size b.
  $\theta_d \leftarrow \theta_d - \eta_t \nabla_{\theta_d} \mathcal{L}_{\text{INV}}(\theta_d | \theta_\varphi; \mathcal{B}_S, \mathcal{B}_T)$
  $\theta_{\mathbf{d}} \leftarrow \theta_{\mathbf{d}} - \eta_t \nabla_{\theta_{\mathbf{d}}} \mathcal{L}_{\text{TSF}}(\theta_g, \theta_\varphi, \theta_{\mathbf{d}} | \theta_d, \tau_t)$
  $\theta_\varphi \leftarrow \theta_\varphi - \eta_t \nabla_{\theta_\varphi} \left(\mathcal{L}_c(\theta_g, \theta_\varphi | \theta_d, \tau_t) - \alpha_t \mathcal{L}_{\text{TSF}}(\theta_\varphi, \theta_{\mathbf{d}} | \theta_g, \theta_d, \tau_t) \right)$
  $\theta_g \leftarrow \theta_g - \eta_t \nabla_{\theta_g} \mathcal{L}_c(\theta_g, \theta_\varphi | \theta_d, \tau_t)$
  $t \leftarrow t + 1$
end while