Towards Adaptive Learning with Invariant Representations

Victor Bouvier

université PARIS-SACLAY

Towards Adaptive Learning with Invariant Representations

Vers l'apprentissage adaptatif à l'aide de représentations invariantes

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573 : interfaces : matériaux, systèmes, usages (INTERFACES)
Spécialité de doctorat : INFORMATIQUE
Graduate School : Sciences de l'ingénierie et des systèmes
Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche MICS (Université Paris-Saclay, CentraleSupélec), sous la direction de Céline Hudelot, Professeure des Université, et le co-encadrement de Philippe Very, Ingénieur.

Thèse soutenue à Paris-Saclay, le 13 Décembre 2021, par

Victor Bouvier

Composition du jury

Pablo Piantanida Professeur des Universités, Université Paris-Saclay	Président
Amaury Habrard Professeur des Universités, Université Jean Monnet de Saint-Étienne	Rapporteur & Examinateur
José Hernández-Orallo Professeur des Universités, Université Polytechnique de Valence (Espagne)	Rapporteur & Examinateur
Élisa Fromont Professeure des Universités, Université Rennes 1	Examinatrice
Aurélien Bellet Chargé de recherche, Inria	Examinateur
Céline Hudelot Professeure des Universités, Université Paris-Saclay	Directrice de thèse

THESE DE DOCTORAT
NNT : 2021UPAST141

université PARIS-SACLAY | ÉCOLE DOCTORALE Interfaces: matériaux, systèmes, usages

Titre : Vers l'apprentissage adaptatif à l'aide de représentations invariantes
Mots clés : Apprentissage automatique, Adaptation, Représentations Invariantes, Échantillonnage d'importance, Apprentissage avec peu d'exemples.

Résumé : Bien que l'apprentissage à partir de données (apprentissage automatique) ait considérablement amélioré les systèmes d'Intelligence Artificielle, ces algorithmes sont sensibles aux changements de distribution des données, une situation omniprésente dans l'industrie. L'adaptation des modèles d'apprentissage automatique a fait l'objet de recherches fructueuses, avec une ligne d'étude influente qui apprend des représentations invariantes, c'est-à-dire insensibles aux changements de distributions dans les données. Cette thèse montre que l'apprentissage de représentations invariantes expose au risque de détruire leur adaptabilité, une quantité que nous ne pouvons malheureusement pas contrôler. Nous proposons une analyse théorique introduisant un nouveau terme d'erreur, appelé erreur de réduction de classe d'hypothèse, qui capture l'adaptabilité d'une représentation. Deuxièmement, cette thèse unifie deux domaines de recherche sur l'adaptation, l'échantillonnage d'importance et les représentations invariantes, dans un même cadre théorique. En particulier, nous montrons la nécessité d'un biais inductif pour l'apprentissage adaptatif, replaçant l'expertise humaine au centre de l'apprentissage automatique. Enfin, nous remettons en question une hypothèse fondamentale lors de l'apprentissage de représentations invariantes : l'accès à un grand échantillon de données non étiquetées de la nouvelle distribution. En effet, cette hypothèse est rarement rencontrée en pratique, où l'on souhaiterait idéalement s'adapter avec quelques exemples. Cette thèse contribue à ce nouveau problème en le formalisant et en fournissant à la communauté une base de code pour une recherche reproductible. De plus, nous proposons une référence solide basée sur du Transport Optimal pour cette tâche.

Title : Towards Adaptive Learning with Invariant Representations
Keywords : Machine Learning, Adaptation, Invariant Representations, Importance Sampling, Few-Shot Learning.

Abstract :

Although learning from data (Machine Learning) has dramatically improved Artificial Intelligence systems, these algorithms are not infallible ; they are sensitive to data shift, a ubiquitous situation in the industry. The Adaptation of machine learning models has been the subject of fruitful research, with an influential line of study that learns Invariant Representations, i.e. insensitive to changes in data. In this thesis, we show that learning invariant representations exposes to the risk of destroying their adaptability, a quantity that we, unfortunately, cannot control. We propose a theoretical analysis introducing a new error term, called hypothesis class reduction error, which captures the adaptability of a representation. Secondly, this thesis unifies two research fields for Adaptation, Importance Sampling and Invariant Representations, under the same theoretical framework. In particular, we show the need for inductive bias for adaptive learning, putting human expertise back at the centre of Machine Learning. Finally, we question a fundamental assumption when learning invariant representation ; the access to a large sample of unlabeled data of the new distribution. Indeed, this assumption is rarely met in practice, where we would ideally like to adapt with a few examples. This thesis contributes to this new problem by formalizing it and providing the community with a codebase for a reproducible search. Moreover, we offer a solid baseline based on Optimal Transport for this task.

Acknowledgements

I want to thank Professor José Hernández Orallo and Professor Amaury Habrard for evaluating my work, and Élisa Fromont and Aurélien Bellet for your presence during the defence as examiners. Your remarks and our scientific exchanges during the defence were precious to improving the manuscript. I also wish to thank warmly Professor Pablo Piantanida for chairing my defence, making it a memorable moment for me.

This adventure began in May 2017 with an internship at Sidetrade. As an unbearable intern, I was lucky enough to meet colleagues who helped me grow as a person. More importantly, they became friends. Thank you, Philippe, for these rich scientific exchanges, for always supporting my idea, even if many of them were entirely unreasonable. I still remember dissecting with you Ben-David's theory in the 5th arrondissement with a Greek collation. Thank you, Clément Chastagnol and Clément Barbier, who agreed to take over the torch! Thank you, Jean-Cyril Schutterlé, for supporting this thesis project at Sidetrade. Our long discussions were a genuine breath of fresh air. Thanks to all the Sidetraders for their welcome. Finally, I am also immensely grateful to Sidetrade for their trust.

Thank you, Céline, for having welcomed me into your research team. You always found the words to encourage me when I could no longer bounce back in the tough moments. I transformed the first reviews' uppercuts into noteworthy successes with your guidance. You used to say, "it's *your* research project". This terrifying perspective allowed me to develop personal and independent scientific thinking, which is infinitely more precious than a few papers. Thank you, Myriam, for bringing new blood into the team! Your presence was decisive for my progress. Thank you Etienne, Victor, Thomas, Yassine and Jhun. We had the opportunity to work together; it was an extraordinary experience for me.

I want to thank my family that always supported me. My loving thanks comes to Ève. Thank you for your comfort and your understanding. I know you were looking forward to this defence; *we* did it!

Contents

List of Symbols

Generality

$\mathbb{R}$	Real numbers
$\mathbb{N}$	Natural numbers ($\mathbb{N}^\star$ non-null natural numbers)
$\mathbb{R}^n$	Real vectorial space of dimension $n \in \mathbb{N}$
$\cdot$	Scalar product, $x, y \in \mathbb{R}^n, x \cdot y = \sum_{i=1}^n x_i y_i$
$\oplus$	Concatenation $x \in \mathbb{R}^n, y \in \mathbb{R}^{n'}, z = x \oplus y \in \mathbb{R}^{n+n'}$ where $z_i = x_i$ for $i \leq n$ and $z_i = y_i$ for $i > n$.
$\odot$	Point-wise multiplication of vectors, $x, y \in \mathbb{R}^n, z = x \odot y \in \mathbb{R}^n$ where $z_i = x_i y_i$ for $i \in \{1, \cdots, n\}$
$\otimes$	Multilinear map of vectors, $x \in \mathbb{R}^n; y \in \mathbb{R}^{n'}, z = x \otimes y \in \mathbb{R}^{n \times n'}$ where $z_{i \times j} = x_i y_j$ for $i \in \{1, \cdots, n\}, j \in \{1, \cdots, n'\}$

Probability

IID	Independently and Identically Distributed
$\mathcal{X}$	We note a set with cursive letter
X	Random Variable
x	Realization of X
$\mathbb{P}(X)$	Probability of X
$p(X)$	Probability distribution of X
$\mathcal{P}(\mathcal{X})$	Set of probability distributions of $\mathcal{X}$
	We note $X \sim p$ when X is distributed with respect to p.
$\mathbb{E}_p[X]$	Expectation of $X \sim p$
$\mathbb{V}_p[X]$	Variance of $X \sim p$
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2

Statistical Learning

$\mathcal{X}$	Input space
$\mathcal{Y}$	Label space
$\mathcal{Z}$	Representation space, typically $\mathcal{Z} = \mathbb{R}^m$
C	Number of classes, *i.e.* $C = \lvert\mathcal{Y}\rvert$
p_D	for $D \in \{S, T\}$
	respectively the source or target distribution
C	Number of classes, *i.e.* $C = \lvert\mathcal{Y}\rvert$
$\mathcal{H}$	Set of hypothesis (Hypothesis class),
	i.e. a subset of measurable functions from $\mathcal{X}$ to $\mathcal{Y}$
Φ	Set of representations (Representation class)
	i.e. a subset of measurable functions from $\mathcal{X}$ to $\mathcal{Z}$
$\mathcal{G}$	Set of classifiers (Classifier class)
	i.e. a subset of measurable functions from $\mathcal{Z}$ to $\mathcal{Y}$
	Deep networks are noted $\mathcal{H} := \mathcal{G}\Phi := \{g \circ \varphi, g \in \mathcal{G}, \varphi \in \Phi\}$

$\mathcal{G}_\text{p}$	Set of probability classifiers (Probability classifier class) *i.e.* a subset of measurable functions from $\mathcal{Z}$ to $\mathcal{P}(\mathcal{Y})$
$\mathcal{F}$	Set of measurable functions from $\mathcal{Z}$ to $[-1,1]$
$\mathcal{F}_C$	Set of measurable functions from $\mathcal{Z}$ to $[-1,1]^C$
$\mathcal{D}$	Set of discriminators *i.e.* a subset of measurable functions from $\mathcal{Z}$ to $[0,1]$ Note that it differs with source and target data $\mathcal{D}_S$ and $\mathcal{D}_T$, respectively In Chapters 2 and 6, a dataset is noted $\mathcal{D}$, the context solving the ambiguity
$\mathcal{D}_C$	Set of class-level discriminators *i.e.* a subset of measurable functions from $\mathcal{Z}$ to $[0,1]^C$
$Z = \varphi(X)$	Z is the representation X for $\varphi \in \Phi$, We refer to Z and φ as representation or representations indifferently
$\hat{Y} = g(Z)$	$\hat{Y}$ is the predicted label from Z for $g \in \mathcal{G}$
$\mathbf{u}$	For $u \in \{1, \cdots, C\}$, $\mathbf{u} \in \mathbb{R}^C$ where $\mathbf{u}_c = 1$ if $u = c, 0$ otherwise For $g \in \mathcal{G}, z \in \mathcal{Z}$, we note $\mathbf{u} = \mathbf{g}(z)$ where $u = g(z)$ The notation does not apply for $d \in \mathcal{D}$ and $f \in \mathcal{F}$ to avoid confusion with elements of $\mathcal{D}_C$ and $\mathcal{F}_C$, respectively

Losses

L_c	Cross-entropy loss $L_c(g\varphi) = \mathbb{E}_p[-\mathbf{Y} \cdot \log(g(Z))], g \in \mathcal{G}_\text{p}, \varphi \in \Phi$
H	Entropy loss $H(g\varphi) = \mathbb{E}_p[-g(Z) \cdot \log(g(Z))], g \in \mathcal{G}_\text{p}, \varphi \in \Phi$
L_INV	Invariance loss $L_\text{INV}(w, \varphi) = \mathbb{E}_S[-w(Z) \log d(Z)] + \mathbb{E}_T[-\log(1 - d(Z))]$ Corresponds to $L_\text{dis}(\varphi, d)$ when $w = 1$ $\mathrm{d} \in \mathcal{D}; \varphi \in \Phi$
L_TSF	Transferability loss $L_\text{TSF}(w, \varphi, g) = \mathbb{E}_S[-w(Z)g(Z) \cdot \log \mathbf{d}(Z)] + \mathbb{E}_T[-g(Z) \cdot \log(1 - \mathbf{d}(Z))]$ $\mathbf{d} \in \mathcal{D}_C, g \in \mathcal{G}_\text{p}, \varphi \in \Phi$

List of Abbreviations

ML	**M**achine **L**earning
DL	**D**eep **L**earning
CV	**C**omputer **V**ision
NLP	**N**atural **L**anguage **P**rocessing
ERM	**E**mpirical **R**isk **M**inimization
SRM	**S**tructural **R**isk **M**inimization
IRM	**I**nvariant **R**isk **M**inimization (Arjovsky et al. 2019)
OOD	**O**ut-**o**f-**D**istribution (Generalization)
SSL	**S**emi-**S**upervised **L**earning (Chapelle, Scholkopf, and Zien 2009)
UDA	**U**nsupervised **D**omain **A**daptation
	(Quinonero-Candela et al. 2009; Pan and Yang 2009; Redko et al. 2019)
SSDA	**S**emi **S**upervised **D**omain **A**daptation
ADA	**A**ctive **D**omain **A**daptation
IPM	**I**ntegral **P**robability **M**easure
GRL	**G**radient **R**eversal **L**eversal (Ganin and Lempitsky 2015)
BSP	**B**atch **S**pectral **P**enalization (Chen et al. 2019c)
IB	**I**nductive **B**ias
AL	**A**ctive **L**earning (Settles 2009)

1 Introduction

1.1 Opening words

Although there is not strictly speaking a consensual definition of what *intelligence* refers to (Russell and Norvig 2020), neither an established protocol to evaluate it (Hernández-Orallo 2017), it is accepted that *Artificial Intelligence* (AI) characterizes the ability of a system to solve a task which is the prerogative of humans. AI research has made unprecedented progress building systems capable of achieving, and even sometimes surpassing, human performance in some cognitive tasks, *e.g.* visual recognition tasks by detecting skin cancer (Esteva et al. 2017), automatic translation (Vaswani et al. 2017) or games such as Chess (Silver et al. 2017) or Go (Silver et al. 2016). Even problems thought to be unsolvable by machines, or solved in the far future, are the playground of this discipline where the breakthrough of protein folding is a striking example (Sample 2020).

This scientific and technological *tour de force* has therefore attracted the attention of many industries to automate costly tasks that were performed by humans even with a high level of expertise, *e.g.* the autonomous vehicles industry (Chabot et al. 2017) or decision support with medical imaging (Litjens et al. 2017). AI also brings excellent added value, especially for automating low-value, repetitive and time-consuming tasks, *e.g.* email routing, document reading (ID card, check, video analysis). Given the important economical effect AI can have, many countries, including France (Villani et al. 2018), have put Artificial Intelligence at the heart of their innovation strategy.

The shift from *Symbolic AI*, where humans encode pieces of knowledge through formal symbols to enable computational reasoning (McCarthy 1959), to *Machine Learning* (ML) that aims to replicate some of the mechanisms of human intelligence by learning from data, has undoubtedly contributed to recent breakthroughs in AI. The last ten years have been marked by a greater availability of training data and computational resources that enabled progress in *Deep Learning* (LeCun, Bengio, and Hinton 2015), scaling ML systems to an unequalled complexity. However, is "more data for more complex models" enough for pursuing what seems like unbridled progress?

The progress of Machine Learning, mainly through the increasingly prominent role of Deep Learning, is not without its weakness. By learning from data, ML systems inherit its property, which is not being as representative of the world as we think the data is (Torralba, Efros, et al. 2011; Amodei et al. 2016; Beery, Van Horn, and Perona 2018; Arjovsky et al. 2019; Marcus 2020). For instance, the fact that data used to train a recognition system is represented only with cows on pasture makes the system unable to identify a cow on a beach (Beery, Van Horn, and Perona 2018). During this dissertation, we will follow the denomination adopted in the literature of adaptation (Quinonero-Candela et al. 2009; Pan and Yang 2009); training data will be referred

to as *source data*, and real-world data will be referred to as *target data*. We refer to the situation when source data used for training an ML system, *e.g.* cows on pasture, differs from the target data, *e.g.* cows may exist in any environment, as *data shift* or *distribution shift* (Quinonero-Candela et al. 2009) and is ubiquitous in industrial applications (Amodei et al. 2016). Such a failure is not an isolated case. Unfortunately, ML systems are highly sensitive to data shift, *i.e.* performances they can achieve during learning may be very different from their actual performance once deployed, making the safe deployment of ML systems very challenging, particularly in critical applications.

A strategy to address data shift aims to *adapt* the model observing data from the real world, opening the fruitful field of *Unsupervised Domain Adaptation* (UDA) (Quinonero-Candela et al. 2009; Pan and Yang 2009). The paradigm of UDA builds upon the following assumption: in addition to source, we assume that the system *knows* some data on which it will be deployed, allowing it to adapt itself to fit the real world better. Following the example of cows from (Beery, Van Horn, and Perona 2018), the system learns the concept of a cow based on two examples; the former on pasture, the latter on a beach. The expert indicates the former example is a cow (on pasture) without providing such information for the latter (on a beach). It reflects the situation where a learner learns specific (or biased) concepts and has to generalize them by confronting the real world (adaptation).

1.2 Problematic & Goals

From the anthropomorphic point of view, the concept of *adaptation* is central for developing intelligence. Indeed, the pioneering work of Jean Piaget ($1896 - 1980$), an influential biologist, logician, psychologist, and epistemologist, that has studied intelligence development during childhood highlights that reasoning consists in articulating abstract and generic representations. Importantly, the acquisition of these representations is not static. Instead, it is refined and made more complex by the continual interaction with the environment through the process of *adaptation* (Piaget 1936).

The implementation of Jean Piaget's idea of adaptation into ML systems is still in its infancy, promising exciting future research. In particular, thanks to the development of Deep Learning that learns abstract and meaningful representations of the data (Bengio et al. 2009; LeCun, Bengio, and Hinton 2015), an influential line of study aims to learn *invariant* representations of the data (Ben-David et al. 2007; Ganin and Lempitsky 2015; Arjovsky et al. 2019). Such a paradigm extracts from the data invariant statistical patterns, *i.e.* shared statistical patterns between the source and the target data. Following the example of the cow from (Beery, Van Horn, and Perona 2018), a representation of the cow should remain invariant to the observed background, *e.g.* invariant if the cow appears on a pasture or a beach.

Although there is a mature theory of UDA (Ben-David et al. 2010a), accompanied by evidence of the power of deep learning of invariant representations (Ganin and Lempitsky 2015), we still have a poor understanding of why and when invariant based adaptation will work. First, invariance is not the *panacea* for adaptation. Surprisingly, if it is over-applied (Zhao et al. 2019), it conflicts with the primary objective of learning transferable concepts from training data to the real world. We will call this trade-off the problem of **transferability of domain invariant representations**, a question that will be at the heart of our reflections. Second, invariance is often

quantified as comparing statistics between training and the real world data (Ganin and Lempitsky 2015). Computing such reliable statistics deserves a large number of data, raising an exciting question: can we assume that adaptation requires a complete view of the real world?

The goal of that thesis is to identify the *genuine ingredients* that makes invariant representations a powerful tool for adaptation. In particular, this thesis investigates;

1. the role of expert assumptions enforced into the models (**inductive bias**), putting the human expert back at the centre of learning, with both a theoretical (Part II) and empirical emphasis (Part III).
2. the need to have a complete view of the real world to perform adaptation (Part IV). One can readily argue that adaptation shall operate even when few real world data is available, as a child would do. Thus, we are still far from developing learning systems that adapt, in the idea of Jean Piaget has of adaptation, through the unsupervised observation of the real world.

1.3 Contributions

What place for Adaptation in Intelligence? Our first contribution consists in conducting a reflection on the place that adaptation occupies in the development of intelligence, and by extension, in the development of AI. We return to the founding work of Jean Piaget (Piaget 1936), who describes the development of intelligence as an interaction of a system with its environment to extract more and more general concepts. We claim that how a system reacts to *novelty* is a core component of what could be described as *intelligent*. We put this anthropomorphic analysis into perspective with recent developments in Machine Learning. First, we show that characterizing novelty is an underlying issue of this scientific field, mainly to evaluate what an ML system can do. For example, the overfitting problem, *i.e.* a learning system that retains the training data by heart, and the lack of robustness to data shift, are instantiations of quantifying how an ML system reacts to *novelty*. Second, we provide an industrial flavour of the concept of adaptation through the emerging topic of Machine Learning Operations. In particular, it clarifies the positioning of some active research fields and shows how they implement, complementary, the principle of adaptation as described by Jean Piaget in a complementary way.

Hypothesis Class Reduction. Our second contribution revisits the celebrated theory of learning from different domains (Ben-David et al. 2010a), the building block of learning domain invariant representations for adaptation (Ganin and Lempitsky 2015). The theory breaks the problem of adaptation into three components; achieving a low error on the source data (source error (1)), building a class of models that are insensitive to shift from the source to the target data (distribution discrepancy (2)), and guaranteeing that an ideal model achieves a low error in both domains (adaptability (3)). Prior works focus on addressing the two first components while assuming adaptability has a negligible role during adaptation. Such argument mainly relies on the fact that adaptability can not be computed in a scenario of Unsupervised Domain Adaptation (UDA) (Quinonero-Candela et al. 2009; Pan and Yang 2009) where target labels are absent. However, recent work shows that this assumption is incorrect and eludes the fundamental element of learning invariant representations for adaptation, guaranteeing the transferability of domain invariant representations (Zhao et al. 2019; Johansson, Sontag, and Ranganath 2019). We

elaborate on this theory to highlight a fourth error term named *Hypothesis Class Reduction* (HCR) error. We interpret it as the risk of deleting relevant information in the representations to achieve invariance. We show that the dynamics of the HCR error is directly related to the adaptability term, that is out of reach in UDA. Finally, this analysis allows us to theoretically justify a well-adopted heuristic to improve the transferability of invariant representations (Chen et al. 2019c).

Representations, Weights and Inductive Bias. Our third contribution attempts to unify two complementary approaches to adaptation, namely Importance Sampling (Quinonero-Candela et al. 2009) and Learning Invariant Representations (Ganin and Lempitsky 2015; Long et al. 2018). The former focuses on finding the importance to give to samples in the source domain to represent the target domain better. The latter seeks to identify stable statistical patterns across distributions by learning an invariant representation. We recall that neither of them can solve the adaptation problem in its entirety. For example, importance sampling is bound to fail when the source and target data do not overlap (Johansson, Sontag, and Ranganath 2019). Learning invariant representations cannot succeed if the distribution of labels is different between the domains (Zhao et al. 2019). For this reason, we conduct a theoretical analysis that brings these two approaches together. In particular, we relate the error in the target domain to three components: the source error (1), the invariance of representations (*invariance* term (2)), and a new term that we identify as the transferability of the representations (*transferability* term (3)). We show that weighting the source domain, as described in the importance sampling literature (Quinonero-Candela et al. 2009), can control invariance under certain assumptions. Furthermore, if invariance of representations is achieved, the transferability term is null if the process that gives labels given the representation is conserved across domains. Unfortunately, computing the transferability term requires knowledge of target labels, that is unavailable in UDA. Therefore, computing this term is the remaining difficulty for improving transferability of domain invariant representations. To this purpose, we develop an analysis of the role of **inductive bias**, *i.e.* the set of assumptions enforced into a learner to perform adaptation. In particular, we show theoretically that one can obtain an approximation of the transferability term in presence of a *strong* inductive bias. This result elucidates a common knowledge theoretically: adaptation requires expertise and can not result only from the data.

From Theory to Practice. Our fourth contribution puts these theoretical results into applications.

1. We show that our analysis, that relates invariance and transferability through a unification effort of weights and representations, allows us to build an efficient adaptation algorithm, called *Robust Unsupervised Domain Adaptation* (RUDA), in the challenging scenario where the distribution of labels shifts across domains (Zhao et al. 2019). The algorithm relies on two main ingredients. First, given a representation, we design weights to minimize the invariance term. Second, given weights, we learn the representation to minimize the transferability term. As mentioned above, transferability involves target labels and is not tractable. To circumvent this issue, we replace target labels with the model's prediction in the target domain, a strategy that we refer to as a *weak* inductive bias, following our analysis of inductive bias in adaptation.

2. We show that invariance comes at the expense of robustness in the target domain. To address this weakness, we return to techniques well adopted in the *Semi-Supervised Learning* (SSL) community (Chapelle, Scholkopf, and Zien 2009), namely the cluster assumption, which states that *"If points are in the same cluster, they are likely to be of the same class"*. When enforcing this assumption during invariant representation learning, we show that we achieve state-of-the-art performance compared to other methods. We theoretically interpret this empirical success as an implementation of a strong inductive bias, which, as shown above, allows us to obtain a better approximation of the transferability of the representations.

3. We develop a criterion to quantify the lack of transferability of a sample in the target domain. We express the criterion as the norm of a vector, which we call *Stochastic Adversarial Gradient Embedding* (Sage). In particular, we use this vector to identify the samples for which the model will likely fail. Then, we send such target samples to an expert (Oracle) for annotation. We thus fall into the paradigm of *Active Domain Adaptation* (ADA) and show that this criterion significantly improves ADA performance. From another point of view, this also indicates that a small amount of target annotated data is sufficient to improve the adaptation performance drastically. This is an exciting result for a practitioner who can benefit considerably from a small annotation effort when the application allows it.

Bridging Adaptation and Few-Shot Learning. Our final contribution challenges common assumptions in Unsupervised Domain Adaptation, namely that a large number of data populates both the source and the target domains. We argue that a potentially high impact research line is building algorithms that can adapt with a few examples. In particular, this raises an exciting question: how to achieve invariance when this principle relies on comparing statistics that require a large number of samples to be evaluated reliably? To this purpose, we bridge the gap between adaptation and Few-Shot Learning (FSL). We introduce the novel problem of Few-Shot Learning under Support/Query Shift (FSQS) where the support set, *i.e.* labelled samples, and the query set, *i.e.* unlabelled samples, are sampled from different distributions, *i.e.* the source and the target distributions, respectively. We develop a benchmark on that problem as well as strong baselines. We hope this novel and challenging problem will attract the attention of the community towards building adaptive models.

1.4 Overview of the thesis

The present doctoral thesis contributes to the long-term objective of *learning adaptive models with invariant representations*. Notably, we study the fundamental trade-off between invariance and transferability of representations. Moreover, we open the problem of adapting with few samples that we identify as major for future research in Machine Learning. The dissertation is organized as follows.

I. Part I is intended to a broad audience that wants to discover the principle of adaptation. In particular, we show in Chapter 2 that the need of adaptive models raises naturally from the current state-of-the-art in Machine Learning. We

provide two complementary perspectives. First, we follow an anthropomorphic description of adaptation establishing fruitful connections between developing intelligence and Machine Learning. Second, we show that adaptation also responds to an industrial need that has emerged in recent months under the name of *Machine Learning Operations* (MLOps). Chapter 3 provides the necessary technical background to address the problem of learning adaptive models. Importantly, we characterize the trade-off of learning transferable invariant representations and review some pioneering works that address it.

II. Part II is a theoretical investigation of the fundamental trade-off of learning transferable invariant representations. Chapter 4 revisits the seminal theory of learning from different domains (Ben-David et al. 2010a), introducing a new error term that quantifies the risk of deleting relevant information in the data to achieve invariance. Importantly, this new error term provides the needed theoretical ground of a well-adopted heuristic to learn domain invariant representations (Chen et al. 2019c). Chapter 5 unifies two important lines of study for UDA: importance sampling (Quinonero-Candela et al. 2009) and learning domain invariant representations (Ganin and Lempitsky 2015). In particular, we provide a new theoretical analysis that clarifies the role of invariance and transferability for adaptation. Crucially, we show that invariance is out of reach when target labelled data is unvailable. Such negative result motivates us to study the role of *Inductive Bias*, *i.e.* the set of assumptions enforced into the system, proving that one can obtain a reliable approximation of the transferability when provided with a *strong* inductive bias.

III. Part III focuses on implementing adaptation in light of the need of inductive bias. Chapter 7 studies the problem of label shift known to hurt transferability of domain invariant representations (Zhao et al. 2019). In particular, we derive from the theoretical analysis of Chapter 5 an efficient algorithm that improves adaptation in the challenging scenario of label shift. Chapter 8 studies the effect of enforcing a strong inductive bias during adaptation. We draw inspiration from the *cluster assumption* in *Semi-Supervised Learning* (Chapelle, Scholkopf, and Zien 2009) to promote classifier that provides consistent prediction when perturbing inputs. As most applications may not benefit from such inductive bias, Chapter 9 establishes connections with *Active Learning* (AL) to improve the transferability of domain invariant representations. We develop a criterion that quantifies the lack of transferability and an Oracle annotates such samples accordingly.

IV. Part IV elaborates around the emerging problem of adapting with few samples. Chapter 10 bridges the gap between adaptation and few-shot learning. We specify a new learning task providing the needed benchmark and first baselines to this novel and challenging problem. Chapter 11 concludes this thesis and describes exciting future research directions for which this dissertation may provide some foundations.

1.5 Publications

The present dissertation is based on the following published works listed by chronological order;

1. A technical report (Bouvier et al. 2020a);

Domain-Invariant Representations: A Look on Compression and Weights,
Victor Bouvier, Céline Hudelot, Clément Chastagnol, Philippe Very and Myriam Tami,
Technical report, https://openreview.net/forum?id=B1xGxgSYvH

This report has been originally submitted to ICLR 2019. The Chapter 4 provides a more mature version of this preliminary work.

2. An international publication (Bouvier et al. 2020b);

   Robust Domain Adaptation: Representations, Weights and Inductive Bias
   Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami and Céline Hudelot,
   European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Ghent (Belgium), Online, 2020.

   This publication has received the best (student) Machine Learning paper award. (Bouvier et al. 2020b) has also been presented at the:

   - Best Paper Sister Conference hosted by the International Joint Conference of Artificial Intelligence, Montréal (Canada), Online, 2021 (Bouvier et al. 2021).
   - Conférence sur l'apprentissage Automatique (CAp), Saint-Étienne, 2021, as an oral presentation.

   The Chapters 5, 6 and 7 render this publication.

3. A national publication (Ouali et al. 2020);

   Target Consistency for Domain Adaptation: when Robustness meets Transferability,
   Yassine Ouali, Victor Bouvier, Myriam Tami, Céline Hudelot,
   Conférence sur l'apprentissage Automatique (CAp), Saint-Étienne, 2021.

   Yassine Ouali and Victor Bouvier contributed equally. The Chapter 8 renders this publication.

4. An oral presentation in a workshop hosted by an international conference (Bouvier et al. 2020c):

   Stochastic Adversarial Gradient Embedding for Active Domain Adaptation,
   Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami and Céline Hudelot,
   Interactive Adaptive Learning workshop (IAL),
   Colocated with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Bilbao (Basque Country), Online, 2021.

   The chapter 9 renders this publication.

5. An international publication (Bennequin et al. 2021);

Bridging Few-Shot Learning and Adaptation: New Challenges of Support-Query Shift,
Étienne Bennequin, Victor Bouvier, Myriam Tami, Antoine Toubhans and Céline Hudelot,
European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Bilbao (Basque Country), Online, 2021.

Étienne Bennequin and Victor Bouvier contributed equally. The chapter 10 renders this publication.

6. (In preparation) Chapter 2 will be adapted as *Brève d'IA* for *Société Française des Statistiques*.

Part I

Adaptation in Machine Learning

2 What place for Adaptation in Intelligence?

Contents

The present chapter intends for a broad audience that wants to discover the principle of adaptation and its place in developing intelligence. With a philosophical flavour, we aim to connect these ideas with research in Machine Learning and the industrial needs raised by the emergence of ML technology.

The present Chapter is organized as follows. First, supported by the pioneering work of Jean Piaget, we establish connections between the self-improvement of a system and the ability to adapt, which is a cornerstone of intelligence (Section 2.1). Second, we relate the concept of adaptation with the skill to react correctly from novel observations (Section 2.2). Understanding and characterizing novelty has been a central question around developing systems that learn from data, referred to as Machine Learning (ML). Crucially, we emphasize that such systems are vulnerable when the training data significantly differs from the real-world data, a situation referred to as data shift. We motivate the use of Invariant Representations, *i.e.* representations that remain robust to undesired data changes. Finally, we clarify and position existing approaches in the literature that implement, in a complementary way, the principle of adaptation as described by Jean Piaget (Section 2.3). Through the lens of the emerging topic of *Machine Learning Operations* (MLOps), we expose the industrial promises of adaptation.

2.1 (Artificial) Intelligence and Adaptation

2.1.1 What is missing for Artificial Intelligence to truly be intelligent?

The pioneering idea that computers will match the remarkable ability of humans dates back to the end of the Second World-War (Turing 1948). John McCarty, one of the founding fathers of *Artificial Intelligence* (AI), was the first to mention the "AI" term at the 1956 Dartmouth Conference to describe such systems. After falling into disuse in the early 2000s, this term is now used massively to characterise very different systems, from expert systems to default prediction, from automatic translation systems to autonomous driving systems. What characteristics do these systems share to claim to be intelligent?

Quite surprisingly, it is still pretty challenging to provide a general definition of AI. In particular, it depends on whether[1] we are interested in building systems which act, *e.g.* perceive objects, or that think, *e.g.* produce a logical system (Russell and Norvig 2020). Let us consider the simple instantiation of an AI through an "input-output" system, *i.e.* given an input, the system has to identify the best output in order to maximize a specified objective[2], *e.g.* maximizing the accuracy of identification in images. For the particular example presented in Figure 2.1, an AI would try to emulate the human visual cortex. Thus, as a first approximation, an AI system tries to automate/emulate a non-trivial task, *i.e.* a task that requires a certain form of intelligence. However, is automation the essence of intelligence? What fundamental component of intelligence is artificial intelligence missing today? One can readily argue that the idea of AI, as McCarthy and his peers formulated, is more ambitious:

> "*Probably a truly intelligent machine will carry out activities which may best be described as self-improvement* [...]."
> (McCarthy, Minsky, and Rochester 1955)

The ability of a system to improve by itself by experiencing with its environment, that one can define as *learning*, is undoubtedly a powerful signal of intelligence. We discuss in the following the role of self-improvement in intelligence through the concept of *Adaptation*.

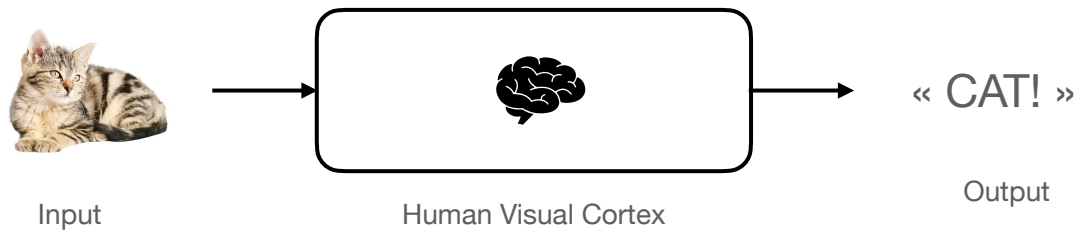


Figure 2.1: Input-Output view of human visual perception. A visual recognition system aims at emulating the performance of humans on that task. Because of its complexity, and as it is a fundamental part of human intelligence, such a system is considered *Artificial Intelligence*.

Figure 2.2: Jean Piaget with two of his children, on whom he studied the development of intelligence and the acquisition of language.

2.1.2 Adaptation as a cornerstone of Intelligence

Intuitively, one can define *adaptability* as the ability to react correctly in a unfamiliar situation, as well as the ability to assimilate new knowledge quickly. Jean Piaget (1896 − 1980), an influential biologist, logician, psychologist, and epistemologist, studied the central role adaptation plays in developing intelligence, particularly during childhood. According to Jean Piaget, adaptation emerges when the balance[3] between an individual and the environment collapses. For instance, balance's collapse occurs when an individual perceives an object that he or she has never seen before. Adaptation aims to re-establish this balance, thus makes it possible to construct increasingly general knowledge. Jean Piaget defines adaptation (Piaget 1936) by the orchestration of two stages, as presented in Figure 2.3;

- *Assimilation*, consists in framing this change, such as a new object that one perceives, to a piece of existing knowledge, referred to as a *psychological schema*.
- *Accommodation*, occurs when assimilation fails. Accommodation modifies an existing psychological schema to integrate this new knowledge.

The pioneering work of Jean Piaget highlights that reasoning consists in articulating abstract and generic representations embodied in psychological schemes (Rosenberg 1980). The acquisition of these representations is not static, as one might read a reference book once and for all to discover a new discipline. Instead, it is refined and made more complex by the continual interaction with the environment through adaptation, as a learner would do when facing a series of application exercises proposed in the reference book. Suppose adaptation is, according to Jean Piaget, the cornerstone of intelligence development, what about its role in developing AI?

[1] According to Russell and Norvig, there exists a second dimension of the definition, depending on the behavior is human-like or "rational". See (Russell and Norvig 2020) for an extended definition of AI.

[2] This simple form of action (output) based on perception (input) is referred to as a *reflex agent* (Russell and Norvig 2020).

[3] Balance refers to the match between individual's mental model, *i.e.* knowledge, and the environment. An analogy with Machine Learning, *i.e.* a data-oriented interpretation, is the concept of *stationarity* of data from the training to real-world.

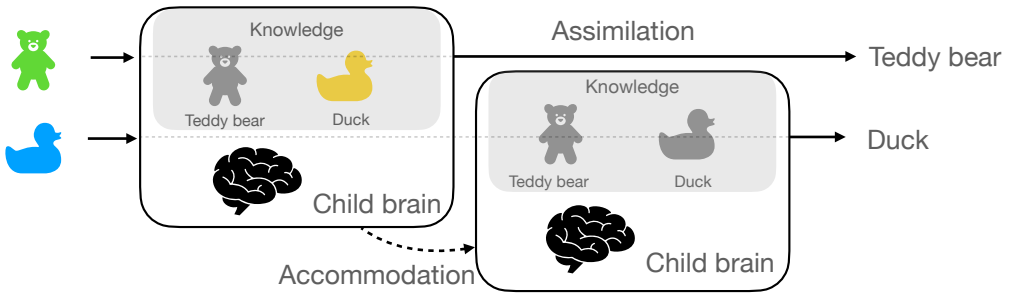


Figure 2.3: Illustration of adaptation according to Jean Piaget. The child has two objects to identify; a green teddy bear and a blue duck. On the one hand, the child knows that colour is not a feature of a teddy bear (existing knowledge of how a teddy bear looks like), then assimilation allows him to infer the former object is a teddy bear. On the other hand, the child has only seen yellow ducks in the past (existing knowledge of how a duck looks like), failing to assimilate the latter object. Accommodation thus begins to work; it modifies knowledge by removing colour as a feature of a duck. One can speculate how to perform accommodation. Either the child's parents explain that the colour does not make the duck (supervised), or the child infers this on his or her own (unsupervised), *e.g.* by deducing that if the colour does not make the teddy bear, this is probably also the case for a duck.

2.1.3 What place for Adaptation in Artificial Intelligence?

A quick tour into AI's history

AI has known a succession of winters and springs corresponding to periods of scientific deception and enthusiasm. Although there are other paths to AI, there are now two relatively established influential and contrasting visions; the *symbolic AI vs data-centric AI* approaches.

Symbolic AI. McCarthy proposes to build an AI by articulating symbols, *i.e.* elementary bricks of knowledge, as presented in Figure 2.4. To this purpose, human encodes pieces of knowledge through symbols and a formalism for logical reasoning (McCarthy 1959). The symbolic AI is thus firmly based on the work of Jean Piaget, where symbols take the role of psychological schemes. However, the symbolic AI approach is incomplete as a model of adaptation, which is a cornerstone of intelligence development. Bridging adaptation to a symbolic AI system is thus a very difficult task since it needs to modify and revise pieces of knowledge. This line of work, known as *Belief Change and Revision* (Alchourrón, Gärdenfors, and Makinson 1985; Aiguier et al. 2018) brings a theoretical perspective of accommodation in

Figure 2.4: Overview of symbolic system for recognition in images. Here, the system benefits from a description logic that enables to infer a cat is in the image based on symbols: "little ears", "cute paws" and "feline face".

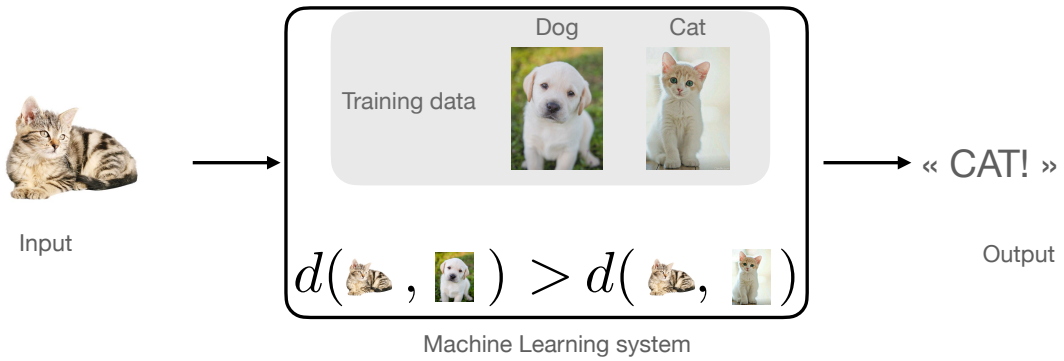


Figure 2.5: Overview of a Machine Learning system for recognition in images. Here, the system benefits training examples of a dog and cat and a distance $d(\cdot,\cdot)$ that allow to compute similarity. Since the distance of the images from the training example of a dog is larger than the distance of the image from the training example of a cat, the system infers a cat is in the image.

symbolic systems. However, it is far from being operational in real world systems. mainly due to their computational complexity and the undecidability.

Data-Centric AI with Machine Learning. Data-centric AI aims to replicate some of the mechanisms of human ability by learning from a finite number of examples, hence its name of *Machine Learning* (ML) (see Figure 2.5 for an illustration). Thus, learning quality depends on both the power of the learning algorithm and the available data. *Supervised Learning* is the most common paradigm. Here, the algorithm emulates the behaviour of a system based on annotated examples. For instance, pattern recognition systems for *Computer Vision* (CV) aim to emulate the human visual cortex based on a dataset of images of animals and objects provided with the information of image content (annotation). *Neural Networks* (NN) have become very popular in the ML community over the last ten years. In Figure 2.6, we depict a neural network, a metaphor of a brain where the ML system is architected as a network of vertices emulating neurons with edges emulating synapses. During learning, NN adjust the interaction strength between neurons to fit the data as well as possible, making learning very similar to the accommodation principle described by Jean Piaget (see Figure 2.6). Data-Centric AI is not without any curse. By learning from data, systems inherit its property, which is not being as representative of the world as we think the data is (Torralba, Efros, et al. 2011; Amodei et al. 2016; Beery, Van Horn, and Perona 2018; Arjovsky et al. 2019; Marcus 2020). In particular, when training data is far from the real-world, that one could identify as balance collapse following Jean Piaget's terminoly, ML systems appear to be inoperative. We will elaborate further this weakness in Section 2.2.3 that will be the main motivation of the present thesis.

Adaptation: Learning new skills or improving an existing one?

Despite their hegemony in the AI field, both the symbolic and data-centric AI do not natively embed the principle of adaptation. In the following, we put a particular focus on ML systems. Once deployed in its environment, a vanilla ML model restricts itself to assimilation (inferring on new data) without adjusting its knowledge by accommodation. Thus, adaptation, as described by Jean Piaget, remains a long-standing challenge in Machine Learning. One can identify two lines of study that

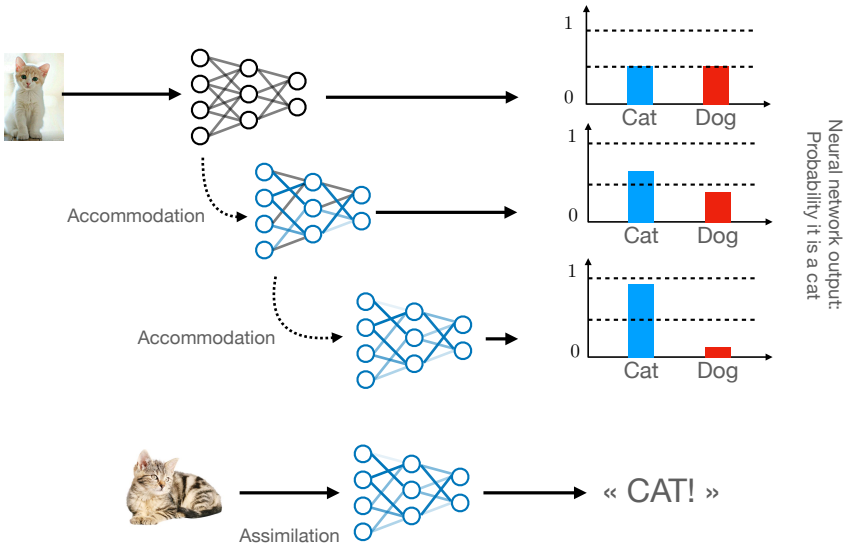


Figure 2.6: Overview of the mechanism of learning in a *Neural Network* (NN), a brain-inspired model for Machine Learning. A NN aims to emulate the neural architecture of the brain by modelling the interactions between neurons. First, given training data (a white cat), the model adapts the interaction strength between neurons in order to detect a cat in the image. This is often modelled as a probability distribution over a set of classes, here *Cat vs Dog*. Second, once the model has exhibited the best neural interaction, it can detect the presence of a cat on new images by the principle of assimilation. Interestingly, the *Deep Learning* community refers to latter phase (assimilation) as the *forward pass* of the NN, while the learning mechanism for finding the best interaction (accommodation) is the *backward pass*, in reference to the backpropagation algorithm.

intend to implement two complementary aspects of human-like adaptation once interacting with the environment: learning new skills or improving an existing one.

1. The former consists in achieving a new task once deployed in the environment. For instance, learning to detect camel while the model only knows to differentiate cats and dogs. The prominent approach is to collect a small number of camel images to learn as fast as possible the concept of a camel. This approach refers to as *Transfer Learning* (Pan and Yang 2009), *i.e.* knowledge acquired in the past helps learn a new task quickly, and relates to *Continual and Lifelong Learning* (Parisi et al. 2019), *i.e.* learning new knowledge without forgetting the old one.

2. The latter consists in improving performances on a well-specified task once deployed in the environment. For instance, extending the concept of bikes to mountain bikes, having seen city bikes during training. This approach addresses the ubiquitous problem of *Out-of-Distribution* (OOD) generalization (Amodei et al. 2016; Marcus 2020; Arjovsky 2020), *i.e.* the ability to infer on data that are significantly different from those seen during the training phase. OOD generalization has attracted attention of the community, motivating groundbreaking works in *Unsupervised Domain Adaptation* (UDA) (Pan and Yang 2009), *Robust Deep Learning* (Hendrycks and Dietterich 2019a) or *Invariant Risk Minimization* (Arjovsky et al. 2019).

In this thesis, we will distinguish between learning a new task, which we will refer to as a *transfer* mechanism, and the idea of improving a task, which we will refer to as an *adaptation* mechanism. The latter will be the subject of investigation of this thesis

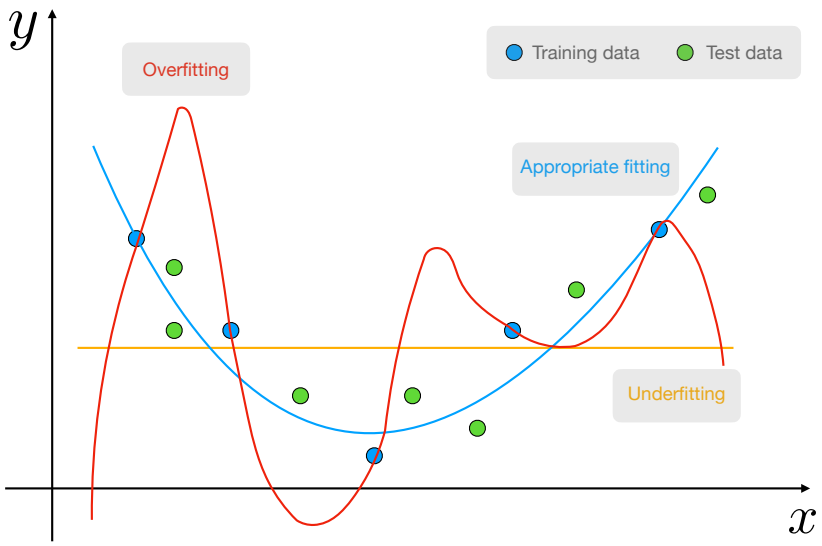


Figure 2.7: Illustration of the ubiquitous problem of overfitting / underfitting in Machine Learning with a simple one dimensional regression. A model is provided with 5 training data points (blue points) and we aim to learn the relation between inputs x and outputs y. We evaluate the goodness of fit on 7 data points (green points) that are not seen during training. In orange, we present the best fit of training data when only considering a simple linear model. Being too simple, the model fails to capture the underlying structure of the data resulting in the situation of underfitting. In red, we present the best fit of training data when considering a complex model, *e.g.* a polynomial function of degree higher than 5. The model is able to learn the training data but fails to understand the simple underlying structure of the data, resulting to a high distance between prediction of test data and the ground-truth, situation referred to as *Overfitting*. In blue, we present a simple, but expressive enough, model. Note that the model may not fit exactly the training data and its simplicity allows it to catch the underlying structure in the data, resulting in an appropriate fit. The trade-off between model simplicity and ability to explain the training data is a crucial aspect of ML.

work. We note that the notion of adaptation described by Jean Piaget encompasses both mechanisms.

2.2 Understanding Adaptation in Machine Learning

Jean Piaget's concept of adaptation refers to the ability to react correctly to something *new*. In the following, we will show how the understanding of the nature of *novelty*, and ultimately characterising what we can do with a ML model once deployed, has been a central issue in developing Machine Learning. First, we will discuss about the *overfitting* problem to revolve around the weakness of deep models when faced with *out-of-distribution* data, *i.e.* data significantly different from data seen during training. We claim that deep models fail to bridge the distribution gap because they do not implement natively the principle of accommodation, motivating our interest into adaptive models.

2.2.1 Generalization and overfitting

The fundamental problem of *Overfitting* occurs when a learner fails to capture the essence of a task but focuses on "remembering" training examples. Thus, evaluating

the model (*i.e.* *testing phase*) on samples that have not be seen during training is arguably the first way to assess how an algorithm reacts to novelty. An illustration of the overfitting phenomenon is presented in Figure 2.7. It shows how a too simple and a too complicated model miss the data's underlying structure, resulting in poor performing models. Formally, the underlying structure in the data refers to the generative process of the couple (x, y) where x is the input and y is the output, viewed as realizations of the random variables X and Y, respectively. It is well-established to frame the generative process as a data *distribution* $p(X, Y)$, where the data[4];

$$\mathcal{D} := ((x_1, y_1), ..., (x_n, y_n)) \sim p(X, Y) \tag{2.1}$$

is an *Independent and Identically Distributed* (IID) sampling from p. Intuitively, each (x_i, y_i) is a sample from $p(X, Y)$ (identically distributed) while it does not depend of (x_j, y_j) for $i \neq j$ (independent). We refer to $\mathcal{D}$ as the *training data*. In that particular case, evaluating novelty consists in computing the model error on data generated from the same process, *i.e.* sampled from the same underlying distribution p, but not seen during training. We refer to this data as the *testing data*.

Statistical Learning Theory (Vapnik 2013) provides fundamental insights about the ability of a learner to infer general patterns from a finite set of samples: one can show the error the model commits on an infinite sampling of data (*Error*, *i.e.* the expectation of the error the model commits on data sampled from $p(X, Y)$) is smaller than the error on the training data (*Empirical Error*, *i.e.* the mean of errors the model commits on training data) plus a term that depends of the model complexity (*Model Complexity*) and the number of training data (*Number of training data*). Such term decreases as more data is available during training (*Number of training data*) as follows;

$$\text{Error} \leq \text{Empirical Error} \underset{\approx}{+} \sqrt{\frac{\log(\text{Model Complexity})}{\text{Number of training data}}} \tag{2.2}$$

where $\underset{\approx}{+}$ disclaims the fact the inequality holds with some probability and involves advanced probability theory tools, and log is the logarithmic function. Crucially, this equation shows that:

1. *Empirical Risk Minimization* (ERM), *i.e.* finding the model that minimizes the empirical error on training data, is a consistent principle for minimizing the overall error on the underlying data distribution.
2. *Generalization*, *i.e.* bridging the gap between the empirical error and the error on the underlying data distribution, can be achieved into two ways:
   - at equal model complexity, one should acquire more training data to bridge the gap,
   - at equal number of training samples, among the models that achieve an equally low empirical error, one should consider the simplest model.

Promoting the simplest model among those that explain equally well the training data, referred to as *Structural Risk Minimization* (SRM), is an instantiation of the philosophical principle of parsimony, also known as the *Occam's razor* which dates to the fourteen century[5].

[4] In the following chapter, except Chapter 6, $\mathcal{D}$ will refer to the set of discriminators.

[5] William of Ockham (circa 1287–1347), *"Numquam ponenda est pluralitas sine necessitate"* ("Plurality must never be posited without necessity").

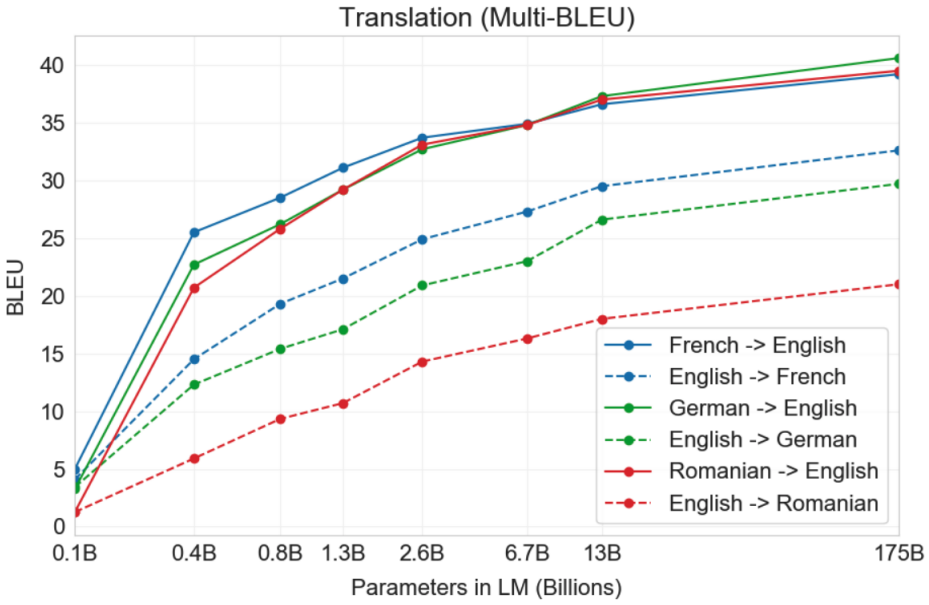


Figure 2.8: As the number of parameters increases in a large language model, here GPT3 (Brown et al. 2020), the performance in few-shot translation, *i.e.* translating with few examples of paired sentences, improves according to the BLEU metric (BiLingual Evaluation Understudy). The number of parameters is here a proxy of model complexity. Figure from (Brown et al. 2020).

From the early eighties to the end of the nineties, *i.e.* before the data explosion era, most of ML community's efforts primarily focused on enforcing appropriate assumptions on the model for preventing from the effect of overfitting, from kernel machines (Cortes and Vapnik 1995; Hofmann, Schölkopf, and Smola 2008) to neural networks such as convolutional (LeCun et al. 1989; LeCun et al. 1998) or recurrent neural networks (Hochreiter and Schmidhuber 1997). The last ten years have been marked by a greater availability of training data and computational resources, thanks in part to affordable *Graphical Processing Units* (GPUs), which are far more powerful than classical *Central Processing Units* (CPUs), that enabled unprecedented progress in *Deep Learning* (LeCun, Bengio, and Hinton 2015). As a result, we have witnessed a shift from the search for simplicity through the principle of parsimony, to the reign of increasingly complex models. In Table 2.1 and Figure 2.8, we report a similar trend we observe both in the Computer Vision (CV) and Natural Language Processing (NLP) communities: as the model complexity increases, the performance improves. Beyond the fact that this race to complexity is within reach of very few players, who combine both access to data and computational resources[6], a more fundamental question is to elucidate if feeding more complex with more data is enough to make ML models better to face novelty?

2.2.2 Beyond the paradigm of "more data for more complex models"

Is "more data for more complex models" enough? Let consider the case of **ImageNet** (Deng et al. 2009), one of the most influential dataset in the Machine Learning community. It contains millions of images, divided into 1,000 classes for the ISLVRC

[6]Trying to replicate GPT3 soon after its release would have cost millions of dollars (https://venturebeat.com/2020/06/11/openai-launches-an-api-to-commercialize-its-research/)

Layers	Top−1 error	Top−5 error
34	24.19%	7.40%
50	22.85%	6.71%
102	21.75%	6.05%
151	21.43%	5.71%

Table 2.1: Number of layers in a Deep Residual Network (He et al. 2016) (ResNet) for image classification on the ImageNet benchmark (Deng et al. 2009). The deeper the network, *i.e.* the more expressive, the lower the model error. The number of layers is here a proxy of complexity.

Predictor	ImageNet (Deng et al. 2009)	ImageNetV2 (Recht et al. 2019)	$\Delta\varepsilon$	$\Delta\varepsilon/\varepsilon$
ResNet	15.8%	24.6%	+ 8.5%	+ 55.7%
Humans	4.8%	5.4%	+ 0.6%	+ 12.5%

Table 2.2: Comparison of top−1 error on ImageNet classification task (ISLVRC challenge with 1,000 classes) between a ResNet classifier (He et al. 2016) and humans (*5 humans were involved in the experiment* (Shankar et al. 2020)) on two different test sets of ImageNet: original (Deng et al. 2009) (ImageNet) and V2 (Recht et al. 2019) (ImageNetV2). Humans show consistent performances while a ResNet (He et al. 2016) classifier suffers of a significant drop of performances. *Original results in (Shankar et al. 2020)*.

challenge[7], and the natural learning task is to predict the correct class of an image. It is a standard benchmark against which the improvement of a new algorithm over the state-of-the-art can be established by its ability to make less errors. Crucially, as the first large-scale dataset, ImageNet significantly contributed to the new advent of neural networks[8].

However, what does ImageNet progress on advances to the field of Computer Vision (CV) mean as a whole? In other words, will the best model on the ImageNet also be the best for real-world applications? As a first attempt to study this thorny question, a line of studies suggests to *reset* the benchmark (Yadav and Bottou 2019; Recht et al. 2019; Miller et al. 2020) to make sure we draw similar conclusion on deep models independently on (random) arbitrary choices. For instance, when resetting the train / test split used for model comparison[9], the findings reveal that the model hierarchy is indeed consistent across test set replicates. However, models suffer of a significant drop of performances while humans perform equally as presented in Table 2.2. The good news is that the best model in a particular configuration is probably the best on a wide variety of configurations[10]. The bad news: the performance on a specific configuration is not a reliable estimate of the true performance metric of a model, *e.g.* its accuracy. Thus, although invisible to humans, models are sensitive to small data shift. In particular, this demonstrates the importance of building reliable benchmarks, as well as the poor ability of metrics, *e.g.* accuracy, to provide a full description of model's behavior (Ferri, Hernández-Orallo, and Modroiu 2009).

[7] www.image-net.org/challenges/LSVRC/

[8] https://paperswithcode.com/sota/image-classification-on-imagenet

[9] It results in having two different test sets, thus two empirical criteria, which should lead to similar conclusions.

[10] Note that it cannot be the best for all configurations according to the *No free-lunch theorem*.

Simply reshuffling the dataset is far from addressing completely our concerns. Over the past decade, a new and more ambitious question has emerged that exhibits some intriguing weaknesses of deep neural networks: how do models behave when faced with something genuinely new, *i.e.* something significantly different from what the model has seen during its learning phase (Torralba, Efros, et al. 2011)? The analysis of image recognition systems from (Beery, Van Horn, and Perona 2018) highlights a striking example of this phenomenon. As presented in Figure 2.9(a), a model fails to recognize a cow on a beach since the beach deviates from the usual context of a pasture. Similarly, (Geirhos et al. 2019) have shown that models trained on ImageNet tend to detect object based on their texture, not their shape (Figure 2.9(b))!

Scaling both dataset size and model complexity is therefore not a *panacea* for Machine Learning! ML models are, in practice, sensitive to subtle shifts in the data or collection bias, *e.g.* wrongly learning that a cow can not be on a beach (Beery, Van Horn, and Perona 2018). Carefully curing large scale datasets while increasing their diversity may prevent a significant part of this undesirable effect. Still, this task is herculean when considering real-world datasets that could typically be composed of thousands to millions of instances. As illustration, ImageNet has 30 mushroom synsets, each with approximately 1000 images.

2.2.3 Generalization to new data distributions

Why models trained by *Empirical Risk Minimization* (ERM) can be easily fooled by spurious patterns in the data, such as "a cow should not be on a beach" (Beery, Van Horn, and Perona 2018)? To gain insight on this weakness, we turn back to the fundamental assumption of ERM; the Independently and Identically Distributed (IID) assumption. The IID assumption means presenting data to the learner without *contextualisation*. For example, imagine a child that has never seen a cow before. Assume we show two pictures of a cow, the former in pasture and the latter in a barn, while indicating the context does not define the animal. Will it be easier for the child to recognise a cow on a beach? Contextualising the image marks a rupture in the environment, to refer to Jean Piaget's terminolgy (Piaget 1936), and forces the accommodation to identify the shared information that defines a cow, thus eventually allows the identification of a cow in significantly new contexts, such as a beach.

This simple example demonstrates that the mathematical assumption that training data is a IID sample has a very tangible connections with the learning dynamic described by Jean Piaget (Piaget 1936). By contrast, one can frame shift in the environment, *e.g.* by contextualizing example, as a violation of the IID assumption. As a result, accommodation, that enables a more powerful learning, can occur only if the distribution of data from the environment differs with the training data. Interestingly, distribution shift is perceived historically as a *risk*, not an *opportunity*[11] for learning which is arguably the vision from Jean Piaget.

The historical vision of distribution as a risk is legitimate. Indeed, one can isolate very simple situations where distribution shift has a dramatic impact on model trained by ERM, as presented in Figure 2.10. Quinonero-Candela et al. describe some

[11] The first prior work that arguably see distribution shift as an opportunity is the pioneering work of *Invariant Risk Minimization* (Arjovsky et al. 2019) through the lens of *causality* (Pearl 2009)

(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97, Seashore: 0.97

(C) No Person: 0.97, **Mammal: 0.96**, Water: 0.94, Beach: 0.94, Two: 0.94

(a) Example from (Beery, Van Horn, and Perona 2018).

(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image
71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

(b) Example from (Geirhos et al. 2019).

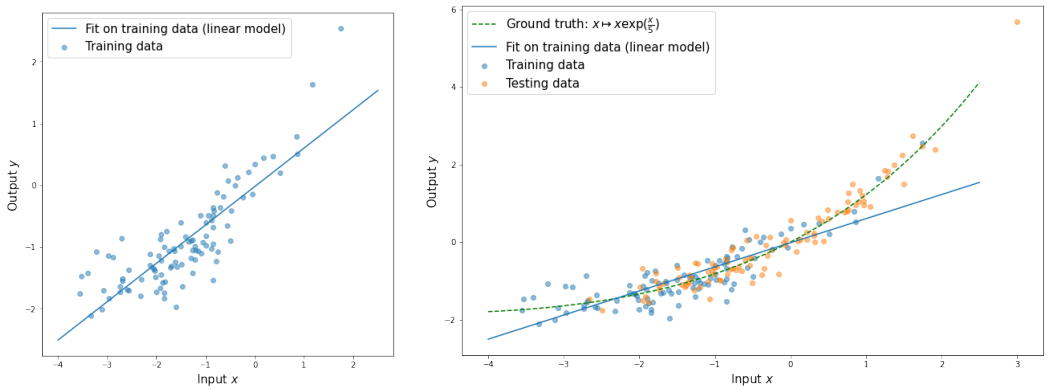
Figure 2.9: (a): A recognition system fails to identify a cow in a image if it differs from the usual context of the pasturage, here a beach. (b): A recognition system is biased towards spurious features, here texture, and forget the shape features. Humans probably use both shapes and textures to detect objects. However, this example shows us that the model greatly favours textures. Indeed, `tabby cat` is not in the top-3 of predictions.

types of shifts that may occur in the real-world by decomposing how the relation between X and Y may vary (Quinonero-Candela et al. 2009);

$$p(X,Y) = p(Y|X)p(X) = p(X|Y)p(Y) \tag{2.3}$$

From this equation, and assuming that only one term can vary once deployed in the environment, there are four types of shifts that we can easily relate to real world situations;

- **Covariate Shift:** $p(X)$ changes while $p(Y|X)$ is conserved. It often refers as a *sample selection bias*, *e.g.* unbalanced number of men and women in the training data while the real-world is balanced. The covariate shift is typically the situation presented in Figure 2.10.
- **Label Shift:** $p(Y)$ changes while $p(X|Y)$ is conserved. The class distribution has changed, for instance for a recognition system that classifies cat or dogs, the training data has the same number of cats and dogs but the real-world is more populated by dogs.



(a) (b)

Figure 2.10: Overview of the risk of distribution shift on a linear model trained by ERM to fit the function $f : x \mapsto x \exp\left(\frac{x}{5}\right)$ (Ground truth in (b)). Training data input x is generated by sampling with respect to $\mathcal{N}(-1.5, 1)$ while the testing data input x is generated with respect to $\mathcal{N}(-0.25, 1)$, where $\mathcal{N}(0, 1)$ is the normal distribution. Given the input x, the output y is generated by sampling with respect to $\mathcal{N}(f(x), 0.1)$ for both the training data and the testing data. (a): When provided with the training data, a linear model seems to explain well the underlying structure. Note that training data located at the upper right are above the model. However, it is difficult to attribute this phenomenon to the noise (aleatoric) or to the fact the target function f can not be fully approximated with a linear model (epistemic). (b): Testing data, *i.e.* data encountered after deployment, has a different distribution of inputs x, while the distribution of outputs y given x is the same than the training data. However, this shift of input distribution shows that the underlying structure of the data can not be fully explained with a linear model which could not be known during training. Thus, the model performs worse on the test data than the training data would suggest.

- **Conditional Shift:** $p(X|Y)$ changes while $p(Y)$ is conserved. The feature distribution has changed, for instance cows are sampled in pasturage in the training data, but cows may also appear on a beach in the real-world.
- **Concept Drift:** $p(Y|X)$ changes while $p(X)$ is conserved. The signification of the features has changed, for instance due to the failure of a sensor.

Prior works have produced an extensive literature to address the problem of covariate and label shift (Quinonero-Candela et al. 2009). However, most real world applications present types of shift that do not fall into the four types of shift presented in (Quinonero-Candela et al. 2009). Indeed distribution shift is characterized with non-overlapping distributions (D'Amour et al. 2021) as presented in Figure 2.11, which is typically the case for high dimensional data such as texts or images[12].

Learning under distribution shift has been theoretically studied in (Ben-David et al. 2010a), *A Theory of Learning from Different Domains*, which provides the most influential depiction of the risk of data shift, as well as the first consistent theory (Ben-David et al. 2010b). Given a model trained on source data and deployed on target data, the target error is related to the source error as follows;

$$\text{Target error} \leq \text{Source error} + \text{Distribution Discrepancy} + \text{Adaptability} \quad (2.4)$$

[12]Texts and images are typical data with high dimension, increasing the risk to obtain non-overlapping supports in a situation of distribution shift.

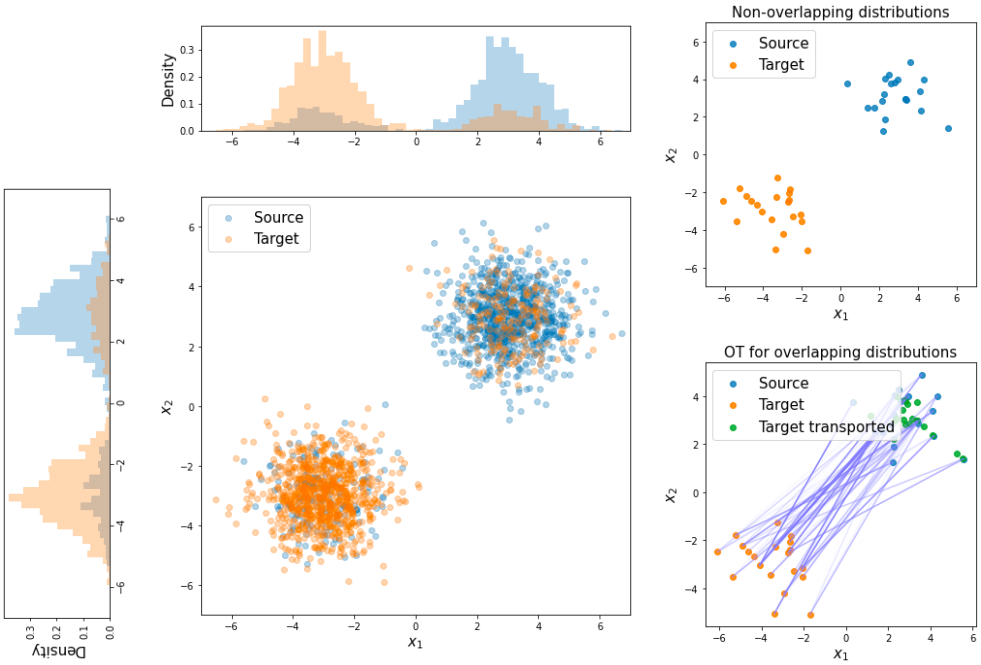


Figure 2.11: Illustration of the difference between overlapping data (a) *vs* non-overlapping data (b). Source and target data are sampled from a two dimensional gaussian mixture model, *i.e.* $\pi \cdot \mathcal{N}(-1, 1) + (1 - \pi) \cdot \mathcal{N}(1, 1)$ where π is Bernoulli random variable with parameter p: $\mathbb{P}(\pi = 1) = p$. (a): Illustration of two overlapping but different distributions where $p = 0.8$ for the source data while $p = 0.2$ for the target data, resulting to overlapping data. (Upper b): Illustration of two non-overlapping distribution where $p = 1$ for the source data while $p = 0$ for the target data, resulting to non-overlapping data. (Lower b): As illustration, a major strategy for obtaining two overlapping distributions from overlapping distributions is to transport one to the other through the lens of Optimal Transport (Peyré, Cuturi, et al. 2019; Courty et al. 2016).

Although this inequality seems simple in form, it captures the underlying dynamics of generalization under distribution shift, namely;

- **Source error:** quantifies if the model works correctly on the source data,
- **Distribution Discrepancy:** quantifies the worst-case model changes due to the shift of distribution of inputs. Thus, one can relate the distribution discrepancy with the notion of sensitivity the model to changes (from source to target) in the data,
- **Adaptability:** quantifies if it exists a model that works correctly on both source and target data.

Fortunately, as described by (Ben-David et al. 2010a), the adaptability is probably a small error for real-world applications and quantifies the ability to adapt at first. For this reason, much research has focused on building machine learning models that are insensitive to distribution shift (Quinonero-Candela et al. 2009; Ganin and Lempitsky 2015; Long et al. 2015).

On the one hand, the analysis of the risk the model faces when applied to a different data distribution is strongly related to the Occam's razor;

- the simpler the model, the lower the sensitivity to changes in the input distribution, thus reducing the distribution discrepancy,
- while one can reasonably believe that it exists a simple model that works correctly on both distributions, resulting in a good adaptability.

On the other hand, the analysis is over-pessimistic for deep learning models due to their high complexity; resulting to a likely high distribution discrepancy.

The seminal idea from (Ben-David et al. 2007) circumvents this problem by transforming the inputs to build a *representation* of the data[13]. Crucially, the more 'similar' are representations when drawn from different distributions, the less sensitive the resulting model. In Figure 2.12(a), we present a visual explanation of the effect of obtaining more similar representations, that are usually referred to as *Invariant Representations*. To relate this approach with the ERM principle, processing of inputs allows to build a version of data such that source and target data look like that are sampled from the same underlying process, despite the fact the original data may not. In Figure 2.12(b) and 2.12(c) , we present a simple case in a polynomial regression of the sine function. Here, the periodicity of the function allows us to extrapolate outside the data available for training. The art of finding a suitable representation, *e.g.* understanding the role of periodicity in the sine regression example that allows to derive the representation $\varphi : x \mapsto x \text{ modulo } 2\pi$, is, therefore, the essence of adaptation. In this dissertation, such a knowledge (knowing the periodicity of sine in the example above) will be referred to as the *inductive bias* and will be formally studied in Chapter 6.

2.3 Problematic & Industrial Goals

Through the founding work of Jean Piaget about the place of adaptation in developing intelligence, we have established fruitful connections with the research path in Machine Learning. In particular, we claim that reacting correctly to *novelty* is the underlying goal when building ML systems. In the following, we take the industrial point of view of the challenge of dealing with novelty. Indeed, the engineering counterpart of ML has recently put a particular focus on deploying ML systems that handle novelty through the lens of Machine Learning Operations (MLOps). For instance, by controlling that the data received in production are sampled from the same distribution as the training data through a shift detector. In this section, we aim to characterize the promise of adaptation for the future of MLOps, a critical industrial topic in ML engineering.

2.3.1 Machine Learning Operations (MLOps)

Machine Learning Operations (MLOps) manages the deployment of models in production to ensure that they operate securely and robustly to deliver value. MLOps is therefore concerned with the life cycle of a model, including model re-training, reproducibility (dataset and model versioning), scalability, detection of anomalous data, and the process of annotating new data to enrich the training dataset. Thus, MLOps bridges the gap between developing a model *in vitro* (with training data) and monitoring it *in-vivo* (with real-world data). An overview of the components of

[13] Even if (Ben-David et al. 2007) precedes (Ben-David et al. 2010a), a modern derivation of the role of representations to address distribution shift can be obtained by applying Equation 2.4, not directly to inputs, but to representations.

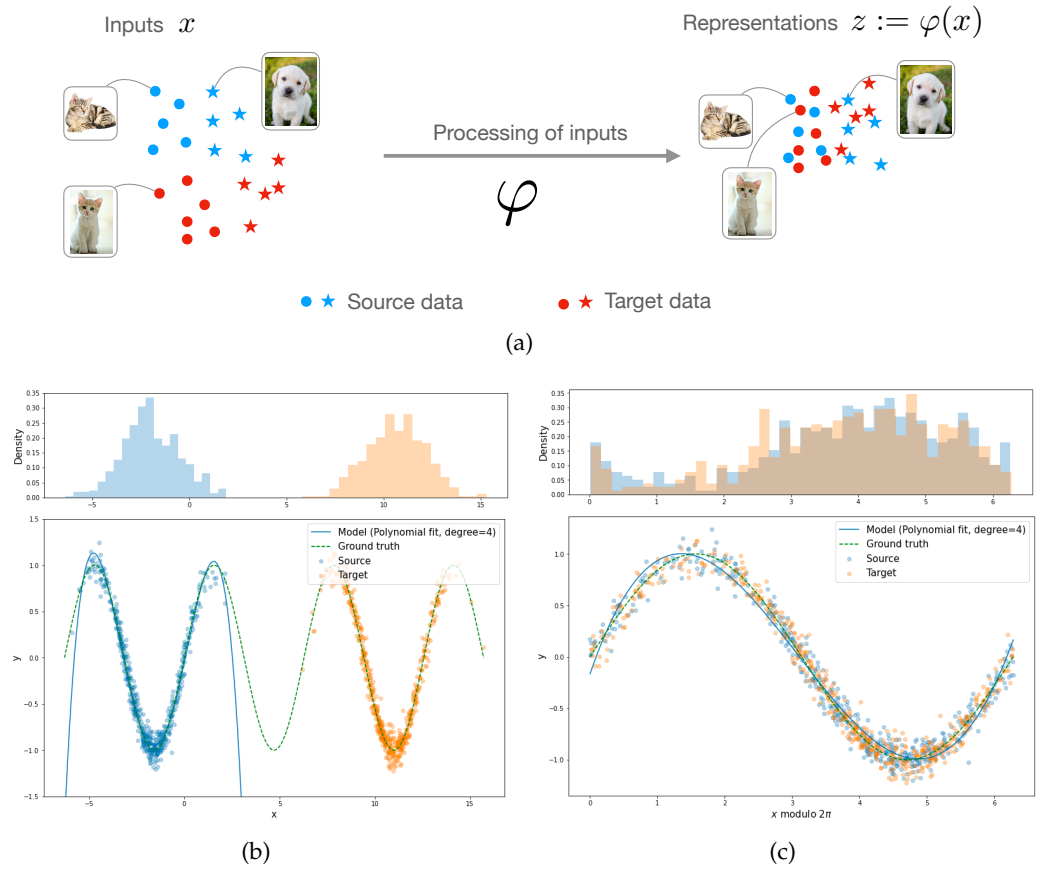


Figure 2.12: (a): Overview of the mechanism of processing for obtaining a representation z of an input x. (Left) The source and target inputs x are different, resulting to a clear separability between the source and the target inputs x. (Right) After processing, through some function φ, of inputs x to build representations z resulting to similar source and target representations z, called *Invariant Representations*. (b) and (c): Illustration of the search invariance for adaptation when regressing the sin function with polynomial. (b): Source and target data are sampled from different distributions, resulting in two non-overlapping histograms of inputs x. While the polynomial fit works correctly on the source data, the model fails to generalize out of the source data, in particular for the target data. (c) We process inputs in both domains by applying the function $x \mapsto x$ modulo 2π, leading to close histograms, *i.e.* data distributions of inputs are similar in both domains. The polynomial fit on processed source data leads to good performances in both domains.

an MLOps architecture is provided in Figure 2.13. For the ease of reading, we use a colour code to differentiate training data, *i.e.* source data, from test data, *i.e.* target data. Objects that depend on either training data or test data inherit this colour code; for example, we note $h^\star$ a model trained on training data.

2.3.2 Positioning

We position important lines of research in ML that intervene in different components, or under different assumptions, in an MLOps architecture. Importantly, it allows to characterize the problem of adaptation with respect to similar topics, as well as position the long-term goals of this thesis. In particular, we describe the paradigms of Transfer Learning, Continual and Lifelong Learning, Active Learning, Robust Deep Learning, Shift Detection, Unsupervised Domain Adaptation and Domain Generalization. A summary of this effort to synthesize and discriminate these

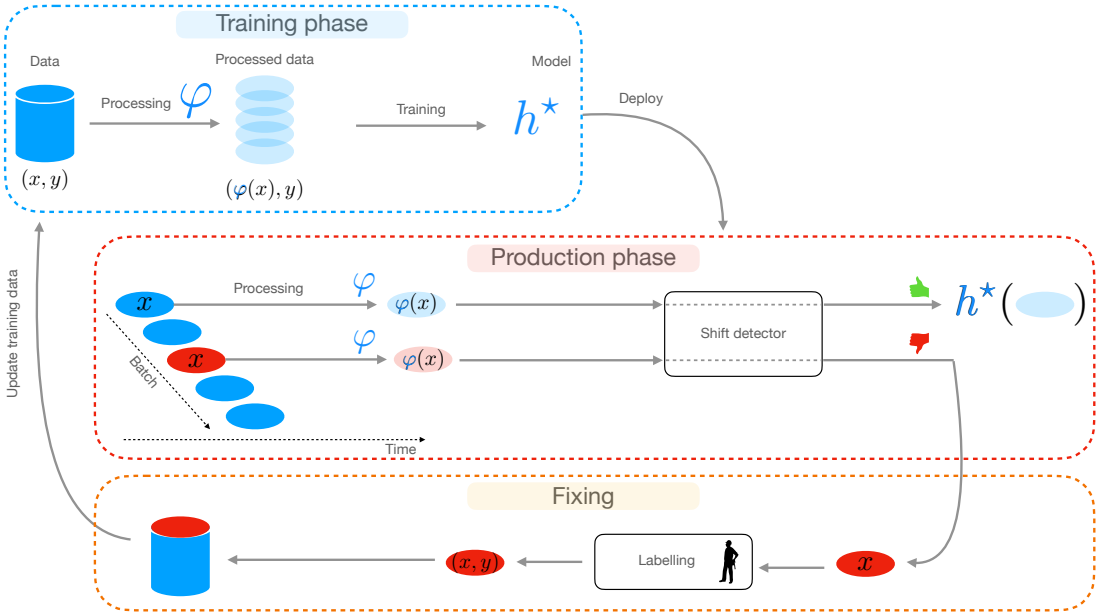


Figure 2.13: Overview of *Machine Learning Operationals* (ML Ops), an industrial key challenge for the safe deployment of Machine Learning models in production. We have detailed three critical components; the *Training phase* (blue), the *Production phase* (red) and the *Fixing phase* (orange). The Training phase includes data collection to build a dataset of samples (x, y) and the processing phase, *i.e.* determining a suitable transformation φ of inputs x to build representations $\varphi(x)$. Then, training consists in learning the best model $h^\star$ from processed inputs $\varphi(x)$ (including validation). The Production phase consists in the inference that we model as follows. Over time, the system receives batches of data for inference, *i.e.* to provide a prediction. The size of these batches may vary, and this will be an essential component of our discussion in Part IV. A shift detector checks whether the data conforms to the data on which the model has learned. If they are consistent, then the inference is performed. If it does not, then it is analysed during the fixing phase. Note that if the data moves away from conformity over time, it may be necessary to stop the model in production. The analysis of non-conforming data during the fixing phase is akin to determining the ground truth of this data, which may require significant human support, mainly through human labelling. When this phase ends, the model is updated, which we show with a feedback loop to the Training phase, where the training data has had the non-conforming data encountered in production added. Note that we emphasize the dependence of both the representation φ and the best model $h^\star$ with the training data with the blue color.

active areas in ML is given in Table 2.3. In particular, we rely on the assumptions made during training (train-time), once the model is deployed (test-time), or if we are interested in learning a new task or improving an existing one, following the discussion from Section 2.1.3.

Transfer Learning. *Transfer Learning* (Pan and Yang 2009) is undoubtedly responsible for the wide adoption of Deep Learning in the Machine Learning community. Today, most vision or natural language understanding models are pre-trained on large datasets to be transferred to a target task where only little data is available. In this way, the model takes advantage of the knowledge acquired on a source task to learn a target task more quickly. For example, deep learning models for object recognition are pre-trained on ImageNet (Deng et al. 2009), where the source task is to classify an image among 1000 available classes. Recently, self-supervised learning (Jing and Tian 2020; Brown et al. 2020) models suggest building a pretext task, allowing to learn from a source task, even without labels. Transfer Learning occurs in

		Train-Time	Test-Time	
			Evaluation	Information
	Supervised Learning	$\mathcal{L}$	$\mathcal{U}$	
	Semi-Supervised Learning	$\mathcal{L}, \mathcal{U}$	$\mathcal{U}$	
New	Transfer Learning	$\mathcal{L}_S$	$\mathcal{U}_T$	$\mathcal{L}_T$
New	Continual and Lifelong Learning	$\mathcal{L}_S$	$\mathcal{U}_S, \mathcal{U}_T, ..., \mathcal{U}_T^\infty$	$\mathcal{L}_T^1, ..., \mathcal{L}_T^\infty$
Improve	Active Learning	$\mathcal{L}$	$\mathcal{U}$	Oracle
Improve	Unsupervised Domain Adaptation	$\mathcal{L}_S, \mathcal{U}_T$	$\mathcal{U}_T$	
Improve	(Offline) Test-Time Adaptation	$\mathcal{L}_S$	$\mathcal{U}_T$	$\mathcal{U}_T$
Improve	Online Test-Time Adaptation	$\mathcal{L}_S$	$\mathcal{U}_T^1, ..., \mathcal{U}_T^\infty$	$\mathcal{U}_T^1, ..., \mathcal{U}_T^\infty$
Improve	Domain Generalization	$\mathcal{L}_S^1, ..., \mathcal{L}_S^\infty$	$\mathcal{U}_T$	

Table 2.3: Overview of ML paradigms that are related to the notion of Adaptation as described by Jean Piaget, including standard Supervised Learning, Semi-Supervised Learning (Chapelle, Scholkopf, and Zien 2009), Transfer Learning (Pan and Yang 2009), Continual and Lifelong Learning (Parisi et al. 2019), Active Learning (Settles 2009), Unsupervised Domain Adaptation (Pan and Yang 2009), offline/online Test-Time Adaptation (Wang et al. 2021a) and Domain Generalization (Gulrajani and Lopez-Paz 2021). We note a labelled dataset $\mathcal{L}$ and an unlabelled dataset $\mathcal{U}$, index notation S and T refers to *source* data , *i.e.* the data available at train-time, and *target* data, *i.e.* the data observed in the environment of deployment, respectively. This denomination is widely used in the ML community (Pan and Yang 2009). We made explicit scenarios that fall into learning a new (**New**) task *vs* improving (**Improving**) an existing one. Our description is inspired from Table 2 of (Gulrajani and Lopez-Paz 2021) but we separate the different assumptions made at train *vs* test-time, *e.g.* information available at test-time for adapting the model. For instance, the problem of Domain Generalization (DG) and Unsupervised Domain Adaptation (UDA) differ in two ways; domain generalization assumes access to source labelled data from different domains while UDA assumes access to unlabelled target data.

an MLOps architecture during model re-training after acquiring new labelled data collected in the environment. In this particular case, model re-training involves only new labelled data, not the old data.

Continual Learning and Lifelong Learning. *Continual and Lifelong Learning* (Parisi et al. 2019) is a close topic to Transfer Learning. Both of them assume that knowledge acquired by achieving a task in the past may help to learn faster a new task in the future. Continual and Lifelong Learning is a more ambitious paradigm since it aims to learn a new task without forgetting the ones learned in the past, which contrasts with Transfer Learning which only focuses on achieving good performances on the new task. Continual and Lifelong Learning occurs in an MLOps architecture during re-training after acquiring new labelled data collected in the environment. In this particular case, model re-training involves both the new labelled data and old data.

Active Learning. *Active Learning* (Settles 2009) addresses the question if some data is more informative than others for learning a model, thus, allowing to train a better model with fewer labelled data that may be expensive and time-consuming to acquire. The standard Active Learning setup assumes that we have access to an unlabelled dataset from which we can query samples for annotation. Once selected, data is sent to an Oracle (*e.g.* a human annotator) to get the label. Finally, we add the new labelled data to the training data set to update the model. Active learning, therefore, consists in designing the query that will maximise model's performance.

Active Learning occurs in an MLOps architecture in the fixing phase by selection a subset of anomalous data observed in production, *i.e.* data subject to improve the model once annotated and added to the training data.

Unsupervised Domain Adaptation. UDA (Quinonero-Candela et al. 2009; Pan and Yang 2009) will be at the heart of this thesis work. One can frame UDA as learning a well-performing model on target **unlabelled** data provided with the knowledge of source **labelled** data. UDA occurs in an ML Ops architecture during model re-training after acquiring new **unlabelled** data collected in the environment. In this particular case, model re-training involves only unlabelled data, not the old data. Note the novel paradigm of source-free UDA focuses on the case where model re-training is performed without the old data. A fundamental difference with UDA and Transfer Learning, and by extension Continual and Lifelong Learning, is that the task, from the old to the new data is the same. In particular, Transfer Learning and Continual and Lifelong require new labelled data, while UDA focuses on information provided by new unlabelled data.

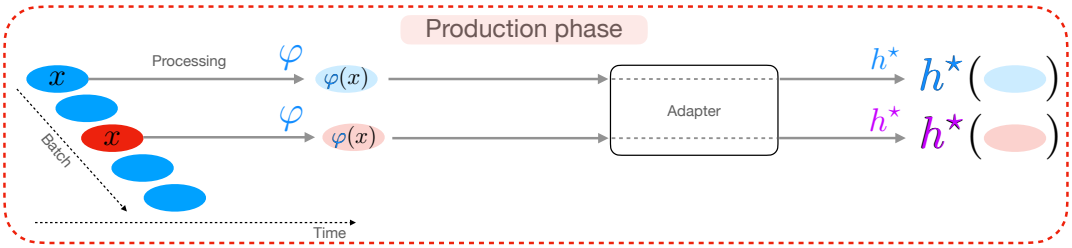
Other related topics. For completeness, we mention the field of *Domain Generalization*, also known *Out-of-Distribution* (OOD) generalization that focuses on generalizing on new data without re-training the model. Addressing this challenging problem requires a model robust to distribution shift. An influential line of study assumes access during training to numerous different data distributions (domains) to learn an invariant model that may result in better performance once deployed in the real world (Arjovsky et al. 2019; Arjovsky 2021; Gulrajani and Lopez-Paz 2021). We also mention the field of *Robust Deep Learning* where one aims to make to natural distribution shift (Hendrycks and Dietterich 2019b) or to adversarial attacks (Madry et al. 2018). Furthermore, *Anomaly Detection* (Chandola, Banerjee, and Kumar 2009; Markou and Singh 2003) aims at isolating data that are significantly different from training data, that follows the field of research of shift detection. Additionally, learning from data streams, *i.e.* assuming that data depends on time, is a very active area of research (Gama 2010).

2.3.3 The promise of Adaptation

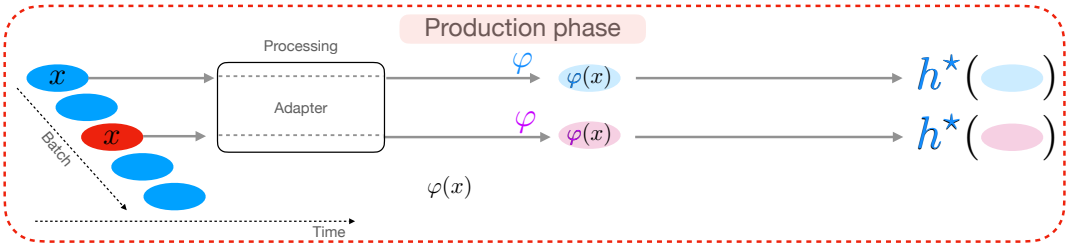
The *shift detector* is a major bottleneck of an MLOps architecture. It detects if data received in production is conformed with training data, that guarantees the model will not face to a detrimental data shift. An extensive comparison of detection strategy is investigated in (Rabanser, Günnemann, and Lipton 2019). When data received in production is subject to shift over-time, one may need to stop the model for model re-training, referred to as the *fixing phase*. In this context, model adaptation offers new opportunities for MLOps. The overall idea is to replace two critical components; shift detection and fixing phase, the latter that may involve human intervention, with an autonomous module named *Adapter*. The Adapter module aims to adapt the model once deployed to better fit the data. We present in Figure 2.14 two strategies for designing an Adapter, depending on adaptation takes on the pre-processing phase (adapting the representation φ) or on the model (adapting the model).

A case of study at Sidetrade. Sidetrade[14] is a French tech company which provides AI solutions for Business-to-Business (B2B) companies. Namely, Sidetrade helps its

[14] www.sidetrade.com



(a) Adapter modifies the model $h^\star$.



(b) Adapter modifies the representations φ.

Figure 2.14: Overview of the Production phase for Machine Learning Operationals (ML Ops). We follow the similar color code than 2.13. Additionally, when an object tends to bridge training and test data, we use the colour purple. For example, we note φ an invariant representation, *i.e.* with respect to the training and test data. The promise is to free ML Ops from the fixing phase, which is often costly because it requires the intervention of human experts, particularly for labelling. We expose two approaches for adaptation. (a): When data is not conformed, the model is adapted based on a batch of data received in production, resulting to a new model $h^\star$. Note that in this case the representation φ is unchanged. (b): When data is not conformed, the representation is adapted based on a batch of data received in production, resulting to a new representation φ. Note that in this case the model $h^\star$ is unchanged. We emphasized the adaptation with the purple color has adaptation is a mix between information gained from source data and target data.

clients with its AI solutions to discover and acquire new customers, estimate the churn risk of existing customers[15] in order to prioritize action plans, and automatically optimize the cash flow by learning strategies to collect unpaid invoices from creditors more efficiently. Core functionalities of those AI solutions are powered by a variety of Machine Learning algorithms. Examples include computing credit-risk scores, categorizing the business activity of companies using multilingual word embeddings, and classifying inbound emails to generate automatic actions. Sidetrade leverages data aggregated from various sources including open data, web data and financial data. In the context of this CIFRE thesis, Sidetrade wants to better address the shift observed between training data and production data. Most of the time, this shift results from learning models on an existing customer data, which is not representative of the diversity for every possible customer. In a sense, we can consider that the customer identity as a nuisance factor, since the goal is to obtain a model that generalizes well to previously unseen customer data.

Example 2.3.1 (A case of study at Sidetrade). *An important step of the invoice cash collection process is the back-and-forth communication between the debtor and the collection agent. Most of the communication occurs over email, and it's a critical part to consider if the process is to even be partially automated. A sizable proportion of the emails falls into*

[15]Customer of Sidetrade solutions are named 'clients'. Customer of Sidetrade clients are called 'customers'.

a small number of cases that are feasible to automatically detect, and are then easy to act upon[16]*. Automating these cases frees up the time of the collection agents and helps them focus on more demanding tasks. We have identified that models have the capacity to fit, from inbound emails, the specific behavior of debtors. This results from a bias in the data where some customers are more subject to lost invoices or to dispute on price and quantities. We have observed that such a phenomenon significantly hurts the performance of the model on customer not seen during training. In this particular case, the nuisance factor is the customer identity. Manually removing from the raw text, words and statistical patterns that directly leak the customer identity is very time consuming since it needs to be done for every Sidetrade client.*

Interestingly, there are situations where the nuisance factors that may impact the data shift can be easily identified and annotated, as described in Example 2.3.1. However, manually correcting the effect of each nuisance factor is time consuming and sometimes not even possible. For instance, if a cash collector knows customers each have a specific behaviour, then we can *a priori* identify the customer identity as a potential nuisance factor. But preventing its impact implies to carefully process / balance the data and to design train / test procedure for sanity check. Developing learning architectures able to correct the effect of identified nuisance factors is then strategic for legitimating the use of AI solutions for addressing hard tasks such that cash flow management. Ultimately, Sidetrade has the ambition to design generic models that will have a reasonable predictive power even for new clients, for which no, or very few, labeled data is available. In an MLOps architecture, transfer learning occurs in the data re-training stage after acquiring new labelled data collected in the environment. In this particular case, model re-training involves only new labelled data, not the old data. Note that this brings new flexibility since it is possible to learn a new task if the expert considers it better matches environment observation.

[16]For instance, a debtor may have lost the original invoice and ask for a duplicate; or he may promise to pay the invoice at a specified date.

3 Background & Related Works

Contents

The present chapter provides the needed background of this thesis. The main objective is to describe the invariance of representations adequately to introduce the trade-off between invariance and transferability of representations. This fundamental trade-off will be at the heart of our reflections. It is a self-contained description of the related works that presents founding results and points to reference works that provide broader investigations.

We first review the principle of *Empirical Risk Minimization* (ERM) (Section 3.1) that still has a profound impact on the field of Machine Learning. The core assumption of the ERM principle, *i.e.* that data is an IID sample from the true generating distribution, is a very strong one. Indeed, it implies that it is not feasible to identify models which will correctly generalize to distributions that shift, something of a common characteristic for real-world applications (see the Chapter 2).

To this purpose, we review in the second part of this chapter (Section 3.2) the seminal theory of learning from different domains, which serves also as the foundation for learning under distribution shift (Ben-David et al. 2010a). This theory introduces a fundamental trade-off by relating the target error with the source error through two terms. The former, called distribution discrepancy, estimates the impact of the shift of inputs distribution on the underlying class of models. The latter, called adaptability, embodies our capacity to learn a model that performs well on both the source and the target distributions. Ultimately, one can estimate the discrepancy through additional knowledge; the access to a set of unlabelled samples from the target distribution. This central idea establishes the field of *Unsupervised Domain Adaptation* (UDA) (Quinonero-Candela et al. 2009; Pan and Yang 2009), where robust models

are obtained from labelled source data and unlabelled target data, a particular instance of the transductive learning paradigm presented in Section 3.1.4.

The third part of this chapter (Section 3.3) is dedicated to the paradigm of learning invariant representations. A crucial insight is that domain invariant representations are closely related to the founding theory of Section 3.2, and allow to derive an efficient deep learning approach for UDA (Section 3.3.2) for adapting models to new domains. However, domain invariant representations are still submitted to the fundamental trade-off of (Ben-David et al. 2010a), opening the field of *improving transferability of domain invariant representations* for which we review founding work in Section 3.4. We note that invariance is a more general principle that irrigates ML and thus finds applications beyond UDA that we present in Section 3.3.3.

3.1 Learning Theory

We introduce the paradigm of *Statistical Learning* and standard notations in Section 3.1.1. We provide a depiction in Section 3.1.2 of the ubiquitous principle of *Empirical Risk Minimization* that is the bedrock of *Machine Learning* (ML). Through theoretical bounds, we exhibit the underlying trade-off behind the consistency of the general principle of ERM; minimizing the error on the available data while maintaining a model as simple as possible. This trade-off is partially addressed by *Structural Risk Minimization* (SRM) presented in Section 3.1.3. The idea of learning a model that performs well on the whole data distribution is referred to as *inductive learning* (Vapnik 2013). Even when equipped with proper structure in the model, *i.e.* suitable regularization, inductive learning is an ambitious objective, which can typically fail when not enough data is available for learning. By opposition to inductive learning, *transductive learning* (Vapnik 2013), presented in Section 3.1.4, aims to simplify this objective through additional knowledge about the distribution of inputs, that has motivated the influential field of *Semi-Supervised Learning* (SSL).

3.1.1 Preliminaries

Statistical Learning aims to derive from a set of observations a model that is used later for making predictions on new data, *i.e.* to perform inference. Statistical Learning is a general principle that embraces many domains of science. *Machine Learning* (ML) aims to *automate* the process of Statistical Learning while *Learning Theory* provides a formal description of ML (Bousquet, Boucheron, and Lugosi 2003).

In this general form, we express a model as a function f from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$; $f : \mathcal{X} \to \mathcal{Y}$. Given an input $x \in \mathcal{X}$, the model predicts an output value $y = f(x) \in \mathcal{Y}$. For instance given an image x containing a dog playing with a ball, the model's output may be $f(x) =$ "dog", $f(x) =$ "ball" or a $f(x) =$ "a dog playing with a ball". A *task* is the specification of targeted relation between an $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ that we want to learn. For instance, in the previous example, if the targeted task is animal recognition on image, the model prediction could be $f(x) =$ "dog". Note that specifying a task consists essentially in specifying the output space $\mathcal{Y}$. When $\mathcal{Y}$ is a finite set, we speak of $y \in \mathcal{Y}$ as a *class* with *label* y and refer to the relation between $\mathcal{X}$ and $\mathcal{Y}$ as a *classification task*. When $\mathcal{Y}$ is an interval of $\mathbb{R}$, we refer to the relation between $\mathcal{X}$ and $\mathcal{Y}$ as a *regression task*. For the sake of simplicity, we restrict our presentation to the case where $\mathcal{Y} = \{0, 1\}$, *i.e.* a binary classification task, since the presented results can be generalized to classification with an arbitrary number of classes, as well as regression.

Given a set of $n \in \mathbb{N}^\star$ observations $\mathcal{D} := (x_i, y_i)_{1 \leq i \leq n} \in (\mathcal{X} \times \mathcal{Y})^n$, Statistical Learning aims to identify the model f, also called a learner, that explains best the relation between inputs x and outputs y, for instance by measuring the *Empirical Error*, noted $\widehat{\mathrm{Err}}_{\mathcal{D}}$, of the model;

$$\widehat{\mathrm{Err}}_{\mathcal{D}}(f) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(f(x_i) \neq y_i) \tag{3.1}$$

where $\mathbb{I}(e)$ is the indicator of an event e, *i.e.* $\mathbb{I}(e) = 1$ if e is True and 0 is e is False, here $\mathbb{I}(f(x_i) \neq y_i) = 1$ if $f(x_i) \neq y_i$ and $\mathbb{I}(f(x_i) \neq y_i) = 0$ otherwise. The empirical error is thus the percentage of errors made by the learner on the set of observations, where an error if $f(x) \neq y$. As $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ come with a dependence structure, which specifies notably the targeted relation between x and y, we formally consider

both x and y as realizations of two random variables X and Y, respectively. Thus, the data structure is fully described by the distribution of the couple (X, Y) noted $p(X, Y) \in \mathcal{P}$ where $\mathcal{P}$ is the set of joint distributions on $\mathcal{X} \times \mathcal{Y}$. We note $(x, y) \sim p$, $x \sim p$ or $y \sim p$ for joint $p(X, Y)$, covariate distribution $p(X)$ or label distribution $p(Y)$, respectively, the context solving the ambiguity. We note $\mathbb{E}_p[\cdot]$ the expectation with respect to p. From this probabilistic perspective, the model's performance is measured as an expectation of errors on a data distribution $p \in \mathcal{P}$:

$$\mathrm{Err}_p(f) := \mathbb{E}_p[\mathbb{I}(f(X) \neq Y)], \quad p \in \mathcal{P} \tag{3.2}$$

which is consistent with $\widehat{\mathrm{Err}}_p(f)$ when the number of observations is a set of independent realizations from $p(X, Y)$ (Identically and Independently Distributed (IID)). Our presentation focuses on the error of a model f defined in Equation 3.2. Note that we can generalize the notion of error through a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ that quantifies the discrepancy between $\hat{y} = f(x)$ for some $x \in \mathcal{X}$ with the ground-truth $y \in \mathcal{Y}$, where $y \sim p(Y|X = x)$. For instance, the usual loss used for regression task is the L^2 risk defined as follows: $\ell_2(y, \hat{y}) := (y - \hat{y})^2$.

Supervised Learning is the task of finding a model f that minimizes $\mathrm{Err}_p(f)$ based on a finite IID sampling from $p(X, Y)$. The term supervision refers to the access of both (x, y), while *Unsupervised Learning* characterizes the situation when the output y is not available, which is *de facto* a more challenging problem. Formally, we look among the set of measurable functions from $\mathcal{X} \to \mathcal{Y}$ for the function that minimizes the error. The general case, *i.e.* for an arbitrary distribution $p \in \mathcal{P}$, it may not exists a (measurable) function f from $\mathcal{X}$ to $\mathcal{Y}$ that achieves a null error, simply because there is not enough information in X for determining precisely Y. However, it exists a measurable function f that achieves the best performance overall;

Theorem 3.1 (Optimal Bayes classifier)**.** *Let $p \in \mathcal{P}$, the Optimal Bayes classifier is defined as $f_\eta : x \mapsto \mathbb{I}\left(\eta(x) \geq \frac{1}{2}\right)$ where*

$$\eta : x \mapsto \mathbb{E}_p\left[Y|X = x\right] \tag{3.3}$$

f_η achieves the lowest error, hence its optimality i.e. for all measurable function f, $\mathrm{Err}_p(f) \geq \mathrm{Err}_p(f_\eta)$.

Proof. Let f a measurable function from $\mathcal{X}$ to $\mathcal{Y}$,

$$\begin{aligned}
\mathrm{Err}_p(f) &= \mathbb{E}_{(X,Y)}[\mathbb{I}(f(X) \neq Y)] \\
&= \mathbb{E}_X \mathbb{E}_{Y|X}[\mathbb{I}(f(X) \neq Y)] \\
&= \mathbb{E}_X\left[\eta(X)\mathbb{I}(f(X) = 0)\right] + \mathbb{E}_X\left[(1 - \eta(X))\mathbb{I}(f(X) = 1)\right]
\end{aligned}$$

To minimize it, we have to assign $f(X) = 0$ if $1 - \eta(X) \leq \eta(X)$ *i.e.* $\eta(X) \geq \frac{1}{2}$ and $f(X) = 1$ if $\eta(X) < 1 - \eta(X)$ *i.e.* $\eta(X) < \frac{1}{2}$ which coincides with the definition of f_η. □

A decisive part of the proof is the fact that given x, $\eta(x)$ represents the probability of y to take 1 as value. It reveals an important property of a classifier; it can be built by estimating the probability that given x, the label y takes 1 as value.

Definition 3.1.1 (Probabilistic classifier). *A probabilistic classifier f is defined as*

$$f : x \mapsto \arg\max_{y \in \mathcal{Y}} \tilde{f}(x)(y) \tag{3.4}$$

where $\tilde{f}$ is a measurable function from $\mathcal{X}$ to $\mathcal{P}(\mathcal{Y})$, the set of distributions on $\mathcal{Y}$.

Note that the Optimal Bayes Classifier is a probabilistic classifier where $\tilde{f} = \eta$. Since $\tilde{f}$ estimates class probability of a sample x, it can reflect the level of uncertainty about the true label. To conclude, learning is intrinsically connected to our ability to recover efficiently, from a finite set of samples, the optimal bayes classifier η. In the following, we present the influential principle of *Empirical Risk Minimization* to achieve such a goal.

3.1.2 Empirical Risk Minimization (ERM)

Machine Learning (ML) aims to build a model f that will predict accurately the output y of a complex system given its input x. In Section 3.1.1, we have formalized this objective through the lens of probability, *i.e.* learning f that minimizes the error defined in Equation 3.2. The principle of *Empirical Risk Minimization* (ERM) suggests to minimize the empirical error, $\mathrm{Err}_{\mathcal{D}}(\cdot)$ where $\mathcal{D} := (x_i, y_i)_{1 \leq i \leq n}$ is a set of Independent and Identically Distributed set of data sampled from p. The ERM principle is backed with strong theoretical guarantee that we review in the following. To this purpose, we introduce the definition of an *hypothesis class* $\mathcal{H}$ which is any subset of measurable functions from $\mathcal{X}$ to $\mathcal{Y}$.

Definition 3.1.2 (Empirical Risk Minimization). *Let an hypothesis space $\mathcal{H}$, $p \in \mathcal{P}$, $n \in \mathbb{N}^\star$ and $\mathcal{D} := (x_i, y_i)_{1 \leq i \leq n}$ a set of n IID realizations of p. Empirical Risk Minimization consists in minimizing:*

$$h^\star := \arg\min_{h \in \mathcal{H}} \widehat{\mathrm{Err}}_{\mathcal{D}}(h)$$

One can note that the bigger the hypothesis space $\mathcal{H}$, the smaller will be the resulting empirical error $\widehat{\mathrm{Err}}_{\mathcal{D}}(h^\star)$. However, if some hypothesis h achieves a small empirical error, does it always guarantee a small error $\mathrm{Err}_p(f)$? Addressing this question is a central issue in Machine Learning and refers to as the problem *overfitting* that we describe in Figure 2.7 of Chapter 2.

For instance, consider $\mathcal{H}$ to be exactly the set of measurable functions and $\tilde{Y}$ a random variable $\mathcal{Y}$ independent of Y, the following hypothesis $\hat{h}$ achieves a null empirical error;

$$\hat{h} : x \mapsto \begin{cases} \mathcal{D}(x) & \text{if } x \in \mathcal{D} \\ \tilde{y} \sim \tilde{Y} & \text{otherwise} \end{cases} \tag{3.5}$$

where we note $\mathcal{D}(x)$ the corresponding label y for $x \in \mathcal{D}$, *i.e.* $\forall (x, y) \in \mathcal{D}, \mathcal{D}(x) = y$, and $\tilde{y}$ a realization of $\tilde{Y}$. It is straightforward to observe that $\widehat{\mathrm{Err}}_{\mathcal{D}}(\hat{h}) = 0$ while $\mathrm{Err}_p(\hat{h}) = \frac{1}{2}$ since $\mathbb{P}(\mathcal{D}) = 0$, *i.e.* $\hat{h}$ does not perform better than random choice. Building such example where ERM fails to address the objective of minimizing $\mathrm{Err}_p(\cdot)$ relies on the fact that $\mathcal{H}$ is too large. Indeed, between two close points in $\mathcal{D}$, which we note without loss of generality x_1 and x_2, the label transition from y_1 to y_2 is random while one can believe the underlying solution is regular enough to guarantee an efficient interpolation. It naturally brings to the notion of model complexity; quantifying the complexity of an hypothesis space $\mathcal{H}$ has profoundly

influenced learning theory. The prominent tool is the Vapnik–Chervonenkis dimension of an hypothesis class, noted $\mathrm{VC}(\mathcal{H})$ that is equal to the maximal number of samples with arbitrary labels an hypothesis class is able to fit.

Definition 3.1.3 (Vapnik-Chervonenkis (VC) dimension)**.** *A (binary) hypothesis class $\mathcal{H}$ is said to shatter a set of data points $\mathcal{D} := (x_1, \cdots, x_n)$ if, for all assignments of labels to those points, there exists a $h \in \mathcal{H}$ such that $\widehat{\mathrm{Err}}_{\mathcal{D}}(h) = 0$. The VC dimension of $\mathcal{H}$, noted $\mathrm{VC}(\mathcal{H})$ is the maximum number of points that can be arranged so that $\mathcal{H}$ shatters them.*

Equipped with a tool that quantifies the model complexity, one can show the consistency of the ERM principle;

Theorem 3.2 (Consistency of ERM)**.** *For any $\delta \in (0, 1]$, $p \in \mathcal{P}$ and $\mathcal{D}$ a set of n IID realization of p, with probability at least $1 - \delta$, the following holds*

$$\forall h \in \mathcal{H}, \quad \mathrm{Err}_p(h) \leq \widehat{\mathrm{Err}}_{\mathcal{D}}(h) + \sqrt{\frac{4}{n}\left(\mathrm{VC}(\mathcal{H}) \log\left(\frac{2en}{\mathrm{VC}(\mathcal{H})}\right) + \log\frac{4}{\delta}\right)} \tag{3.6}$$

This inequality shows that if provided with enough training samples *i.e.* high n, and a sufficiently constrained hypothesis class *i.e.* $\mathrm{VC}(\mathcal{H})$ is finite, ERM is consistent with the underlying objective of minimizing Err_p. We draw important remarks from the bound of Theorem 3.2;

(i) The bound is *distribution-free i.e.* it holds for any $p \in \mathcal{P}$. It implies that we consider the worst case distribution. Thus, the bound can be drastically reduced with appropriate assumption about the data distribution.

(ii) The bound grows with $\mathrm{VC}(\mathcal{H})$. Thus, the bound can be very loose for simple models. Note that many classes of models may not have a finite VC dimension *e.g.* $k-$nearest neighbors.

(iii) The result holds only if $\mathcal{D}$ is an IID set of realizations from p which is far from real-world applications, as presented in Chapter 2.

Addressing (*i*) and (*ii*) has motivated further tools for measuring the complexity of an hypothesis space, such as the *Rademacher complexity* that quantifies the ability of an hypothesis class to fit random labels, as well as the development of various inductive bias to constraint the hypothesis space. Going beyond the IID assumptions (*iii*) is the main subject of investigation of the present work.

3.1.3 Structural Risk Minimization and Regularization

The VC dimension increases with the size of an hypothesis class, *i.e.* for $\mathcal{H}_1 \subset \mathcal{H}_2$, $\mathrm{VC}(\mathcal{H}_1) \leq \mathrm{VC}(\mathcal{H}_2)$. Noting $h_i^\star := \arg\min_{h \in \mathcal{H}_i} \widehat{\mathrm{Err}}_{\mathcal{D}}(h)$ for $i = \{1, 2\}$, the theoretical bound from Theorem 3.2 indicates that if $h_1^\star$ achieves an equally small empirical error than $h_2^\star$ on $\mathcal{D}$, one should prefer the model that comes from the simplest hypothesis class, *i.e.* h_1, since $\mathrm{VC}(\mathcal{H}_1) \leq \mathrm{VC}(\mathcal{H}_2)$. In other hand, it is an instantiation Occam's razor (the parsimony principle) introduced in Chapter 2; among models that achieve an equally small error, one should consider the simpler one.

Structural Risk Minimization (SRM) (Vapnik and Chervonenkis 1971) generalizes this principle to a sequence of growing hypothesis classes $(\mathcal{H}_n)_n$, *i.e.* $\mathcal{H}_n \subset \mathcal{H}_{n+1}$. For $n \in \mathbb{N}^\star$ and noting $h_i^\star := \arg\min_{h \in \mathcal{H}_i} \widehat{\mathrm{Err}}_{\mathcal{D}}(h)$, if it exists n_0 such that for all $n \geq$

n_0, $\widehat{\text{Err}}(h^\star_{n_0}) = \widehat{\text{Err}}(h^\star_n)$, then one should choose $h^\star_{n_0}$ as the best hypothesis. The problem with sequential hypothesis classes can be framed through a penalized objective;

$$h^\star := \arg \min_{h \in \mathcal{H}_i,\ i \in \mathbb{N}} \widehat{\text{Err}}_{\mathcal{D}}(h) + \text{Penalty}(\mathcal{H}_i) \tag{3.7}$$

where $\text{Penalty}(\mathcal{H})$ is a proxy of the complexity of the hypothesis space $\mathcal{H}$. Thus, Structural Risk Minimization frames the learning problem as a trade-off between achieving a low empirical risk and exhibiting the simplest model.

However, measuring the complexity of an hypothesis class is not trivial. To address this issue, we usually penalize the function itself in contrast to the whole hypothesis class, leading to the following objective; for some fixed hypothesis class $\mathcal{H}$,

$$h^\star := \arg \min_{h \in \mathcal{H}} \widehat{\text{Err}}_{\mathcal{D}}(h) + \text{Regularization}(h) \tag{3.8}$$

where $\text{Regularization}(h)$ penalizes the complexity of the hypothesis h. The choice of an appropriate regularization is a difficult problem by itself. A well-adopted approach consists to measure the complexity of an hypothesis h expressed as a set of parameters $\theta \in \Theta$, through the norm of θ;

$$\text{Regularization}(h) = \lambda \cdot ||\theta|| \tag{3.9}$$

where $\lambda > 0$ trades-off the strength of the regularization in the learning objective of Equation 3.8. For instance, in the particular case where h is a linear regressor, the choice of the norm $||\cdot|| = ||\cdot||_1$ falls into the problem of Lasso regression while the choice of $||\cdot|| = ||\cdot||_2$ falls into the problem of Ridge regression.

3.1.4 Generalization, Inductive, Transductive and Semi-Supervised Learning

Evaluating Generalization

We have formulated learning as minimizing an error $\text{Err}_p(\cdot)$, computed over an infinite number of independently generated data from an unknown distribution p, from a finite number n of independently generated data from that same distribution $\mathcal{D} = (x_i, y_i)_{1 \leq i \leq n}$, called the training data. Concretely, learning exhibits a function h from a hypothesis class $\mathcal{H}$ that achieves a small error on p, *i.e.* small $\text{Err}_p(h)$. In the previous section, we showed the consistency of the ERM by relating it to the complexity of the hypothesis class, through its VC dimension $\text{VC}(\mathcal{H})$, and the training sample size n.

The concept of generalisation relates to the ability of a function from the hypothesis to achieve a low error on the whole underlying distribution. We usually represent the concept of generalisation by plotting the error against the hypothesis class complexity, thus obtaining a U-shaped curve. The left side of U characterises the underfitting situation, while the right side of U characterises the overfitting situation, that we have also illustrated in Figure 2.7. The minimum of the U-curve is the optimal situation where we reach a minimal error[1]. However, evaluating the generalisation

[1]Note that this interpretation fails to explain the generalisation of deep learning models, where we observe a double-U curve, phenomenon refered to as double descent where we observe an improving generalization in the over-parametrized regime. See the works (Belkin et al. 2019; Mei and Montanari 2019; Geiger et al. 2020; Hastie et al. 2019) reviewed in (Bach 2021).

of a hypothesis remains hard in practice. Indeed, it involves access to an infinite IID sample from p, which is not available in practice. To circumvent this issue, we usually separate at random the data $\mathcal{D}$ into two datasets: the training dataset $\mathcal{D}_{\text{train}}$ and the testing dataset $\mathcal{D}_{\text{test}}$. Then, we exhibit the best function on the training data and compute an approximation of generalisation on the test data. Crucially, learning consists of determining a model that provides good predictions on data that has never been seen during training.

Inductive *v.s.* Transductive Learning

In the mid-1970, Vapnik came up with an original idea; if we are interested in performance on the test dataset $\mathcal{D}_{\text{test}}$, why not simply providing a prediction on this dataset, rather than constructing a function that minimises $\text{Err}_p(\cdot)$ (Vapnik 2013)? Does it make learning a more manageable problem? This consideration has motivated the opposition between two learning paradigms; *inductive* learning, *i.e.* building a hypothesis that predicts any sample from p, *v.s. transductive* learning, predicting a predefined set of samples, here the test set $\mathcal{D}_{\text{test}}$. We provide a formal definition of transductive learning;

Definition 3.1.4 (Transductive Learning)**.** *Let a distribution $p \in \mathcal{P}$ and U, such that for any labelled dataset $(x_j, y_j)_j$, U returns the unlabelled dataset $(x_j)_j$, formally $\mathsf{U}((x_j, y_j)_j) = (x_j)_j$. The problem of Transductive Learning consists in minimizing the empirical error:*

$$h^\star := \arg\min_{h \in \mathcal{H}} \widehat{\text{Err}}_{\mathcal{D}_{\text{test}}}(h) \tag{3.10}$$

provided with:

- $\mathcal{D} = (x_i^\ell, y_i^\ell)_{1 \leq i \leq n_\ell}$*, a set of n_ℓ IID realizations from the distribution $p(X, Y)$ called the labelled dataset.*
- $\mathcal{U} := \mathsf{U}(\mathcal{D}_{\text{test}})$*, where $\mathcal{D}_{\text{test}}$ is a set of n_u IID realizations from the distribution $p(X, Y)$ called the test dataset. We note $\mathcal{U} =: (x_j^u)_{1 \leq j \leq n_u}$ the unlabelled dataset.*

As announced, transductive learning focuses on minimizing the error $\widehat{\text{Err}}_{\mathcal{D}_{\text{test}}}(\cdot)$, not $\text{Err}_p(\cdot)$ which is the interest of inductive learning. Crucially, transductive learning assumes that inputs x_j^u of $\mathcal{D}_{\text{test}}$ are available during learning. Pragmatically, one can frame the fundamental difference between transductive and inductive learning as follows; the inductive paradigm provides prediction on each samples of the test set independently from other predictions, while transductive learning predicts on the test set as a whole. A line of research has investigated in depth this field referred to as transductive inference, developing both theoretical insights of the superiority of transductive inference and algorithms (Vapnik and Sterin 1977; Chapelle, Vapnik, and Westontt 2000; Sinz et al. 2007; Gammerman, Vovk, and Vapnik 2013), such as the Transductive Support Vector Machine (Vapnik 2013). Interestingly, transductive learning has been the subject of active discussion within the community, as reported in Chapter 25 of (Chapelle, Scholkopf, and Zien 2009). As one of the researcher participating to the discussion guessed;

> "[...]. *I am convinced that in ten years the concept of noninductive inference will be much more popular than inductive inference.* [...]."
> From researcher B during a fictitious discussion inspired by real discussions that took place during the edition of (Chapelle, Scholkopf, and Zien 2009). See Chapter 25 of (Chapelle, Scholkopf, and Zien 2009).

Even if we can not say that transductive inference is now more popular than inductive inference, it is true that it has gained a lot of interest. For instance, transductive learning is now a central aspect of *Semi-Supervised Learning* (SSL) that we introduce in the following, Unsupervised Domain Adaptation (UDA) and the novel paradigm of Test-Time Adaptation (TTA), that will be subjects of investigation of this thesis.

Semi-Supervised Learning

Transductive learning is driven by the idea that minimising the error on a defined dataset $\mathcal{D}_{\text{test}}$ simplifies *de facto* the learning problem. In the transductive paradigm, the knowledge of the dataset on which the model is evaluated takes the shape of an unlabelled dataset $\mathcal{U} := (x_j^u)_j$, *i.e.* knowing the inputs on which the model will be evaluated. We can also formulate further speculation as to the interest of having an unlabelled dataset for learning; it provides fruitful information about the underlying distribution p. This seminal idea lays the foundation for Semi-Supervised Learning (SSL) (Zhou et al. 2004; Chapelle, Scholkopf, and Zien 2009), an influential field in ML. As described by Chapelle et al. in (Chapelle, Scholkopf, and Zien 2009);

> "*Semi-supervised learning (SSL) is halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information – but not necessarily for all examples. Often, this information will be the targets associated with some of the examples.*"
> (Unformal) Definition of SSL from Chapelle et al. (Chapelle, Scholkopf, and Zien 2009)

Let us provide a formal definition of Semi-Supervised Learning;

Definition 3.1.5 (Semi-Supervised Learning (SSL))**.** *Let a distribution $p \in \mathcal{P}$ and a hypothesis space $\mathcal{H}$. The problem of Semi-Supervised Learning (SSL) consists in minimizing the error:*

$$h := \arg\min_{h \in \mathcal{H}} \text{Err}_p(h) \tag{3.11}$$

provided with:

- $\mathcal{D} = (x_i^\ell, y_i^\ell)_{1 \leq i \leq n_\ell}$*, a set of n_ℓ IID realizations from the distribution $p(X, Y)$ called the labelled dataset.*
- $\mathcal{U} = (x_j^u)_{1 \leq j \leq n_u}$*, a set of n_u IID realizations from the distribution $p(X)$ called the unlabelled dataset.*

Similarly to transductive learning, SSL assumes access to a set of unlabelled samples $\mathcal{U}$ during learning. In contrast, SSL focuses on minimizing the error $\text{Err}_p(\cdot)$, *i.e.* on the whole distribution p. Thus, SSL, in its more general form, is inductive but with side information concerning the distribution p taking the shape of a set of unlabelled samples $\mathcal{U}$. SSL is often followed with appropriate assumptions about the distribution p (Chapelle, Scholkopf, and Zien 2009);

- **Smoothness assumption:** *If two points x_1, x_2 reside in a high-density region are close, then so should be their corresponding outputs y_1, y_2* (Chapelle, Scholkopf, and Zien 2009). High density regions, *i.e.* high $p(x)$, characterize clusters of inputs data, thus, data in the same cluster shall have the same label. Conversely, if inputs data are separated by a low density region, *i.e.* low $p(x)$, they are likely to be from different classes.

- **Cluster assumption:** *If points are in the same cluster, they are likely to be of the same class* (Chapelle, Scholkopf, and Zien 2009). This is a particular case of the smoothness assumption where inputs data is organized in well-separated clusters, which is typically useful for a classification task. The cluster assumption supports method based on model consistency through data augmentation that preserves the class, resulting to a decision boundary located in low density region of inputs data, *i.e.* low $p(x)$.
- **Manifold assumption:** *The (high-dimensional) data lies (roughly) in a low-dimensional manifold* (Chapelle, Scholkopf, and Zien 2009). To adress the curse of dimensionality, the manifold assumption assumes it exists a low dimensional representation of the data that is sufficient to solve the learning task.

Semi-Supervised Learning has been the object of a substantial literature reviewed in (Chapelle, Scholkopf, and Zien 2009), and (Ouali, Hudelot, and Tami 2020) for deep learning approaches for SSL.

Discussion

Semi-Supervised Learning faces to the same problem than standard inductive learning; it needs to evaluate generalization of the learner on a test dataset $\mathcal{D}_{\text{test}}$ of samples that have not been seen during learning. The particular case where $\mathsf{U}(\mathcal{D}_{\text{test}}) = \mathcal{U}$ is refered to as transductive SSL, but note that it is a very similar setup than transductive learning described by Vapnik et al. (Vapnik 2013). We refer to the interesting discussion from Chapter 24 and 25 from (Chapelle, Scholkopf, and Zien 2009). However, in the theoretical analysis of transductive learning in (Vapnik 2013), Vapnik et al. raise an interesting point where the theoretical guarantee holds even if $\mathcal{D}_{\text{test}}$ and the training data are not sampled from the same distribution. This challenging setup will be described in depth in the next section.

3.2 Learning from different distributions

3.2.1 Motivations

Empirical Risk Minimization (ERM) is a general principle for learning from a finite set of samples. We have presented the ubiquitous problem of overfitting that has motivated the principle of *Regularized Risk Minimization* introduced in Equation 3.8. However, ERM as well as SRM and Regularized Risk Minimization are based on the IID assumption, that is the data is an independent and identically distributed sample from the the targeted distribution. We discussed in Chapter 2 that this assumption might appear unrealistic in many real applications, thus motivating our interest in understanding the downside of the IID assumption. In this section, we present the problem of *learning from different distributions* where the IID assumption does not hold anymore. This problem is characterized by two distributions. The former, named the *source distribution* and noted $p_S(X, Y)$, is the distribution from which we obtain a sample of data. The latter, named the *target distribution* and noted $p_T(X, Y)$, is the distribution from which we aim to minimize the error of prediction. We note $\mathcal{D}_S := (x_i^S, y_i^S)_{1 \leq i \leq n_S}$ (respectively $\mathcal{D}_T := (x_j^T, y_j^T)_{1 \leq j \leq n_T}$) a set of n_S (respectively n_T) IID realizations from the source distribution $p_S(X, Y)$ (respectively the target distribution $p_T(X, Y)$). We note $\mathbb{E}_D[\cdot] = \mathbb{E}_{p_D}[\cdot]$ and $\text{Err}_T(h) = \mathbb{E}_D[\mathbb{I}(Y \neq h(X))]$ for $D \in \{S, T\}$. We provide a formal statement of the problem of learning under distribution shift;

Definition 3.2.1 (Learning under distribution shift)**.** *Let a source distribution $p_S \in \mathcal{P}$ and a target distribution $p_T \in \mathcal{P}$. The distribution shift situation is characterized by $p_S \neq p_T$. Given an hypothesis space $\mathcal{H}$, learning under distribution shift consists in minimizing the target risk:*

$$h_T := \arg\min_{h \in \mathcal{H}} \mathrm{Err}_{p_T}(h) \tag{3.12}$$

provided with a set $\mathcal{D}_S$ of IID realizations from the source distribution p_S called the source dataset.

The main difficulty with this framework lies in the difference between the two distributions, which no longer provides the consistency guarantee of empirical risk minimisation. We start by an introducing example of such a difficulty that demonstrates that theoretical guarantee from Theorem 3.2 vanishes in the context of distribution shift.

Example 3.2.1 (Linear Regression under Covariate Shift)**.** *Let $\mathcal{H}$ the hypothesis class of linear regressor, i.e. $h \in \mathcal{H}$ if it exists $\theta := (w, b)$ such that $f_\theta(x) := x^\top w + b$. Let $\mathcal{D} := (x_i, y_i)_{1 \leq i \leq n}$ a training dataset obtained by IID sampling of the following distribution:*

$$\begin{cases} X \sim & \mathcal{N}(-1.5, 1.) \\ Y|X \sim & \mathcal{N}\left(f(X), 0.1\right), \quad \textit{where} \quad f(X) := X \exp\left(-\frac{X}{5}\right) \end{cases} \qquad \text{(Source distribution)}$$

We assume that the target distribution is obtained by shifting the source distribution of X, now $X \sim \mathcal{N}(-0.25, 1)$ while $Y|X$ is conserved, which is a typical case of covariate shift. The example is detailed in Figure 2.10.

Nevertheless, we can already formulate an intuitive hypothesis; the more similar the distributions are, the more we will benefit from the theoretical guarantees of ERM. In contrast, the more different the source and the target distributions are, the more difficult it will be to draw interesting conclusions about the target distribution from a sample of the source distribution. The notion of similarity will be our particular focus in the next section.

3.2.2 Unsupervised Domain Adaptation

In the following, we present the problem of *Unsupervised Domain Adaptation* that aims to identify similarity between the source and the target distributions through the access to unlabelled data from the target distribution;

Definition 3.2.2 (Unsupervised Domain Adaptation (UDA))**.** *Let a source distribution $p_S \in \mathcal{P}$ and a target distribution $p_T \in \mathcal{P}$. Given an hypothesis space $\mathcal{H}$, the problem of Unsupervised Domain Adaptation (UDA) consists in minimizing the target risk:*

$$h_T := \arg\min_{h \in \mathcal{H}} \mathrm{Err}_{p_T}(h) \tag{3.13}$$

provided with:

- *$\mathcal{D}_S = (x_i^S, y_i^S)_{1 \leq i \leq n_S}$, a set of n_S IID realizations from the source distribution $p_S(X, Y)$ called the source dataset.*
- *$\mathcal{D}_T^u = (x_j^T)_{1 \leq j \leq n_T}$, a set of n_T IID realizations from the target distribution $p_T(X)$ called the target (unlabelled) dataset.*

Unsupervised Domain Adaptation is then an instance of problems of learning under distribution shift, but with side information since it assumes that a set of unlabelled samples in the target domain ($\mathcal{D}_T^u$) is available during learning. Note that this setup is much more challenging than the setup where we assume a set of labelled samples in the target domain is available during learning. Indeed, the latter problem is more similar to the standard setup of ERM (since we have access to an IID sample from the target distribution) or *Transfer Learning* (TL) in context of little data. As presented in Chapter 2, access to a set of unlabelled samples is a reasonable assumption since this data is typically much cheaper to acquire than labelled data from the target distribution.

3.2.3 Importance Sampling, a simple but not sufficient approach

Motivations

We present a particular case of distribution similarity that relies on the overlap of the source and the target distributions. To this purpose, we introduce the *support* of a distribution; for a distribution p on a set $\mathcal{U}$, we note $\text{Supp}(p)$ the support of p defined as $\text{Supp}(p) := \{u \in \mathcal{U}, p(u) > 0\} \subset \mathcal{U}$. In the following, we consider an important case where the target distribution is *included* into the source distribution, *i.e.* $\text{Supp}(p_T) \subset \text{Supp}(p_S)$. We address the case where this assumption does not hold anymore in the Section 3.2.4. When the target distribution is included into the source distribution, given $h \in \mathcal{H}$, one can express the target error as a source error with a well-suited importance given to the source data, a strategy called *Importance Sampling* or *Importance Weighting* (Quinonero-Candela et al. 2009);

$$\begin{aligned}
\text{Err}_T(h) :&= \mathbb{E}_T\left[\mathbb{I}(Y \neq h(X))\right] && (3.14)\\
&= \int_{x,y} \mathbb{I}(y \neq h(x)) p_T(x,y) \mathrm{d}(x,y) && (3.15)\\
&= \int_{x,y} \mathbb{I}(y \neq h(x)) \frac{p_T(x,y)}{p_S(x,y)} p_S(x,y) \mathrm{d}(x,y) && (3.16)\\
&= \mathbb{E}_S\left[\mathbb{I}(Y \neq h(X)) \frac{p_T(X,XY)}{p_S(X,Y)}\right] && (3.17)
\end{aligned}$$

Note that Equation 3.16 relies on $p_T(x,y) := \frac{p_T(x,y)}{p_S(x,y)} p_S(x,y)$, that is a licit operation under the assumption that the target distribution is included into the source distribution, indeed $\forall (x,y), p_T(x,y) \neq 0 \Rightarrow p_S(x,y) \neq 0$. This shows the target error $\text{Err}_T(h)$ is equal to a source error (*i.e.* involves an expectation over source sample wise error $\mathbb{I}(Y \neq h(X))$) as long as we carefully weight the contribution of source instances (x,y) into the expectation by a factor $w(x,y) := \frac{p_T(x,y)}{p_S(x,y)}$. Thus, if provided with a source dataset $\mathcal{D}_S := (x_i^S, y_i^S)_{1 \leq i \leq n_S}$, *i.e.* a set of n_S IID realizations from $p_S(X,Y)$, one can approximate the target error as follows;

$$\text{Err}_T(h) \approx \frac{1}{n_S} \sum_{i=1}^{n_S} w(x_i^S, y_i^S) \mathbb{I}(y_i^S \neq h(x_i^S)) := \widehat{\text{Err}}_{w \cdot S}(h) \tag{3.18}$$

where we use the notation $w \cdot S$ to emphasize the weighting of source instances of $\mathcal{D}_S$ through weights w. This analysis leads us to formulate two questions.

1. One can obtain an approximation of the error as long as it is enabled to compute weights $w(x,y)$ where $(x,y) \in \text{Supp}(p_T)$. *How hard is to compute $w(x,y)$?*

2. We have shown in Section 3.1.2 that the convergence speed in ERM is related to the complexity of the hypothesis class and the number of available training data. *Does the introduction of weights in the summation has an impact on this speed?*

We will see in the following that the former question is related to the inductive bias about the distribution shift, *i.e.* what reasonable assumptions can be made about the shift from the source to the target distribution, that allows to drastically reduce the complexity of weights computation. The latter question is a classical sample complexity analysis that takes into account weights w.

On the difficulty of estimating weights

In this section, we address the first question about the difficulty of computing the weighting $w(x, y)$ of source instances. The importance given to source instance, through the weight $w(x, y) = p_T(x, y)/p_S(x, y)$ for $(x, y) \in \text{Supp}(p_T)$, involves the target distribution $p_T(x, y)$ which is unknown when learning under distribution shift. However, the chain rule of probability provides fruitful insights;

$$w(x, y) = \frac{p_T(x, y)}{p_S(x, y)} = \frac{p_T(y|x)p_T(x)}{p_S(y|x)p_S(x)} = \frac{p_T(x|y)p_T(y)}{p_S(x|y)p_S(y)} \tag{3.19}$$

As presented in Section 2.2.3, by assuming that only one factor shift between the source and the target distribution of the conditional law or the marginal (Quinonero-Candela et al. 2009), one can distinguish four cases;

- **Covariate Shift:** $p_T(y|x) = p_S(y|x)$, then $w(x, y) = \frac{p_T(x)}{p_S(x)}$.
- **Label Shift:** $p_T(x|y) = p_S(x|y)$, then $w(x, y) = \frac{p_T(y)}{p_S(y)}$.
- **Conditional Shift:** $p_T(y) = p_S(y)$, then $w(x, y) = \frac{p_T(x|y)}{p_S(x|y)}$.
- **Concept Shift:** $p_T(x) = p_S(x)$, then $w(x, y) = \frac{p_T(y|x)}{p_S(y|x)}$.

Covariate shift and label shift involve both shift of marginals, X and Y respectively, while conditional shift and concept shift involve shift of conditional, $X|Y$ and $Y|X$ respectively.

Interestingly, the scenario of covariate shift does not involve the knowledge of labels in the target domain for computing weight, since here $w(x, y) = p_T(x)/p_S(x)$. As a result, this scenario is particularly well-suited for Unsupervised Domain Adaptation, that assumes access to an unlabelled dataset $\mathcal{D}_T^u$ from the marginal target distribution $p_T(X)$. Thus, UDA in a context of covariate shift is then reduced to compute $\hat{w}(x)$ for $x \in \text{Supp}(p_S(X))$, an estimation of the distribution ratio $p_T(x)/p_S(x)$ $x \in \text{Supp}(p_S(X))$ obtained from two datasets drawn from the source distribution $p_S(X)$ and $p_T(X)$ respectively. Once $\hat{w}$ is computed, learning simply consists in minimizing the weighted source error;

$$h^\star := \arg\min_{h \in \mathcal{H}} \frac{1}{n_S} \sum_{i=1}^{n_S} w(x_i^s) \mathbb{I}(h(x_i^S) \neq y_i^S) \tag{3.20}$$

The scenario of covariate shift for UDA, also referred to as *covariate shift adaptation*, has produced a substantial literature (Quinonero-Candela et al. 2009; Gretton et al.

2009; Sugiyama, Krauledat, and MÃžller 2007; Sugiyama et al. 2008; Li, Lam, and Prusty 2020).

The scenario of label shift for UDA, also refered to as *label shift adaptation*, only recquires to estimate the class proportion in the target domain. Note that it is far from a trivial problem since we do not have access to target labels in a scenario of UDA. Nevertheless, the literature has produced positive theoretical results to tackle this problem (Blanchard, Lee, and Scott 2010; Scott, Blanchard, and Handy 2013; Sanderson and Scott 2014). More recently, Lipton et al. has shown that it was possible to infer target classes proportion through a black-box model in a scenario of UDA, simply from the confusion matrix in the source domain and the black-box prediction on the unlabelled target data (Lipton, Wang, and Smola 2018).

Concept shift and conditional shift are both much more challenging scenario. Indeed, they both require knowledge about labels in the target domain, that are absent in the standard setup of UDA, through the shift of conditionals, $Y|X$ and $X|Y$ respectively[2]. Thus, addressing concept and conditional shifts involves some knowledge about the coupling between x and y. In particular, it is not necessary to know the pairing between a realization x and its label y in the target domain in a scenario of label shift. Importance Sampling is not the prominent tool to address this challenging scenario of distribution shift. Note that pioneering works address scenario of concept shift and conditional shift, (Wang, Huang, and Schneider 2014) and (Zhang et al. 2013) respectively, through location and scaling[3] of labels and inputs respectively, that is reviewed in Section 2.2 of (Redko et al. 2019).

Theoretical analysis

In this section, we provide theoretical insights about the impact of weighting instance in the source domain for approximating the error in the target domain. Our discussion focused on the particular scenario of covariate shift adaptation which is, as presented in the previous section, the most prominent approach for Unsupervised Domain Adaptation based on Importance Sampling. We recall that such assumption relies on the equality between labelling function across domains, *i.e.* $p_T(Y|X) = p_S(Y|X)$. We present main results from (Cortes, Mansour, and Mohri 2010) that we organize as follows; we consider the (idealistic) case where we can compute the exact ratio of distribution $w(x) = p_T(x)/p_S(x)$ then we analyse the case where we rely on a estimation $\hat{w}$ of the exact w. We present these important results in the case of the binary classification error, however, they remain valid under broader assumptions as presented in (Cortes, Mansour, and Mohri 2010).

In the idealistic case where the exact ratio of distribution $w(x) = p_T(x)/p_S(x)$, one can bound the target error in a very similar fashion than the theoretical bound for ERM presented in Theorem 3.2.

Proposition 3.2.1 (Cortes, Mansour, and Mohri 2010)**.** *For any $\delta \in (0, 1]$, $p_S, p_T \in \mathcal{P}$ such that* $\text{Supp}(p_T) \subset \text{Supp}(p_S)$ *and $\mathcal{D}_S$ a set of n IID realization of p_S, with probability*

[2]Pragmatically, it is related to the access to the pairing between target input x and the target label y. Note that it differs strongly from the underlying difficulty of label shift, which only requires knowledge about target labels through the target classes proportion.

[3]An operation that modifies the location and scale of an inputs *i.e.* a mapping $x \mapsto w \odot x + b$ for some w and b.

at least $1-\delta$*, the following holds*

$$\mathrm{Err}_T(h) \leq \widehat{\mathrm{Err}}_{w\cdot S}(h) + 2^{5/4}\sqrt{\mathbb{E}_S\left[w^2\right]}\ \sqrt[\frac{3}{8}]{\frac{1}{n_S}\left(\mathrm{VC}(\mathcal{H})\log\left(\frac{2en_S}{\mathrm{VC}(\mathcal{H})}\right)+\log\frac{4}{\delta}\right)} \tag{3.21}$$

where n_S *is the number of samples in the source domain and* $w(x) = p_T(x)/p_S(x)$*.*

To bound the target error $\mathrm{Err}_T(h)$, the empirical error from Theorem 3.2 is now replaced by the weighted source empirical error $\widehat{\mathrm{Err}}_{w\cdot S}(h)$ in Proposition 3.2.1. Sample complexity terms have both similar shape than in Theorem 3.2 and Proposition 3.2.1 while we note a shift in rate from $\mathcal{O}(n_S^{-1/2})$ to $\mathcal{O}(n_S^{-3/8})$. Crucially, the sample complexity term is scaled by a factor $\mathbb{E}_S[w^2]$ which embeds the similarity between the two distributions. Note that in the particular case where $p_S(x) = p_T(x)$, then $\mathbb{E}_S[w^2] = 1$. We now review the case where we rely on an estimation $\hat{w}$ of w, *i.e.* $\hat{w} \neq w$ for the general case.

Proposition 3.2.2 (Cortes, Mansour, and Mohri 2010)**.** *For a given* $h \in \mathcal{H}$*, there is for any* $\delta > 0$ *with at least probability* $1-\delta$*:*

$$\begin{aligned}\mathrm{Err}_T(h) \leq \widehat{\mathrm{Err}}_{w\cdot S}(h) &+ \left|\mathbb{E}_T\left[(w(X)-\hat{w}(X))\,\mathbb{I}(Y\neq h(X))\right]\right| \\ &+ 2^{5/4}\sqrt{W(h)}\ \sqrt[\frac{3}{8}]{\frac{1}{n_S}\left(\mathrm{VC}(\mathcal{H})\log\left(\frac{2en_S}{\mathrm{VC}(\mathcal{H})}\right)+\log\frac{4}{\delta}\right)}\end{aligned} \tag{3.22}$$

where $W(h) = \max\left\{\mathbb{E}_S\left[\hat{w}(X)^2\mathbb{I}(Y\neq h(X))\right], \frac{1}{n_S}\sum_{i=1}^{n_S}\hat{w}(x_i^S)^2\mathbb{I}(y_i^S \neq h(x_i^S))\right\}$ *where* n_S *is the number of samples in the source domain and* $w(x) = p_T(x)/p_S(x)$*.*

This new bound is within the same spirit than the bound from Proposition 3.2.1, the main difference is the introduction of a new term $\left|\mathbb{E}_T\left[(w(X)-\hat{w}(X))\,\mathbb{I}(Y\neq h(X))\right]\right|$ that reflects the error when using an approximation $\hat{w}$ instead of w. The term $W(h)$ in Proposition 3.2.1 has a very similar role than $\mathbb{E}_S[w^2]$ in Theorem 3.2.

Discussion

Importance Sampling is an appealing approach to address the problem of distribution shift. In some particular case, such as the scenario of Covariate Shift or Label Shift, there are both mature theoretical analysis and learning methods to mitigate the effect of distribution shift. However, Importance Sampling faces three important limitations;

1. It heavily relies on the situation of inclusion $\mathrm{Supp}(p_T) \subset \mathrm{Supp}(p_S)$. Pragmatically, this means that for a target sample (x^T, y^T) from p_T, it is likely to obtain a source sample $(x^S, y^S) \sim p_S$ in the neighborhood of (x^T, y^T). Such assumption fails to be met in real-world scenario involving high dimensional data such as text or images as suggested in (D'Amour et al. 2021).

2. The scenario of covariate shift or label shift are very restrictive. For instance, what is the meaning of the covariate shift situation when the inclusion assumption is not satisfied? Furthermore, even when the favorable assumption of covariate shift is met, it is not sufficient to guarantee a successful adaptation (Ben-David et al. 2010b).

3. Deep Learning models become ubiquitous for processing high dimensional data. Importance Sampling interacts poorly with over-parametrized models

(Byrd and Lipton 2019) urging the need to provide a novel theoretical understanding of adaptation.

3.2.4 A seminal theory

We present a theory for learning under distribution shift that still has a profound impact on the understanding on the underlying difficulty of adaptation. The theory has been first introduced by Ben-David et al. in (Ben-David et al. 2007) and consolidated in (Ben-David et al. 2010a). At the highest level, the theory can be summarized as follows;

$$\text{Target error} \leq \text{Source error} + \text{Discrepancy} + \text{Adaptability} + \text{Sample Complexity} \tag{3.23}$$

Each error term embeds a specific dynamic:

- **Source error:** quantifies if the model works correctly on the source distribution,
- **Discrepancy:** quantifies the discrepancy between marginals across domains $p_T(X)$ and $p_S(X)$,
- **Adaptability:** quantifies if it exists a model that works correctly on both source and target distribution. Thus, it embeds if the adaptation is possible at first.
- **Sample complexity:** quantifies the speed of convergence with respect to the number of samples and the complexity of the hypothesis class.

This theory has given rise to many variations, mainly based on the choice of the discrepancy measure used to quantify the difference between the source and the target distribution. A line of works focuses on defining the discrepancy that depends on the hypothesis class (Ben-David et al. 2010a; Cortes, Mansour, and Mohri 2010; Zhao et al. 2019; Zhang et al. 2019) while an other line provides a hypothesis class free analysis of the discrepancy based on *Integral Probability Measure* (IPM) with MMD (Redko 2015) or Wassertein distance for Optimal Transport (Redko, Habrard, and Sebban 2017; Courty et al. 2016; Courty et al. 2017). Notably, some works focus on generalizing the theory to other paradigms for sample complexity, such as Rademacher complexity (Mansour, Mohri, and Rostamizadeh 2009) or PAC-bayes complexity (Germain et al. 2013; Germain et al. 2016). We recommend the comprehensive overview presented in Table 2 of (Redko et al. 2020) for the different improvements and generalizations of the seminal theory from (Ben-David et al. 2010a).

We aim to provide a modern and straightforward exposition that focuses on describing the fundamental trade-off in adaptation, that is achieving a low *discrepancy* while ensuring a good *adaptability*. To this purpose, we do not follow the historical development of this theory. We recommend to the reader the thorough depiction of (Redko et al. 2019) for the exposition of the historical development of this theory.

To conduct the analysis, we introduce two additional tools: the labelling functions $f_D : x \mapsto \mathbb{E}[Y|X]$ and the disagreement between two hypotheses $(h, h') \in \mathcal{H}^2$: $\text{Err}_D(h, h') := \mathbb{E}_{p_D}[h(X) \neq h'(X)]$ for $D \in \{S, T\}$. In particular, one can observe that $\forall h \in \mathcal{H}, \text{Err}_D(h, f_D) \leq \text{Err}_D(h)$ for $D \in \{S, T\}$. For a given $h \in \mathcal{H}$, the founding theoretical works (Ben-David et al. 2007; Ben-David et al. 2010a) relate the target risk $\text{Err}_T(h)$ with the source risk $\text{Err}_S(h)$ as follows:

Theorem 3.3 (Ben-David et al. bound (Ben-David et al. 2007; Ben-David et al. 2010a))**.** *Let an hypothesis space $\mathcal{H}$ and $h \in \mathcal{H}$:*

$$\mathrm{Err}_T(h) \leq \mathrm{Err}_S(h) + \delta_{\mathcal{H}} + \lambda_{\mathcal{H}} \tag{3.24}$$

where $\delta_{\mathcal{H}} := \sup_{(h,'h)\in\mathcal{H}^2} \mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h')$ *and* $\lambda_{\mathcal{H}} := \inf_{h'\in\mathcal{H}} \mathrm{Err}_S(h') + \mathrm{Err}_T(h')$.

We describe the steps that lead to this important result for gaining insights about the role of each error terms $\mathrm{Err}_S(h)$, $\delta_{\mathcal{H}}$ and $\lambda_{\mathcal{H}}$. We underline that proof provided in the present work deviates slightly from the historical proof from (Ben-David et al. 2010a). In particular, we follow the major idea from (Mansour, Mohri, and Rostamizadeh 2009).

Proof. Let $h' \in \mathcal{H}$ an additional hypothesis, we apply the triangular inequality;

$$\mathrm{Err}_T(h) \leq \mathrm{Err}_T(h,h') + \mathrm{Err}_T(h') \tag{3.25}$$

The main idea is to introduce the error in the source domain by making it appears artificially in the inequality through the disagreement error $\mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h')$. By re-organizing terms, one can obtain;

$$\mathrm{Err}_T(h) \leq \mathrm{Err}_S(h,h') + \mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h') + \mathrm{Err}_T(h') \tag{3.26}$$

We bound the source error using the triangular inequality on f_S;

$$\mathrm{Err}_S(h,h') \leq \mathrm{Err}_S(h,f_S) + \mathrm{Err}_S(f_S,h') \leq \mathrm{Err}_S(h) + \mathrm{Err}_S(h'), \tag{3.27}$$

leading to;

$$\mathrm{Err}_T(h) \leq \mathrm{Err}_S(h) + \mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h') + \mathrm{Err}_S(h') + \mathrm{Err}_T(h') \tag{3.28}$$

First, $\mathrm{Err}_S(h)$ is the source error. From this perspective, we have succeeded in relating the target risk with the source risk. Second, we provide a deeper analysis of the two remaining terms *i.e.* $\mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h')$ and $\mathrm{Err}_T(h') + \mathrm{Err}_S(h')$, **Discrepancy** and **Adaptability** respectively, in order to break the dependence with h'.

(i) **Discrepancy.** $\mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h')$ exhibits an interesting behavior: the fact that $p_S = p_T$ is sufficient to vanish this term. Importantly, it holds independently of h and h', *e.g.* h and h' can be very different to labelling functions f_S and f_T. As a result, this term reflects the similarity between the source and the target distributions. Thus, we provide a pessimistic estimation of it with respect to h and h', *i.e.* we bound $\mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h')$ with the worst cases classifier $(h,h') \in \mathcal{H}^2$, bounding the target risk as follows:

$$\mathrm{Err}_T(h) \leq \mathrm{Err}_S(h) + \delta_{\mathcal{H}} + \mathrm{Err}_T(h') + \mathrm{Err}_S(h') \tag{3.29}$$

where:

$$\delta_{\mathcal{H}} = \sup_{(h,'h)\in\mathcal{H}^2} \mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h') \tag{3.30}$$

(ii) **Adaptability.** The remaining term which depends on h' is $\mathrm{Err}_S(h') + \mathrm{Err}_T(h')$. It reflects the performance of the fictional classifier h' on both the source and the target distributions. It is often referred to as a *combined error*. Contrary to $\mathrm{Err}_T(h,h') - \mathrm{Err}_S(h,h')$, this term can be small only if h' is close to both f_S and

f_T. Hopefully, the inequality holds for any $h' \in \mathcal{H}$, in particular for the one that achieves the smallest combined error, bounding the target risk as follows:

$$\text{Err}_T(h) \leq \text{Err}_S(h) + \delta_{\mathcal{H}} + \lambda_{\mathcal{H}} \tag{3.31}$$

where:

$$\lambda_{\mathcal{H}} := \inf_{h' \in \mathcal{H}} \text{Err}_S(h') + \text{Err}_T(h') \tag{3.32}$$

is referred to as the *Adaptability* of $\mathcal{H}$.

□

Crucially, the founding theory from (Ben-David et al. 2010a) has succeeded in relating the source risk and the target risk through the discrepancy of distributions, embodied by $\delta_{\mathcal{H}}$ that reflects the sensitivity of the hypothesis space with respect to changes from the source and the target distributions, and the similarity between labelling functions embodied by $\lambda_{\mathcal{H}}$.

A detailed view of $\delta_{\mathcal{H}}$

We provide additional insights about $\delta_{\mathcal{H}}$ since it has inspired an important line of works based on *adversarial* learning. We take a few step back and present the so-called $\mathcal{H}-$divergence (Ben-David et al. 2010a), based on Kifer, Ben-David, and Gehrke 2004):

Definition 3.2.3 ($\mathcal{H}$-divergence)**.** *Let $\mathcal{H}$ a hypothesis space, the $\mathcal{H}-$divergence, noted $d_{\mathcal{H}}$ is defined as:*

$$d_{\mathcal{H}} := 2 \sup_{h \in \mathcal{H}} |\mathbb{P}_T [h(X) = 1] - \mathbb{P}_S [h(X) = 1]| \tag{3.33}$$

We provide an alternative definition of $\mathcal{H}-$divergence with a similar shape than an Integral Probability Metric (IPM, (Müller 1997)):

$$d_{\mathcal{H}} := 2 \sup_{h \in \mathcal{H}} |\mathbb{E}_S[h(X)] - \mathbb{E}_T[h(X)]| \tag{3.34}$$

Proof. We show the equivalence between the two definitions. Since $\mathbb{I}(h(x) = 1) = h(x)$ because $h(x) \in \{0, 1\}$, we observe $\mathbb{P}(h(X) = 1) = \mathbb{E}[\mathbb{I}(h(X) = 1)] = \mathbb{E}[h(X)]$. □

We provide an interesting interpretation of $d_{\mathcal{H}}$ when $\mathcal{H}$ is symmetric. In particular, the $\mathcal{H}-$ divergence is related to the performance of a classifier trained to discriminate the source from the target distribution *i.e.* $h(x) = 1$ for $x \sim p_T$ while $h(x) = 0$ for $x \sim p_S$.

Proposition 3.2.3. *Let $\mathcal{H}$ a symmetric hypothesis space i.e. $h \in \mathcal{H} \Rightarrow 1 - h \in \mathcal{H}$, then:*

$$d_{\mathcal{H}} := 2 \left(1 - \min_{h \in \mathcal{H}} \{\mathbb{E}_S[1 - h(X)] + \mathbb{E}_T[h(X)]\} \right) \tag{3.35}$$

This result, proposed originally in (Ben-David et al. 2010a), is of the greatest importance since it has inspired an important line of works on adversarial learning. In the following, we relate the divergence of an hypothesis with $\delta_{\mathcal{H}}$. To this purpose, we recall that $\text{Err}_D(h, h') = \mathbb{E}_D[h(X) \neq h'(X)] = \mathbb{E}_D[h(X) \oplus h'(X)]$ where $\oplus$ is the XOR function.

Definition 3.2.4 (Symmetric difference of an hypothesis space (Ben-David et al. 2010a))**.** *Let $\mathcal{H}$ an hypothesis space. The symmetric difference hypothesis space of $\mathcal{H}$, noted $\mathcal{H}\Delta\mathcal{H}$, is the set of hypotheses:*

$$g \in \mathcal{H}\Delta\mathcal{H} \iff \exists (h, h') \in \mathcal{H}^2, g = h \oplus h' \tag{3.36}$$

As a result, $\delta_{\mathcal{H}}$ is related to $\mathcal{H}\Delta\mathcal{H}-$divergence by $\delta_{\mathcal{H}} = \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}$. It is worth noting that $\mathcal{H}\Delta\mathcal{H}$ is an non-intuitive space since $g \in \mathcal{H}\Delta\mathcal{H}$ requires two hypotheses of $\mathcal{H}$ to be computed. We circumvent this problem by observing that $\delta_{\mathcal{H}}$ involves a supremum over $\mathcal{H}\Delta\mathcal{H}$. We introduce the following set of discriminators.

Definition 3.2.5 (Discriminators of an hypothesis space)**.** *Let $\mathcal{H}$ an hypothesis space. A discriminator space of $\mathcal{H}$, noted $D_{\mathcal{H}}$, is any subset of $[0,1]^{\mathcal{X}}$ such that:*

$$\mathcal{H}\Delta\mathcal{H} \subset D_{\mathcal{H}} \tag{3.37}$$

The set of discriminators allows us to bound $\delta_{\mathcal{H}}$ based on the accuracy of a discriminator trained to differentiate the source from the target distribution.

Proposition 3.2.4. *Let $\mathcal{H}$ an hypothesis space and $D_{\mathcal{H}}$ a discriminator space of $\mathcal{H}$. Let $\mathcal{D}_S$ and $\mathcal{D}_T$, two set of IID realizations from p_S and p_T respectively. For the sake of simplicity, we assume $|\mathcal{D}_S| = |\mathcal{D}_T| = n$. For any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\delta_{\mathcal{H}} \leq \left(1 - \frac{1}{n} \min_{d \in D_{\mathcal{H}}} \left\{ \sum_{x \in \mathcal{D}_S} 1 - d(x) + \sum_{x \in \mathcal{D}_T} d(x) \right\}\right) + 2\sqrt{\frac{\mathrm{VC}\left(D_{\mathcal{H}}\right) \log(2n) + \log \frac{2}{\delta}}{n}} \tag{3.38}$$

Proof. First, we observe that $\delta_{\mathcal{H}} = \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}$. Second, $d_{\mathcal{H}\Delta\mathcal{H}} \leq d_{D_{\mathcal{H}}}$ since $\mathcal{H}\Delta\mathcal{H} \subset D_{\mathcal{H}}$. Finally, we apply Lemma 1 from (Ben-David et al. 2010a) leading to the stated result. □

In this section, we have related the error in the target domain with the error in the source domain, the discrepancy between the source and the target domains and the adaptability. Importantly, the source error and the discrepancy are tractable in a scenario of UDA, not the adaptability that involves target labels. We have elaborated on the expression of discrepancy $\delta_{\mathcal{H}}$ proving it is related to the accuracy of a binary classifier trained to separate the source from the target domains based on inputs, as presented in Proposition 3.2.4.

3.3 Learning Invariant Representations

3.3.1 Motivations

Learning representations has became a tool of choice in Machine Learning, especially since the celebration of *Deep Learning* (LeCun, Bengio, and Hinton 2015). In this section, we motivate the use of representations learned to address the problem of adaptation. We review the basics of the paradigm of learning representations with deep networks *i.e.* through the composition of non-linear functions of the inputs, in Appendix A. We provide fruitful insights about the role of representations to make the source and the target domains more similar.

In Section 3.2, we have related the risk of distribution shift with respect to the similarity between the source and the target distribution of inputs X, quantified by the $\mathcal{H}\Delta\mathcal{H}$-divergence, *i.e.* the sensitivity of the hypothesis class $\mathcal{H}$ to change in the input distribution $p(X)$. The seminal theory from Ben-David et al. introduces the ubiquitous term of *adaptability* (Ben-David et al. 2010a) defined as the minimal combined error, *i.e.* the sum between the source and the target errors achieved by the best classifier. Ben David et al. argue that adaptability embeds our capacity to perform adaptation at first; one can assume this adaptability to represent a small error term. Under this assumption, the adaptation problem is thus limited to reducing the divergence between distributions of inputs, namely by constraining the hypothesis class.

Although this vision is adopted by the vast majority of the literature, it is not sufficient to guarantee a successful adaptation. When the source and target distribution of inputs do not overlap (D'Amour et al. 2021), even a simple hypothesis class, such as LogisticRegressor, can separate the source from the target distribution, thus resulting to $\delta_{\mathcal{H}} = 1$. To overcome this issue, a line of studies focuses on learning representations of inputs, through a mapping φ such that $Z := \varphi(X)$ where Z is the representation of input X[4]. It aims to reconcile the two non-overlapping distributions by matching the source and target distributions of representations, resulting to a so-called *Invariant Representation* (Ben-David et al. 2007; Ganin and Lempitsky 2015). From this perspective, given a representation φ, the hypothesis $h \in \mathcal{H}$ becomes a composition between a representation φ and a classifier g from a set of classifiers $\mathcal{G}$;

$$h = g \circ \varphi, \tag{3.39}$$

i.e. $\mathcal{H} = \{g \circ \varphi; g \in \mathcal{G}\}$, that we note $\mathcal{H} = \mathcal{G}\varphi$. Ultimately, φ should be chosen to promote similarity between source and target representation, *i.e.* a small discrepancy between $p_S(\varphi(X))$ and $p_T(\varphi(X))$. Through an example, we present the fundamental trade-off that we are exposed when relying on a representation for Unsupervised Domain Adaptation;

Example 3.3.1 (Unit circle of the plan)**.** *Let π the uniform distribution on the unit circle of $\mathbb{R}^2$ centered in $(-1, 0)$ and with radius 1; we generate the source and the target data as follows;*

$$\begin{cases} Z & \sim & \pi \\ Y & \leftarrow & \mathbb{I}(Z_1 \leq Z_2) \\ X_S & \leftarrow & Z \\ X_T & \leftarrow & Z + (2, 0) \end{cases} \tag{3.40}$$

Let consider the class of logistic regressors as the hypothesis class $\mathcal{H}$. First, we observe that $\min_{h\in\mathcal{H}} \mathrm{Err}_S(h) = 0$, thus it is possible to achieve a null error on the source domain. Second, we can draw the same conclusion for the target distribution; i.e. $\min_{h\in\mathcal{H}} \mathrm{Err}_T(h) = 0$. Note that it does not imply that we can achieve a null combined error; since here $\lambda_{\mathcal{H}} \neq 0$. Additionally, since the source and target distributions are linearly separable, we have $\delta_{\mathcal{H}} = 1$. The fact that we can not achieve a null adaptability results from the fact that $\mathcal{H}$ has not enough capacity, thus, let consider a bigger hypothesis class;

$$\mathcal{H}_{\oplus} := \left\{x \mapsto \mathbb{I}(x_1 \leq 0)h(x) + \mathbb{I}(x_1 > 0)h'(x), (h, h') \in \mathcal{H}^2\right\} \tag{3.41}$$

to obtain $\lambda_{\mathcal{H}_{\oplus}} = 0$. However, it is straightforward to observe that $\mathcal{H} \subset \mathcal{H}_{\oplus}$, thus $\delta_{\mathcal{H}_{\oplus}} \geq \delta_{\mathcal{H}}$ leading to $\delta_{\mathcal{H}_{\oplus}} = 1$. Thus, the theory from Ben-David et al. (Ben-David et al. 2010a),

[4] We refer indifferently to φ and Z as representation(s).

introduced in section 3.2, can not address this problem of adaptation.

Nevertheless, the source and target distributions are related with a simple relation; $X_T = X_S + (2,0)$ *meaning that the target inputs are related to the source inputs by a translation. Leveraging this simple underlying relation, we apply Theorem 3.3 to a representation of the inputs* $\varphi(X)$*, defined as follows;*

$$\varphi : x \mapsto \begin{cases} x & \text{if} \quad x_1 \leq 0 \\ x - (2,0) & \text{otherwise.} \end{cases} \tag{3.42}$$

where it is worth noting that $\varphi(X) = Z$ *for* $X \sim p_S$ *or* $X \sim p_T$*. We note* $\mathcal{H}\varphi = \{h \circ \varphi, h \in \mathcal{H}\}$*, the hypothesis class obtained from representations. Similarly,* $\min_{h \in \mathcal{H}\varphi} \mathrm{Err}_S(h) = 0$ *and* $\min_{h \in \mathcal{H}\varphi} \mathrm{Err}_T(h) = 0$*. Interestingly, by applying it to the representation, we have* $\lambda_{\mathcal{H}\varphi} = 0$*. Furthermore, since* $p_S(\varphi(X)) = p_T(\varphi(X))$*, we have* $\delta_{\mathcal{H}\varphi} = 0$*. By working from a suitable representation and not from the inputs, the adaptation is a strong success since we achieve simultaneously a null source error, a null discrepancy and a null adaptability.*

However, we mention that simply finding φ *such that* $\delta_{\mathcal{H}\varphi} = 0$ *is not sufficient to guarantee adaptation. Indeed, let us consider the representations;*

$$\varphi_\theta : x \mapsto \begin{cases} x & \text{if} \quad x_1 \leq 0 \\ R_\theta(x - (2,0)) & \text{otherwise, where } R_\theta := \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \end{cases} \tag{3.43}$$

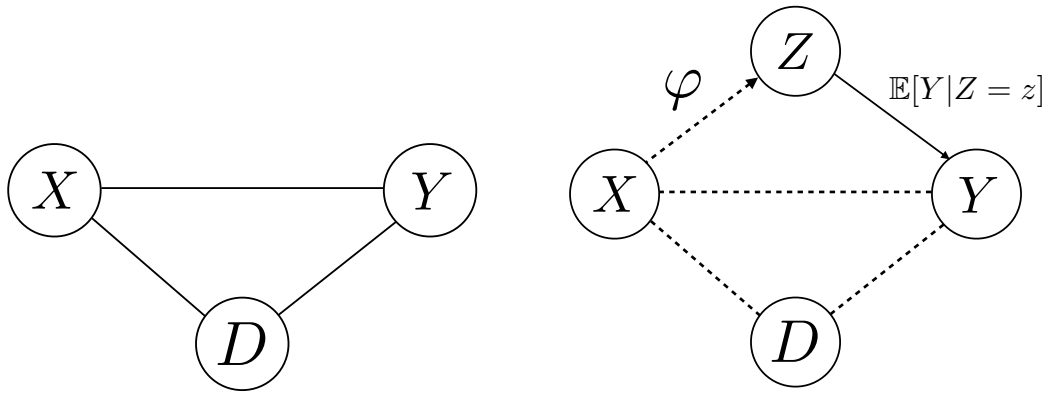
we can observe that $\forall \theta \in [0, 2\pi]$*,* $p_S(\varphi_\theta(X)) = p_T(\varphi_\theta(X))$ *thus* $\delta_{\mathcal{H}\varphi_\theta} = 0$*, but* $\lambda_{\mathcal{H}\varphi_\theta} = \frac{\theta}{2\pi}$*. The difficulty of finding an invariant representation while ensuring a good adaptability, i.e. a low* $\lambda_{\mathcal{H}\varphi_\theta}$*, is the heart of unsupervised domain adaptation.*

If it seems reasonable to believe that the adaptability error remains small when working in the input space, the use of a representation completely reshuffles the deck; one can find representation that achieves a low distribution discrepancy in the representation space, with an arbitrary high adaptability, as presented in Example 3.3.1.

Intuitively, the similarity between labelling functions $f_D : x \mapsto \mathbb{E}_D[Y|X = x]$ for $D \in \{S, T\}$ is crucial for achieving a low error combined error $\mathrm{Err}_S(\cdot)$ and $\mathrm{Err}_T(\cdot)$, *i.e.* a low adaptability error. Although one can reasonably believe labelling functions f_D are indeed similar, it may not be true when transforming the inputs X into Z through a representation φ, *i.e.* similarity between f_S and f_T does not guarantee a similarity between $\mathbf{f}_S$ and $\mathbf{f}_T$ where $\mathbf{f}_D : z \mapsto \mathbb{E}_D[Y|\varphi(X) = z]$. Thus, a suitable representation for adaptation shall verify the following equality ultimately;

$$\mathbb{E}_T[Y|\varphi(X)] = \mathbb{E}_S[Y|\varphi(X)] \tag{3.44}$$

We refer to this favorable property by calling *φ a representation that elicits an invariant predictor*, as defined in (Arjovsky et al. 2019). We provide a graphical overview of this kind of representation in Figure 3.1 where couple (X, Y) depends on a third variable D which embodies the domain; D can be either S or T. In particular, if φ elicits an invariant predictor, the relation that gives $Z \to Y$ does not depend on D; in particular $\mathbb{E}_T[Y|\varphi(X)] = \mathbb{E}_S[Y|\varphi(X)]$.



(a) A graphical overview of the inputs X and outputs Y and the nuisance factor D

(b) Representation φ eliciting an invariant predictor.

Figure 3.1: (a): A graphical overview of the inputs X and outputs Y that are related with a third variable D, called a *nuisance factor*. Note that the graph is not directed, then there is no assumption of conditional independence or the causality that is underlying X, Y and D. Modelling learning with three random variables (here X, Y and D) encapsulates several paradigms, including Domain Adaptation (D indicates the domain) or Fairness (D indicates the sensitive variable). (b): A representation φ is said to elicit an invariant predictor if $Y|Z$ is independent from D. Put simply, the nuisance factor may change the distribution of (X, Y) but the representation φ allows to identify a variable $Z := \varphi(X)$ such that $Y|Z$ is invariant *i.e.* is independent from the nuisance factor D.

3.3.2 Domain Invariant Representations

Theoretical Analysis

In addition to an input space $\mathcal{X}$ and an output space $\mathcal{Y}$, we assume there is a *representation* $\mathcal{Z}$ obtained through a mapping, called representation, $\varphi : \mathcal{X} \to \mathcal{Z}$. We refer indifferently to φ as a representation and the random variable $Z := \varphi(X)$. Note that we usually define φ as a deep neural network. In the previous section, we have motivated our interest into representations that reconcile two non-overlapping supports in the input space $\mathcal{X}$ to obtain overlapping distributions in the representation space $\mathcal{Z}$. In particular, we have exhibited an appealing property called *invariance*, for which we provide a formal statement;

Definition 3.3.1 (Domain Invariant Representations)**.** *A representation $\varphi \in \Phi$ is domain invariant if $\varphi(X)$ is (statistically) independent of the domain i.e.*

$$p_S(Z) = p_T(Z) \tag{3.45}$$

for $Z := \varphi(X)$.

It is worth noting that domain invariance can be relaxed by measuring how similar are source and target distributions of representations when provided with a distance between distributions. Crucially, the search of statistical invariance in the representation space is closely related with the bound from Theorem 3.24 as described in the following. We assume that the hypothesis class $\mathcal{H}$ is a composition of a representation class Φ and a classifier class $\mathcal{G}$ *i.e.* $\mathcal{H} = \{g \circ \varphi, g \in \mathcal{G}, \varphi \in \Phi\} =: \mathcal{G} \circ \Phi$. 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*. We present a variation of theorem 3.24 when applied in the representation space:

Theorem 3.4 (Analysis of representation for Domain Adaptation (Ben-David et al. 2007))**.** *Let $g \in \mathcal{G}$ and $\varphi \in \Phi$:*

$$\mathrm{Err}_T(g\varphi) \leq \mathrm{Err}_S(g\varphi) + \delta_{\mathcal{G}}(\varphi) + \lambda_{\mathcal{G}}(\varphi) \tag{3.46}$$

where:

$$\delta_{\mathcal{G}}(\varphi) := \sup_{g,g' \in \mathcal{G}^2} \mathrm{Err}_S(g\varphi, g'\varphi) - \mathrm{Err}_T(g\varphi, g'\varphi) \tag{3.47}$$

$$\lambda_{\mathcal{G}}(\varphi) := \inf_{g \in \mathcal{G}} \mathrm{Err}_S(g\varphi) + \mathrm{Err}_T(g\varphi) \tag{3.48}$$

Proof. Apply theorem 3.24 where $\varphi(X)$ and $\mathcal{G}$ has the role of X and $\mathcal{H}$, respectively. □

This generalization bound ensures that the target risk $\mathrm{Err}_T(g\varphi)$ is bounded by the sum of the source risk $\mathrm{Err}_S(g\varphi)$, the discrepancy between source and target distributions of representations $\delta_{\mathcal{G}}(\varphi)$, 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* of representations.

Crucially, both the divergence $\delta_{\mathcal{G}}(\varphi)$ and the adaptability $\lambda_{\mathcal{G}}(\varphi)$ depends on the representation φ, which characterizes a fundamental difference with the theory from (Ben-David et al. 2010a) presented in Section 3.2. Indeed, in Theorem 3.3, both the discrepancy $\delta_{\mathcal{H}}$ and the adaptability $\lambda_{\mathcal{H}}$ depend on the hypothesis class $\mathcal{H}$, thus do not depend on a particular choice of an hypothesis $h \in \mathcal{H}$. Conversely, both the discrepancy $\delta_{\mathcal{G}}(\varphi)$ and the adaptability $\lambda_{\mathcal{G}}(\varphi)$ depend on a particular choice of $h = g \circ \varphi \in \mathcal{H}$, more precisely through its representation φ and the whole classifier class $\mathcal{G}$. We exhibit two crucial inequalities, first observed in (Johansson, Sontag, and Ranganath 2019); $\forall \varphi \in \Phi$,

$$\begin{aligned} \delta_{\mathcal{G}}(\varphi) &\leq \delta_{\mathcal{H}} \\ \lambda_{\mathcal{G}}(\varphi) &\geq \lambda_{\mathcal{H}} \end{aligned} \tag{3.49}$$

that can be simply derived from the following inclusion of hypothesis class $\mathcal{G}\varphi \subset \mathcal{H}$. On the one hand, one can expect from an invariant representation to reduce drastically $\delta_{\mathcal{G}}(\varphi)$. On the other hand, using a representation increases the adaptability error. Importantly, in a context of Unsupervised Domain Adaptation, we can not compute $\lambda_{\mathcal{G}}(\varphi)$ since we do not have access to the target labels. Ultimately, a UDA method that aims to leverage domain invariant representation should guarantee that the look for invariance is beneficial, even if we are exposed to the risk of increasing the adaptability error.

Similarly to Proposition 3.2.4, one can relate $\delta_{\mathcal{G}}(\varphi)$ to the accuracy of a discriminator trained to discriminate the source from the target domain based on representations $Z = \varphi(X)$;

Proposition 3.3.1. *Let Φ a representations class, $\mathcal{G}$ a classifier class and $D_{\mathcal{G}}$ a discriminator class of $\mathcal{G}$. Let $\mathcal{D}_S$ and $\mathcal{D}_T$, two set of IID realizations from p_S and p_T respectively. For the sake of simplicity, we assume $|\mathcal{D}_S| = |\mathcal{D}_T| = n$. For any $\delta \in (0, 1)$, with probability at*

least $1-\delta$,

$$\delta_{\mathcal{G}}(\varphi) \leq \left(1 - \frac{1}{n} \min_{d \in D_{\mathcal{G}}} \left\{ \sum_{x \in \mathcal{D}_S} 1 - d(\varphi(x)) + \sum_{x \in \mathcal{D}_T} d(\varphi(x)) \right\}\right) + 4 \sqrt{\frac{\mathrm{VC}\left(D_{\mathcal{G}}\right) \log(2n) + \log \frac{2}{\delta}}{n}} \tag{3.50}$$

Crucially, when φ is an invariant representation, *i.e.* $p_T(\varphi(X)) = p_S(\varphi(X))$ as presented in Definition 3.3.1, the discrepancy is null; $\delta_{\mathcal{G}}(\varphi) = 0$.

Domain Adversarial Learning of Invariant Representations

We show how to traduce the theoretical result from Proposition 3.3.1 into an efficient algorithm for learning a domain invariant representations, *i.e.* learning $\varphi \in \Phi$ such that $p_S(\varphi(X))$ and $p_T(\varphi(X))$ are close ensuring a small $\delta_{\mathcal{G}}(\varphi)$.

Formulation. We present the founding algorithm learning domain invariant representations (Ganin and Lempitsky 2015; Ganin et al. 2016) based on *Domain Adversarial Learning*, *i.e.* through a discriminator that provides a feedback about the invariance of representations. We first describe the overall strategy, which consists in achieving a trade-off between source classification error, *i.e.* a small $\mathrm{Err}_{\mathcal{D}_S}(g\varphi)$, while controlling the invariance of representations *i.e.* a small $\delta_{\mathcal{G}}(\varphi)$. Second, we present the novelty of this paradigm that is to estimate invariance of representations through the performance of a classifier trained to separate the source from the target domain. Let provide the general formulation:

$$g^\star, \varphi^\star := \arg \min_{g \in \mathcal{G}, \varphi \in \Phi} L_c(g\varphi) + \lambda \cdot L_{\text{inv}}(\varphi) \tag{3.51}$$

where $L_c(g\varphi)$ is the cross-entropy computed on the source dataset $\mathcal{D}_S$, $L_{\text{inv}}(\varphi)$ reflects the level of domain invariance of φ and $\lambda > 0$ controls the strength of the invariance constraint. In this section, we dive into details for elaborating the invariance loss $L_{\text{inv}}(\varphi)$ (**Invariance loss**) as well as the algorithm for efficiently solving the equation 3.79 (**Gradient Reversal Layer**).

Invariance loss. According to proposition 3.2.4, $\delta_{\mathcal{G}}(\varphi)$ is bounded by the classification performance of a discriminator $D \in D_{\mathcal{G}}$ trained to separate the source from the target domain. It is equivalent to a binary classification problem where samples from the source domain are labelled as positive instances while samples from the target domain are labelled as negative instances. As a result, Ganin et al. define $L_{\text{inv}}(\varphi)$ as follows (Ganin and Lempitsky 2015; Ganin et al. 2016):

$$L_{\text{inv}}(\varphi) := - \inf_{D \in D_{\mathcal{G}}} L_{\text{dis}}(\varphi, d) \tag{3.52}$$

$$L_{\text{dis}}(\varphi, d) := \frac{1}{n_S} \sum_{x \in \mathcal{D}_S} - \log d(\varphi(x)) + \frac{1}{n_T} \sum_{x \in \mathcal{D}_T} - \log\left(1 - d(\varphi(x))\right) \tag{3.53}$$

It is worth noting that $L_{\text{dis}}(\varphi, d)$ is the cross-entropy where the task of separating domains is framed as a binary classification of domains. Additionally, $L_{\text{inv}}(\varphi)$ involves

$\inf_{d \in \mathcal{D}_\mathcal{G}}$ reflecting the performance of the best domain discriminator. Crucially, L_{inv} is the opposite of L_{dis}. Since L_{inv} promotes domain invariance of representations, a discriminator should not be able to separate domains *i.e.* $L_{\text{dis}}(\varphi, d)$ should remain high. This new paradigm is usually referred to as *Domain Adversarial Learning* since it is closely related to the *MiniMax* problem:

$$\min_{\varphi \in \Phi} \max_{d \in \mathcal{D}_\mathcal{G}} -L_{\text{dis}}(\varphi, d) \tag{3.54}$$

This problem implies a new challenge; computing $L_{\text{inv}}(\varphi)$ is an optimization problem by itself. In (Ganin and Lempitsky 2015; Ganin et al. 2016), Ganin et al. suggest a simple approximation for solving the problem that appears to work well in practice. Rather than computing the best domain discriminator at each SGD step, a domain discriminator is continuously learned during training in a similar fashion than representation φ and classifier g:

$$\begin{cases} (g, \varphi) \leftarrow (g, \varphi) - \alpha \left(\nabla_{(g,\varphi)} \left\{ L_c(g\varphi) - \lambda L_{\text{dis}}(\varphi, d) \right\} \right) \\ d \leftarrow d - \alpha \nabla_d \lambda L_{\text{dis}}(\varphi, d) \end{cases} \tag{3.55}$$

Gradient Reversal Layer. The *MiniMax* problem, which involves respectively φ and D, leads to a bi-level SGD update, as presented in equation 3.55. However, the two updates share similarity through $L_{\text{dis}}(\varphi, d)$. In the (Ganin and Lempitsky 2015; Ganin et al. 2016), authors suggest an elegant solution for deriving a single SGD update from 3.55 using a *Gradient Reversal Layer* (`GRL`):

$$\texttt{GRL} : \begin{cases} x \mapsto x & (\texttt{Forward}) \\ -\mathbf{g} \leftarrow \mathbf{g} & (\texttt{Backward}) \end{cases} \qquad \text{(Gradient Reversal Layer)}$$

The Gradient Reversal Layer has an unusual behavior by uncoupling the forward and the backward pass. The forward pass is the identity *i.e.* $\texttt{GRL}(x) = x$, while the backward pass reverse the sign of the gradient *i.e.* $\nabla(\texttt{GRL})(\mathbf{g}) = -\mathbf{g}$ where $\mathbf{g}$ is a gradient. For the purpose of illustration, we provide a minimal implementation of `GRL`. As a result, passing the representations $\varphi(x)$ through the Gradient Reversal Layer *i.e.* $\texttt{GRL}(\varphi(x))$, before feeding the discriminator allows to implement the bi-level update rule from 3.55 as a single update rule:

$$(g, \varphi, d) \leftarrow (g, \varphi, d) - \alpha \nabla_{(g,\varphi,d)} \left\{ L_c(g\varphi) + \lambda \cdot L_{\text{inv}}^{\texttt{GRL}}(\varphi, d) \right\} \tag{3.56}$$

where $L_{\text{dis}}^{\texttt{GRL}}(\varphi, d) := L_{\text{dis}}(\varphi, d \circ \texttt{GRL})$.

```python
class GradientReversalLayer(nn.Module):
    def __init__(self):
        super(GradientReversalLayer, self).__init__()
        self.hook = lambda grad : - 1.0 * grad.clone()

    def forward(self, x):
        x.register_hook(self.hook) # Reverse the gradient.
        return x
```

A `Pytorch` implementation of the Gradient Reversal Layer.

Practical Improvements

Learning domain invariant representations relies on the adversarial training of a representation φ and a discriminator D, which is a sub-optimal approximation of the true discrepancy $\delta_{\mathcal{G}}(\varphi)$ between the source and the target distributions of representations. Some works provide simpler strategies for learning domain invariant representations. A prior work to (Ganin and Lempitsky 2015) from (Baktashmotlagh et al. 2013) suggests to learn a domain invariant projection of representations. Similarly, a feature scaling transformation of inputs, leading to a new representations of inputs, has been proposed in (Zhang et al. 2013) and (Wang, Huang, and Schneider 2014). More recently, (Sun, Feng, and Saenko 2016; Sun and Saenko 2016) introduce a simple baseline consisting in simply aligning mean and covariance of source and target representations.

In practice, domain adversarial training as introduced in (Ganin and Lempitsky 2015) remains unstable. Such a phenomenon is similar to the training of GANs (Goodfellow et al. 2014) where its difficulty has been studied in depth in (Salimans et al. 2016; Arjovsky and Bottou 2017; Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017). Several works aim to improve learning of domain invariant representations through the use of different measure to compare distributions. The work of (Long et al. 2015), concomitant to (Ganin and Lempitsky 2015), inspired by (Gretton et al. 2012), uses a kernel approximation of the *Maximum Mean Discrepancy* (MMD) between the source and the target distributions of representations, thus not involving a domain discriminator. (Shen et al. 2018) suggests to improve the domain adversarial training with the Wasserstein distance between the source and the target distributions of representations, following fruitful conclusions about adversarial training developed in (Arjovsky, Chintala, and Bottou 2017). Following this line of study, (Bhushan Damodaran et al. 2018) incorporates the fast approximation of the (entropic regularized) transportation plan, through the Sinkhorn algorithm (Cuturi 2013), between the source and the target distributions of representations. Recently, the work (Zhang et al. 2019) introduces a novel invariance loss (Margin Disparity Discrepancy) promoting large margin. Learning domain invariant representations has been also extending in a context of multi-sources domain adaptation (Zhao et al. 2018; Peng et al. 2019) and the line of works that aims to select the source domains in this context (Afridi, Ross, and Shapiro 2018; Bascol, Emonet, and Fromont 2019).

3.3.3 The General Principle of Invariance

Looking for well-suited invariance has played a key role for empowering generalization, particularly for over-parametrized models in Deep Learning (LeCun, Bengio, and Hinton 2015). For instance, data augmentation (Schmidhuber, Meier, and Ciresan 2012; Sato, Nishimura, and Yokoi 2015; Simard, Steinkraus, Platt, et al. 2003; Wan et al. 2013; Cubuk et al. 2018) aims to enforce the model to remain invariant to standard transformations such as elastic distortions, re-scaling, translation and rotation. Modern deep convolution neural networks (LeCun et al. 1998), with mechanisms of parameters sharing and pooling, take advantage of the spatial structure of the image in order to infer on images with some degree of invariance, for instance with respect to (local) translations (LeCun, Bengio, and Hinton 2015). We review some other implementations of the principle of invariance beyond the scope of Unsupervised Domain Adaptation.

Invariance to a nuisance factor

As designing hypothesis class that remains invariant to a specific nuisance factor, *e.g.* rotation of images or world deletion of text, is time consuming, the work (Xie et al. 2017) suggests to learn invariant representation $Z = \varphi(X)$ through the following Information Bottleneck (Tishby, Pereira, and Bialek 2000) objective;

$$\varphi^\star = \underset{\varphi\in\Phi, Z:=\varphi(X)}{\arg\max} \; I(Y;Z) - \lambda \cdot I(Z;S) \tag{3.57}$$

where S is the identified nuisance factor and I the mutual information between two random variables. $I(Y;Z)$ promotes a good predictive power of representations Z to learn Y while $I(Z;S)$ promotes the independence of Z and S and $\lambda > 0$ is a trade-off parameter. A variational formulation has been derived by (Moyer et al. 2018). This framework coincides with learning domain invariant representations in domain adaptation (Ganin and Lempitsky 2015), where S is a binary variable indicating the domain. The work (Feutry et al. 2018) extends this framework to the case where S is a categorical random variable and applies this principle to learn anonymous representation with respect to a user.

Invariant Risk Minimization and Domain Generalization

Invariant Risk Minimization (IRM) (Arjovsky et al. 2019) aims to learn a representation that *elicits an invariant predictor*, that is a representation that verifies;

$$\mathbb{E}_T[Y|\varphi(X)] = \mathbb{E}_S[Y|\varphi(X)] \tag{3.58}$$

Looking for such a representation finds several motivations. First, we have developed in Section 3.3.1 the intuition that to obtaining a low adaptability error requires the similarity of labeling function $f_D : z \mapsto \mathbb{E}_D[Y|\varphi(X) = z]$. We will elaborate theoretically such a motivation in Chapter 5. Second, one can relate this underlying objective through the lens of causality (Pearl 2009). More specifically, if the true labels Y are *caused* by variable Z, *i.e.* $Z \to Y$, one can reasonably believe the relation between Z to Y does not depend on variation in the environment, *i.e.* is invariant on environments. The intimate connection between invariance and causality suggests that invariant descriptions of objects relate to the causal explanation of the object itself (Lopez-Paz et al. 2017). The IRM's objective is to learn such variable Z as a function of inputs X, *i.e.* $Z = \varphi(X)$. To gain insight about the relation between invariance and causality, we depict the founding example from IRM.

Example 3.3.2 (Invariance and Causality through a Structural Equation Model)**.** *Let $\sigma > 0$ and the following Structural Equation Model*

$$X_1 \leftarrow \mathcal{N}(0, \sigma^2); \tag{3.59}$$

$$Y \leftarrow X_1 + \mathcal{N}(0, \sigma^2); \tag{3.60}$$

$$X_2 \leftarrow Y + \mathcal{N}(0, 1) \tag{3.61}$$

There is a causal relation between feature X_1 and target Y ($X_1 \to Y$) and an anti-causal relation between feature X_2 and target Y ($Y \to X_2$). By changing the value of $\sigma > 0$, we generate shift in the environment, e.g. $\sigma_S^2 = 10$ for the source domain and $\sigma_T^2 = 20$ for the target domain. We predict Y from $\mathbf{X} = (X_1, X_2)$ with a least-square predictor $f(\mathbf{X}) = \alpha \cdot X_1 + \beta \cdot X_2$. We consider three cases;

- *Regressing from X_1, i.e. enforcing $\beta = 0$, leads to $\alpha = 1$.*
- *Regressing from X_2, i.e. enforcing $\alpha = 0$, leads to $\beta = \frac{\sigma^2}{\sigma^2+\frac{1}{2}}$.*
- *Regressing from (X_1, X_2) leads to $\alpha = \frac{1}{\sigma^2+1}$ and $\beta = \frac{\sigma^2}{\sigma^2+1}$.*

The only model that does not depend on σ, i.e. an invariant predictor, is the regression from X_1, i.e. the variable that causes Y. In particular, one can also note that $\mathbb{E}[Y|X_1] = X_1$, thus does not depend on σ. Therefore, the representation $\varphi(\mathbf{X}) = X_1$ is a representation that elicits an invariant predictor.

To discover a representation that elicits an invariant predictor, the work from (Arjovsky et al. 2019) suggests to look for a representation $\varphi \in \Phi$ such that it exits a classifier $g \in \mathcal{G}$ which is optimal in both domains;

$$\arg\min_{g \in \mathcal{G}} \text{Err}_S(g\varphi) = \arg\min_{g \in \mathcal{G}} \text{Err}_T(g\varphi) \tag{3.62}$$

To traduce this joint optimality, *i.e.* a classifier is simultaneously optimal in both domains, into a training objective, authors of (Arjovsky et al. 2019) suggest a simple, yet elegant, phrasing. Provided with a loss $L_D(g, \varphi)$ that acts as a proxy of $\text{Err}_D(g\varphi)$, *e.g.* L_D is the cross-entropy error L_c computed in domain D, if g is optimal in domain D then;

$$\nabla_g L_D(g, \varphi) = 0 \tag{3.63}$$

To enforce the constraint $\nabla_g L(g, \varphi)$ for both the source and the target domains while minimizing the source and the target errors, IRM is then the following relaxed optimization problem;

$$L_{\text{IRM}}(g, \varphi) = \sum_{D \in \{S,T\}} L_D(g, \varphi) + \lambda \cdot ||\nabla_g L_D(g, \varphi)||^2 \tag{3.64}$$

where $\lambda > 0$ is a trade-off parameter between;

1. achieving a low error in the source and the target domains simultaneously,
2. controlling the resulting classifier is optimal in both domains.

Now that we have explained the general purpose of IRM, we note that it deviates from the setup of UDA since it requires labels in the target domain for computing $L_T(g, \varphi) + \lambda \cdot ||\nabla_g L_T(g, \varphi)||^2$. More precisely, IRM addresses the problem of *Domain Generalization*, *i.e.* generalizing to domains for which no information is available during learning, *e.g.* we do not have unlabeled data as it is assumed in UDA. Formally, we assume that N source domains, noted $S_1, \cdots, S_N$ where domain S_k is populated with labelled samples $(x_i^{S_k}, y_i^{S_k})_{1 \leq i \leq n_{S_k}}$, and the model is evaluated on an unknown target domain T. In a context of Domain Generalization (Section 2.3.2), the IRM objective becomes;

$$L_{\text{IRM}}(g, \varphi) = \sum_{D \in \{S_1, \cdots, S_n\}} L_D(g, \varphi) + \lambda \cdot ||\nabla_g L_D(g, \varphi)||^2 \tag{3.65}$$

Theoretically, IRM empowers generalization for linear representations, providing a strong extension of results of (Peters, Bühlmann, and Meinshausen 2016). In particular, the higher the rank of the linear representation, a smaller number of domains is required to learn a linear representation that elicits an invariant predictor. The case of a non-linear representations, that are ubiquitous tools in Deep Learning, remains

an open and highly challenging problem. Practically, the problem of Domain Generalization, typically when using deep neural networks, remains a difficult problem where it is not clear if IRM can extrapolate to new domains (Gulrajani and Lopez-Paz 2021). However, IRM has impacted substantially the field (Ahuja et al. 2020; Chang et al. 2020; Krueger et al. 2021). (Arjovsky 2021) provides a thorough elaboration of IRM.

Fair Representations

Fair Machine Learning (Corbett-Davies and Goel 2018) aims to learn models that do not harm individuals based on sensitive attributes, *e.g.* sex or ethnic origin. Learning *Fair Representations* has attracted a lot of attention in recent years (Zemel et al. 2013; Louizos et al. 2016; Edwards and Storkey 2016; Madras et al. 2018). The main idea is to enforce invariance with respect to a sensitive attribute in the data in order to achieve a targeted property of fairness. We provide a review below of the main fairness criteria and of the established connections between representations invariance and fairness. In the following, we note X the raw features, labels Y and the sensitive attribute S (S indicates the gender or the 'race' for instance), conserving similar notations with the problem of distribution shift, emphasizing the similarity between the two fields.

Definition 3.3.2 (Demographic Parity)**.** *An hypothesis $h \in \mathcal{H}$ satisfies demographic parity if*

$$\forall (s, s') \in \mathcal{S}, \mathbb{P}(\hat{Y}|S = s) = \mathbb{P}(\hat{Y}|S = s') \tag{3.66}$$

Definition 3.3.3 (Equalized odds)**.** *An hypothesis $h \in \mathcal{H}$ satisfies equalized odds if*

$$\forall (s, s') \in \mathcal{S}, \forall y \in \mathcal{Y}, \mathbb{P}(\hat{Y}|Y = y, S = s) = \mathbb{P}(\hat{Y}|Y = y, S = s') \tag{3.67}$$

A weaker notion is also introduced in the literature, as equalized opportunity where $p(\hat{Y}|Y = y, S = s) = p(\hat{Y}|Y = y, S = s')$ concerns only a particular value of y.

Definition 3.3.4 (Predictive Value Parity)**.** *An hypothesis $h \in \mathcal{H}$ satisfies predictive value parity if*

$$\forall (s, s') \in \mathcal{S}, \forall y \in \mathcal{Y}, \mathbb{P}(Y|\hat{Y} = y, S = s) = \mathbb{P}(Y|\hat{Y} = y, S = s') \tag{3.68}$$

The choice of the relevant fairness criterion is the object of philosophical and moral discussions beyond the scope of this work. Invariant Representations is an elegant manner to achieve a fairness criterion by enforcing it directly in representations. Namely, it is straightforward to show the following property for a given representation $Z = \varphi(X)$:

- If $\forall (s, s') \in \mathcal{S}^2, \mathbb{P}(Z|S = s) = \mathbb{P}(Z|S = s')$ then for any $g \in \mathcal{G}$, noting $h = g \circ \varphi$, h achieves demographic parity.
- If $\forall (s, s') \in \mathcal{S}^2, \forall y \in \mathcal{Y}, \mathbb{P}(Z|Y = y, S = s) = \mathbb{P}(Z|Y = y, S = s')$ then for any $g \in \mathcal{G}$, noting $h = g \circ \varphi$, h achieves equalized odds.

These properties make learning fair representation well-suited for adversarial learning (Madras et al. 2018).

3.4 Transferability of Domain Invariant Representations

3.4.1 Motivations

Learning domain invariant representations presented in Section 3.3 and depicted in Equation 3.55 is exposed to numerous limitations that has led to a wide literature to address it. One can frame those limitations into two important lines of study. The first aims to improve the approximation of the discrepancy $\delta_{\mathcal{G}}(\varphi)$, embedded by the invariance loss $L_{\text{inv}}(\varphi)$, in order to make the learning of domain invariant representations more efficient. This aspect has been reviewed in Section 3.3.2. The second is concerned by the fundamental trade-off of learning domain invariant representations as described by the influential theory from (Ben-David et al. 2007; Ben-David et al. 2010a) and exposed in Equation 3.49;

How to guarantee that invariance does not impact badly the adaptability $\lambda_{\mathcal{G}}(\varphi)$?

The risk of a high adaptability error is a problem that has recently become apparent and is surprisingly under-investigated. The analysis of adaptability in multi-source domain adaptation (Redko, Habrard, and Sebban 2019) is the first work, to our knowledge, that deliberately addresses the problem of estimating the adaptability error. We review some important works that aims to improve transferability of domain invariant representations, both theoretically and practically.

3.4.2 Theory

A fundamental trade-off

Learning domain invariant representations is exposed to fundamental theoretical limits. As firstly described by the work (Johansson, Sontag, and Ranganath 2019), learning invariant representations is always made with the drawback of increasing the risk of adaptability. Indeed, when considering an hypothesis space $\mathcal{H} = \mathcal{G} \circ \varphi$ where $\mathcal{G}$ is a set of classifiers and φ is a representation, there is $\lambda_{\mathcal{H}} \leq \lambda_{\mathcal{G}\varphi}$, *i.e.* the adaptability error increases, as presented in Equation 3.49. Note that this result holds for any representation and is, in particular, true for a representation that aims to reduce the discrepancy $\delta_{\mathcal{G}}(\varphi)$. Thus, controlling the discrepancy $\delta_{\mathcal{G}}(\varphi)$ through domain invariant representations, should be more beneficial than the risk of increasing the adaptability error. Note that, such statement is impossible to guarantee in the standard setup of UDA since the adaptability error requires the knowledge of labels in the target domain. We refer to the example 3.3.1 for an intuitive illustration of such a phenomenon.

The challenge of label shift.

One can construct realistic case of adaptation where the seek of invariance is guarantee to fail, *i.e.* minimizing $\delta_{\mathcal{G}}(\varphi)$ through the lens of domain invariant will lead to high adaptability error $\lambda_{\mathcal{G}}(\varphi)$ with theoretical guarantee. In the work (Zhao et al. 2019), authors shows that it is impossible to obtain a successful adaptation algorithm based on domain invariant representations when the distribution of labels is significantly different across domains, a situation refered to as *label shift*; $p_T(Y) \neq p_S(Y)$ (Section 2.2.3), which is very common in practice (Lipton, Wang, and Smola 2018). More specifically, Zhao et al. show that the adaptability error is greater than;

$$\frac{1}{2}\left(\Delta_{\text{JS}}(\varphi(X)) - \Delta_{\text{JS}}(Y)\right)^2 \leq \lambda_{\mathcal{G}}(\varphi) \tag{3.69}$$

where for some random variable U, $\Delta_{\text{JS}}(U) = \text{JS}(p_S(U), p_T(U))$ is the Jensen-Shannon divergence between source and the target distributions of U. This results is of the upmost importance since when learning a domain invariant representation $\varphi(X)$ to control $\delta_{\mathcal{G}}(\varphi)$, *i.e.* $\Delta_{\text{JS}}(\varphi(X))$, then the adaptability is directly linked to the difference of label distributions across domains, through $\frac{1}{2}\Delta_{\text{JS}}(Y)^2 \leq \lambda_{\mathcal{G}}(\varphi)$. Thus, if provided a domain invariant representations φ, one can not consider the adaptability error to be small when $\Delta_{\text{JS}}(Y)$, that quantifies the amount of label shift, is high.

On the difficulty of estimating adaptability.

As aforementioned, the crucial problem of estimating the adaptability error $\lambda_{\mathcal{G}}(\varphi)$ is an under-investigated problem. To our knowledge, (Redko et al. 2019) is the first work that specifically addresses it. Note that the analysis does not take place under the paradigm of learning representations. In (Redko, Habrard, and Sebban 2019), the authors provide a consistent approximation of the adaptability error, based on the Wasserstein distance between label distributions, when several source domains are available, a setup referred to as *Multi-Domains Adaptation*. Such result holds under the assumption that, when sampling a source domain D, the associated labelling function $f_D : x \mapsto \mathbb{E}_D[Y|X = x]$ is an independent realization of $\mathbb{P}_f \in \mathcal{P}\left([0,1]^{\mathcal{X}}\right)$ from the set of distributions of functions from $\mathcal{X} \to [0,1]$. In particular, the estimation of adaptability error $\lambda_{\mathcal{H}}$ increases with the number of domains. Intuitively, each domain brings information about the underlying distribution of labelling functions $\mathbb{P}_f$[5].

When learning domain invariant representations, the work (Chuang, Torralba, and Jegelka 2020) has made a substantial effort for estimating the target error without labels in the target domain. The main idea is to consider the set of hypothesis built upon an invariant representation associated with the best source classifier;

$$\mathcal{H}_0 := \left\{ g_S\varphi : g_S = \arg\min_{g\in\mathcal{G}} \text{Err}_S(g\varphi), \varphi \in \Phi_0 \right\} \tag{3.70}$$

where $\Phi_0 = \{\varphi \in \Phi \text{ such that } p_S(\varphi(X)) = p_T(\varphi(X))\}$ is the set of domain invariant representations. Given $g\varphi \in \mathcal{H}_0$, one can bound the error $\text{Err}_T(g\varphi)$, based on the triangle inequality $\text{Err}_T(g\varphi) \leq \text{Err}_T(g\varphi, h) + \text{Err}_T(h)$ for any $h \in \mathcal{H}_0$, leading to;

$$\text{Err}_T(g\varphi) \leq \underbrace{\sup_{h\in\mathcal{H}_0} \text{Err}_T(g\varphi, h)}_{\text{Proxy risk}} + \underbrace{\inf_{h\in\mathcal{H}_0} \text{Err}_T(h)}_{\text{Bias}} \tag{3.71}$$

First, $\inf_{h\in\mathcal{H}_0} \text{Err}_T(h)$ called the *Bias* of $\mathcal{H}_0$ quantifies the error of the best classifier in $\mathcal{H}_0$ and involves the labels in the target domain to be computed, thus it is not tractable in the standard setup of UDA. Intuitively, this quantity is small if it exists a domain invariant representation and a classifier trained on source samples that achieves a small target error. Second, $\sup_{h\in\mathcal{H}_0} \text{Err}_T(g\varphi, h)$ called the *Proxy risk* of $\mathcal{H}_0$, quantifies if the classifiers trained on source samples based on a domain invariant representation yield to different predictions. Crucially, the proxy risk of $\mathcal{H}_0$ does not involve the labels in the target domain, then it is *a priori* tractable in the standard setup of UDA. This analysis can be straightforwardly extended to representations that do not achieve exact invariance, noted $\Phi_\varepsilon := \{\varphi \in \Phi, d(p_S(\varphi(X), p_T(\varphi(X)) \leq$

[5]Interestingly, this setup of adaptation shares many connections with *Invariant Risk Minimization* (IRM) (Arjovsky et al. 2019) that we review in 3.3.3

$\varepsilon\}$ for some distance d between distributions and ε is a small positive real number. One can then define $\mathcal{H}_\varepsilon := \{g_S\varphi : g_S := \arg\min_{g\in\mathcal{G}} \mathrm{Err}_S(g\varphi), \varphi \in \Phi_\varepsilon\}$, leading to the same Equation 3.71 replacing $\mathcal{H}_0$ by $\mathcal{H}_\varepsilon$.

To apply this result in a setup of UDA, where labels are not available in the target domain, one can enforce additional assumption. In a similar spirit than (Ben-David et al. 2010a), which assumes that the adaptability error $\lambda_\mathcal{H}$ is small and embodies our capacity to adapt at first, the work (Chuang, Torralba, and Jegelka 2020) assumes that it exists $h^\star \in \mathcal{H}_0$ such that $\mathrm{Err}_T(h^\star)$ is small, leading to a small $\inf_{h\in\mathcal{H}_0} \mathrm{Err}_T(h)$. Under this assumption, one can only be concerned by the proxy risk $\sup_{h\in\mathcal{H}_0} \mathrm{Err}_T(g\varphi, h)$ to bound the target error $\mathrm{Err}_T(g\varphi)$. Interestingly, this quantity is directly related to the size of representations φ that are invariant, *i.e.* $\varphi \in \Phi_0$; the smaller Φ_0, the lower the proxy risk, and conversely. Intuitively, this relates to the underlying complexity of the representation class Φ, does invariance is a sufficient constraint for exhibiting a unique, or a small number, of invariant representations?, a phenomenon referred to as the *Embedding Complexity* of Φ measured through the proxy risk (Chuang, Torralba, and Jegelka 2020). However, this theoretical analysis does not escape from the fundamental trade-off of UDA, *e.g.* it may not exist an invariant representation φ such that a classifier g_S trained in the source domain achieves a small target error $\mathrm{Err}_T(g_S\varphi)$ as supported by Equation 3.69 in a context of label shift.

3.4.3 Improving Transferability of Domain Invariant Representations

Ensuring that invariant representations do not degrade the adaptability error $\lambda_\mathcal{G}(\varphi)$ is a new challenge for UDA and has motivated a new line of works that aims to improve transferability of domain invariant representations. This difficult is a central aspect of this thesis and we review in the following some important contributions.

Conditional Domain Adaptation Network (CDAN).

Conditional Domain Adaptation Network (CDAN) (Long et al. 2018) has dramatically improved the efficiency of domain adversarial learning of invariant representations. CDAN addresses some limitations of Domain Adaptation Neural Network (DANN) (Ganin and Lempitsky 2015) that typically fails to align complex multimodal distribution of data, *i.e.* mixture of distributions where each component corresponds to a specific class. The elaboration of CDAN follows insights obtained from training of *Generative Adversarial Networks* (GANs) (Goodfellow et al. 2014) that note that adversarial training is improved when conditioned with respect to relevant information such as labels or known modality (Mirza and Osindero 2014; Odena, Olah, and Shlens 2017; Isola et al. 2017). CDAN suggests to overcome this issue by incorporating prediction $\hat{Y} := g \circ \varphi(X)$ into the domain adversarial objective. For the exposition of CDAN, we deviate from the definition of the classifier that outputs a $\hat{y} \in \mathcal{Y}$. Here, $g\varphi(x)$ is the vector of probabilities where $g\varphi(x)_y$ is the (estimated) probability of sample x to belong to class $y \in \mathcal{Y}$. The domain adversarial objective of CDAN is;

$$L_{\mathrm{CDAN}}(\varphi, D) := \frac{1}{n_S} \sum_{x\in\mathcal{D}_S} -\log D(T_{g,\varphi}(x)) + \frac{1}{n_T} \sum_{x\in\mathcal{D}_T} -\log\left(1 - D(\varphi(T_{g,\varphi}(x)))\right) \tag{3.72}$$

where $T_{g,\varphi}(x) = \varphi(x) \otimes g\varphi(x)$ where $\otimes$ is multilinear map. Intuitively, the CDAN objective provides information about corresponding class for the discriminator by conditioning with respect to predicted class $\hat{Y}$. Indeed, the expectation of $\mathbb{E}[T_{g,\varphi}(X)]$

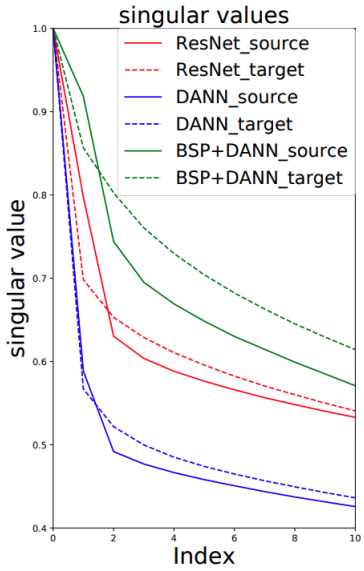


Figure 3.2: Effect of *Batch Spectral Penalization* on the transferability of domain invariant representations. We represent the normalized singular values $\sigma_{D,i}(\varphi)/\sigma_{D,1}(\varphi)$ where i is the index, D is source or target domain, and φ is a pre-trained ResNet (He et al. 2016) on ImageNet (Deng et al. 2009) (red), or adapted with DANN loss (Ganin and Lempitsky 2015) (blue) or which benefits of the BSP (Chen et al. 2019c) penalization (green). When learning an invariant representation with DANN (blue), the singular values are highly dominated by the first singular values, a phenomenon which is not observed for ResNet representations (red), possibly indicating the lack of transferability of invariant representations as speculated by authors in (Chen et al. 2019c). When provided with the *Batch Spectral Penalization* (Chen et al. 2019c), which consists to penalize the higher singular values of batch of representations, representations is more transferable as indicated by the smaller importance given to the higher singular values. Figure from (Chen et al. 2019c).

has the interpretable following form;

$$\mathbb{E}[T_{g,\varphi}(X)] = \mathbb{E}[\varphi(X)|\hat{Y}=0] \oplus \cdots \oplus \mathbb{E}[\varphi(X)|\hat{Y}=C] \tag{3.73}$$

Note that the discriminator is then a function from $\mathbb{R}^{d\times C}$ to $[0,1]$, where d is the dimension of $\mathcal{Z}$ and C is the number of class, which may be of high dimension in practice. To address this issue, we usually rely on $\tilde{T}_{g,\varphi}(X) := \tilde{R}T_{g,\varphi}(X)$ where $\tilde{R}$ is a random projection from $\mathbb{R}^{d\times C}$ to $\mathbb{R}^{d'}$ with $d' \ll d \times C$.

However, the theoretical aspect of why incorporating prediction $\hat{Y}$ when learning domain invariant representations, and creates a positive feedback loop for invariance, remains poorly understood. In Chapter 5, we provide a theoretical understanding through the lens of the inductive bias that has been a prominent tool in Machine Learning.

Batch Spectral Penalization (BSP).

The work (Chen et al. 2019c) brings crucial insight about the difficulty of learning domain invariant representations while conserving their transferability. To highlight such evidence, they conducted an empirical study based on a ResNet (He et al. 2016), pre-trained on ImageNet (Deng et al. 2009), that we note φ_{ResNet}, or adapted using L_{DANN} (Ganin and Lempitsky 2015) (See Section 3.3.2) on a standard benchmark for

UDA (Office-31 (Saenko et al. 2010)), that we note φ_{DANN}. They draw the following observations;

- As expected, using domain invariant representations leads to an improvement of performances in the target domain compared to a classifier obtained from pre-trained representations.
- However, the adaptability error on φ_{DANN} is higher than φ_{ResNet}, *i.e.*

$$\lambda_{\mathcal{G}}(\varphi_{\text{ResNet}}) \leq \lambda_{\mathcal{G}}(\varphi_{\text{DANN}}),$$

  thus invariance has degraded transferability of representations. Note that this is a typical example where the benefit of looking for domain invariant representations is higher than their lose of transferability since we improve the target classification when using φ_{DANN} compared to φ_{ResNet}.
- When looking at the principal components obtained from a *Principal Components Analysis* (PCA), we observe that higher eigen-values are more prominent compared to smaller eigen-values for φ_{DANN} than it is for φ_{ResNet}. Authors claim that it may indicate a lack of transferability (authors refer to *discriminability* of representations) of φ_{DANN} compared to non-adapted representations from φ_{ResNet}. We provide more details about this intriguing phenomenon in the following.

Given a representation φ, we note its covariance matrix Λ_D in a domain $D \in \{S, T\}$;

$$\Lambda_D(\varphi) := \mathbb{E}_D[\varphi(X)^\top \varphi(X)] \in \mathbb{R}^{d \times d} \tag{3.74}$$

associated with (positive) eigen-values $\sigma^2_{D,1} \geq \cdots \geq \sigma^2_{D,d}$ where $\sigma_{D,i}$ is the singular value of index i. In Figure 3.2, we report the importance of the highest singular values in the decomposition by plotting $\sigma_{D,i}/\sigma_{D,1}$ with respect to the index i. We observe that singular values are highly dominated by the first singular value for φ_{DANN} than it is φ_{ResNet}, possibly indicating a higher adaptability error for φ_{DANN} than for φ_{ResNet}. To learn domain invariant representations that do not fall into such a domination of the highest singular value, the work (Chen et al. 2019c) suggests the *Batch Spectral Penalization* (BSP). It consists into penalizing the top k singular values, the learning objective depicted in Equation 3.79 becomes;

$$g_{\text{BSP}}, \varphi_{\text{BSP}} := \arg \min_{g \in \mathcal{G}, \varphi \in \Phi} L_c(g\varphi) + \lambda \cdot L_{\text{inv}}(\varphi) + \gamma \cdot L_{\text{BSP}}(\varphi), \tag{3.75}$$

for some $\gamma > 0$, where;

$$L_{\text{BSP}}(\varphi) := \sum_{i=1}^{k} \sigma_{i,S}(\varphi) + \sigma_{i,T}(\varphi) \tag{3.76}$$

As suspected by authors, such a penalization reduces drastically the domination of the highest singular value (see Figure 3.2), resulting to a drastic improvement of performances presented in Table 3.1. BSP is an influential tool for improving transferability of domain invariant representations but lacks of theoretical ground. In Chapter 4, we will develop a theoretical analysis providing theoretical insight about the power of BSP.

Method	Office-31	Office-Home	VisDA
ResNet (He et al. 2016)	76.1	46.1	45.6
DANN (Ganin et al. 2016)	82.2	57.6	55.0
DANN+BSP (Chen et al. 2019c)	87.7	64.9	72.1

Table 3.1: Selected average accuracy (%) results from (Chen et al. 2019c) to demonstrate the effect of the *Batch Spectral Penalization* (BSP) when learning domain invariant representations.

Partial, Universal and Open set adaptation.

We present the problem of *Partial Domain Adaptation* (Cao et al. 2018), and its extension *Universal Domain Adaptation* (You et al. 2019) (related to *Open Set Adaptation* (Panareda Busto and Gall 2017)), as our first example where Equation 3.69 prevents from learning transferable domain invariant representations. To highlight such difficulty, it has been proposed in (Cao et al. 2018) the problem of Partial Domain Adaptation, where target classes are a strict subset of source classes, *i.e.* target classes are a subset of source classes with a lower number of classes.

Definition 3.4.1 (Partial Domain Adaptation)**.** *Partial Domain Adaptation is an instance of problem of Unsupervised Domain Adaptation (See Definition 3.2.2) where the target classes is a strict subset of source classes.*

$$\text{Supp}(p_T(Y)) \subsetneq \text{Supp}(p_S(Y)) \tag{3.77}$$

As the source and target distributions of labels are different, one can expect to observe a high $\Delta_{JS}(Y)$, thus a high adaptability error according to Equation 3.69. To address Partial Domain Adaptation, (Cao et al. 2018) suggests to combine importance with domain invariant representations, that are originally two independant lines of study, a method named *Partial Adversarial Domain Adaptation* (PADA). More specifically, both the classification loss and the domain adversarial loss and the domain adversarial are weighted such that the label distribution of source samples better represents the label distribution of target samples. Cao et al. build a weight vector $w = (w_1, \cdots, w_C) \in \mathbb{R}^C$ as the empirical distribution of $\hat{Y} = g\varphi(X)$ in the target domain, *i.e.* $w_c(g, \varphi) = p_{\mathcal{D}_T}(\hat{Y} = c)$, *i.e.* ;

$$w_c(g, \varphi) := \frac{\sum_{j=1}^{n_T} g\varphi_c(x_j^T)}{\sum_{c'=1}^{C} \sum_{j=1}^{n_T} g\varphi_{c'}(x_j^T)}, \quad \text{for } c \in \{1, \cdots, C\} \tag{3.78}$$

where we emphasize the dependence of w with both g and φ. The domain adversarial objective from Equation 3.79 is as follows;

$$g^\star, \varphi^\star := \arg \min_{g \in \mathcal{G}, \varphi \in \Phi} L_c^w(g\varphi) + \lambda \cdot L_{\text{inv}}^w(\varphi) \tag{3.79}$$

Method	Office-31 (10)	Office-Home (25)	VisDA (6)
ResNet (He et al. 2016)	75.6	53.7	54.8
DANN (Ganin et al. 2016)	42.4	47.4	62.4
PADA (Cao et al. 2018)	92.7	62.1	65.0

Table 3.2: Selected average accuracy (%) results from (Cao et al. 2018) to demonstrate the effect of the *Partial Adversarial Domain Adaptation* (PADA) when learning domain invariant representations in a context of *Partial Domain Adaptation* (PDA). Office-31 (10), Office-Home (25) and VisDA (6) indicate that such datasets are built from a subset of the first classes; 10, 25 and 6 respectively, to emulate a situation of PDA.

where;

$$L_c^w(g,\varphi) := \frac{1}{n_S}\sum_{i=1}^{n_S} w_{y_i^S}(g,\varphi)\ell_c(y_i^S, g\varphi(X)) \tag{3.80}$$

$$L_{\text{inv}}^w(\varphi) := -\inf_{d\in D_{\mathcal{G}}} L_{\text{dis}}^w(\varphi,d) \tag{3.81}$$

$$L_{\text{dis}}^w(\varphi,d) := \frac{1}{n_S}\sum_{i=1}^{n_S} -w_{y_i^S}(g,\varphi)\log d(\varphi(x_i^S)) + \frac{1}{n_T}\sum_{j=1}^{n_T} -\log(1-d(\varphi(x_j^T))) \tag{3.82}$$

where ℓ_c denotes the cross-entropy loss. This simple strategy that takes into account the shift of label distribution across domains leads to substantial improvement of performances as presented in Table 3.2.

Partial Domain Adaptation has allowed the emergence of new problems for UDA, *e.g.* the universal (or open set (Panareda Busto and Gall 2017)) domain adaptation (You et al. 2019) where the target domain may contain classes that are not present in the source domain. Partial Domain Adaptation shares strong connection with UDA under label shift *i.e.* when $p_S(Y) \neq p_T(Y)$ (see Section 3.2.3). Learning domain invariant representation under label shift has attracted a recent attention (Zhang et al. 2018; Wu et al. 2019; Combes et al. 2020a) with various strategies to design weights. The theoretical understanding on how to design weight will be discussed in Section 5.3 of Chapter 5.

Part II

Ingredient of Adaptation: A Theoretical View

4 Hypothesis Class Reduction

Contents

Ensuring a good adaptability of invariant representations is the bottleneck of Domain Adversarial Learning for UDA. Indeed, invariance can be conflicting with a low adaptability error, as exposed in Section 3.4. Unfortunately, computing the adaptability error is impossible in a UDA scenario since it requires labels in the target domain. The present Chapter provides an extension of the theory (Ben-David et al. 2010a) that aims to control the adaptability error of representations better.

In Section 4.1, we first start by introducing a founding example where the source and target supports of distributions do not overlap, a typical case of distribution shift of high dimensional data (D'Amour et al. 2021). In this example, learning invariant representations for reconciling support of representations identifies two models. Both models achieve an equally small distribution discrepancy in the representation space, but result in very different adaptabilities; the former has a low adaptability error while the latter is high. Thus, even if an invariant representation that leads to a successful adaptation exists, simply looking for invariance is insufficient to identify such a representation.

To overcome this lack of identifiability of invariant representations, we look for an additional condition in Section 4.2. In particular, we focus our effort on the last

remaining unexplored term from the theory (Ben-David et al. 2010a); the adaptability error itself. As exposed above, the adaptability error is a term over which we have little control since it involves labelled data in the target domain. Nevertheless, the adaptability error is defined as the minimal combined error achieved by a classifier trained from representations, thus has a concrete structure. By exploiting this condition of minimality, we derive a new error term in the seminal theory from (Ben-David et al. 2010a). We interpret this error term as the risk of *Hypothesis Class Reduction* (HCR), *i.e.* the risk that invariance deletes information in the representations space, reducing the size of the hypothesis class on which we look for the classifier achieving the smallest combined error. Crucially, by considering the risk of hypothesis class reduction, we gain better control over the adaptability error without involving the knowledge of labels in the target domain. Therefore, we can now better understand the conditions under which the adaptability error may be high.

In Section 4.3, we provide at the theoretical level potential applications of the new error term of HCR. Namely, we derive a generic algorithm for boosting the learning of domain invariant representations. Additionally, HCR provides the needed theoretical ground of (Chen et al. 2019c), an empirical work that aims to improve transferability of domain invariant representations by penalizing a domination of large eigen-values in the representation space.

The present chapter is an extension of the technical report (Bouvier et al. 2020a) and similar ideas have been explored in the same time (Chuang, Torralba, and Jegelka 2020) that we reviewed in Section 3.4.2.

4.1 Preliminaries

4.1.1 The fundamental trade-off between invariance and transferability of representations

Domain Adversarial Learning has dramatically shifted the paradigm of Unsupervised Domain Adaptation (UDA) with deep neural networks. However, ensuring a good transferability of representations while learning domain invariant representations remains an open problem that requires a new look. We first start by recalling a quick overview of the seminal theory from (Ben-David et al. 2007), introduced in Theorem 3.4 of Section 3.3.2. For a given representation $\varphi \in \Phi$, and for a classifier $g \in \mathcal{G}$, one can bound the target risk of $g\varphi := g \circ \varphi$ as follows;

$$\mathrm{Err}_T(g\varphi) \leq \mathrm{Err}_S(g\varphi) + \delta_{\mathcal{G}}(\varphi) + \lambda_{\mathcal{G}}(\varphi) \tag{4.1}$$

where $\delta_{\mathcal{G}}(\varphi) := \sup_{g,g' \in \mathcal{G}^2} \mathrm{Err}_S(g\varphi, g'\varphi) - \mathrm{Err}_T(g\varphi, g'\varphi)$ is the distribution discrepancy error and;

$$\lambda_{\mathcal{G}}(\varphi) := \inf_{g \in \mathcal{G}} \mathrm{Err}_S(g\varphi) + \mathrm{Err}_T(g\varphi) \tag{4.2}$$

is the adaptability error of representation φ. We have shown in Section 3.4 that the work (Johansson, Sontag, and Ranganath 2019) has exhibited a fundamental trade-off, a representation $\varphi \in \Phi$ always reduces the discrepancy error while increasing the adaptability error, *i.e.* $\delta_{\mathcal{G}}(\varphi) \leq \delta_{\mathcal{H}}$ and $\lambda_{\mathcal{G}}(\varphi) \geq \lambda_{\mathcal{H}}$, as presented in Equation 3.49. At first glance, this result, which is valid for any representation, seems surprising. Intuitively, a representation removes information initially present in the data, which brings the source distribution closer to the target distribution. More formally, the following inclusion $\mathcal{G}\varphi \subset \mathcal{H}$ is sufficient to prove the result.

We extend this result when provided with two representations, respectively φ and $\psi \in \Phi$. To this purpose, we need to introduce a central concept of this chapter that defines the *reduction* of a representation;

Definition 4.1.1 (Hypothesis Class Reduction (HCR))**.** *Let $(\varphi, \psi) \in \Phi^2$, ψ is reduced with respect to φ if $\mathcal{G} \circ \psi \subset \mathcal{G} \circ \varphi$, and we note $\psi \prec \varphi$.*

The representation ψ is reduced with respect to φ if the hypothesis class resulting from ψ ($\mathcal{G}\psi$) is smaller than the hypothesis class resulting from φ ($\mathcal{G}\varphi$), *i.e.* $\mathcal{G}\psi \subset \mathcal{G}\varphi$. Thus, the reduction of a representation φ is measured through another representation ψ that we refer to as a *witness representation*.

For instance, let consider a two dimensional problem where $\mathcal{X} = \mathbb{R}^2$, we note $\mathbf{x} = (x_1, x_2) \in \mathcal{X}$, Φ the set of linear applications from $\mathcal{X}$ to itself and $\mathcal{G}$ the set of linear classifiers $\mathcal{G} = \{\mathbf{x} \mapsto \mathbb{I}(\mathbf{w}^\top \mathbf{x}), \mathbf{w} = (w_1, w_2) \in \mathbb{R}^2\}$. Let $\psi : \mathbf{x} \mapsto (x_1, 0)$ and $\varphi : \mathbf{x} \mapsto (x_1, x_2)$. We can observe that any hypothesis $g'\psi(x) = \mathbf{w}'^\top \psi(x)$ can be expressed as an hypothesis $g\varphi(x) = \mathbf{w}^\top \psi(x)$ by enforcing $w_2 = 0$, thus ψ is reduced with respect to φ. This property results from the fact that ψ deletes information from the input since it removes x_2.

The definition of Hypothesis Class Reduction (HCR) finds natural connections with the principle of *Structural Risk Minimization* (SRM), reviewed in Section 3.1.3, where for two hypothesis class $\mathcal{H}_1 \subset \mathcal{H}_2$ that achieve the same empirical error, one should consider the simplest hypothesis class, *i.e.* $\mathcal{H}_1$, to prevent from the risk of overfitting.

The concept of reduction has a tangible implication on the fundamental trade-off of learning transferable domain invariant representations;

Proposition 4.1.1 (Reduction improves invariance but hurts transferability (Johansson, Sontag, and Ranganath 2019; Zhao et al. 2019))**.** *Let $\varphi, \psi \in \Phi$ such that $\psi \prec \varphi$, then,*

$$\delta_{\mathcal{G}}(\psi) \leq \delta_{\mathcal{G}}(\varphi) \text{ while } \lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi) \tag{4.3}$$

This property offers a new look on the risk of hurting transferability of representations; if ψ is reduced with respect to φ and $\delta_{\mathcal{G}}(\varphi) = \delta_{\mathcal{G}}(\psi)$, then the transferability of φ is better than ψ since $\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi)$. The objective of this Chapter is to understand how this consideration can impact the theory (Ben-David et al. 2010a) and ultimately may derive a new learning objective for improving transferability of domain invariant representations.

4.1.2 Intuition through a structural equation model

We provide a founding example where domain invariance is not sufficient, but, when equipped with the concept of reduction, we can guarantee the success of adaptation. Our example draws inspiration from the example from *Invariant Risk Minimization* (IRM) (Arjovsky et al. 2019) that we have exposed in Section 3.3.3. Pragmatically, we deviate from Example 3.3.2 to build a source and target distribution such that inputs have non-overlapping support;

Example 4.1.1 (A Structural Equation model)**.** *We consider the following structural equation model where for $\sigma > 0$;*

$$X_1 \leftarrow \mathcal{N}(0, \sigma^2); \tag{4.4}$$

$$Y \leftarrow X_1 + \mathcal{N}(0, \sigma^2); \tag{4.5}$$

$$X_2 \leftarrow \begin{cases} Y + \mathcal{N}(0,1) & \text{in the source domain;} \\ 0 & \text{in the target domain.} \end{cases}; \tag{4.6}$$

$$X_3 \leftarrow \begin{cases} 0 & \text{in the source domain;} \\ Y + \mathcal{N}(0,1) & \text{in the target domain.} \end{cases}. \tag{4.7}$$

where we aim to fit Y from $\mathbf{X} = (X_1, X_2, X_3)$.

The example deviates from (Arjovsky et al. 2019) since X_3 is not specified in Example 3.3.2. Here, we keep σ^2 fixed, *e.g.* $\sigma^2 = 1$, and the distribution shift situation rises from the symmetric role of X_2 and X_3 in the source and target domains, reflecting an information that is not encoded in the same dimension across domains. Our objective is to identify, in a scenario of UDA, that X_2 and X_3 play a similar role, *i.e.* reconciling the supports of distributions. In particular, we can observe that the linear representation $\varphi : \mathbb{R}^3 \to \mathbb{R}^2$ that maps $\mathbf{X}$ to $(X_1, X_2 + X_3)$ aligns the source and the target distributions.

Although this example seems artificial, non-overlapping distributions are ubiquitous when dealing with non-overlapping data. We provide an illustration of this phenomenon for images and texts;

- **images:** the process of image centring may vary across domains, the resulting semantic information are not encoded in the same pixels.

- **text:** the spelling of some words may vary across domains (*e.g.* 'favour' and 'favor' for the non-British variant), the resulting semantic are not encoded in the same dimension of a tokenizer[1].

4.1.3 Invariant Linear Regressor

Problem statement

We study how a linear regression, equipped with an invariance constraint, can adapt from the source to the target distribution specified in the Example 4.1.1. More precisely, we express a linear regressor h as;

$$h = g\varphi \tag{4.8}$$

where both g and the representation φ are linear, *i.e.* noting $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Z} = \mathbb{R}^m$, then $\varphi \in \mathbb{R}^{m\times d}$ and $g \in \mathbb{R}^{1\times m}$. The representation φ will be used to enforce property of invariance for the linear regressor, hence its name of linear invariant regressor.

In a scenario of regression, it is traditional to use the L_2 risk of h in domain $D \in \{S, T\}$; $R_D(h) = \left(\mathbb{E}_D\left[(Y - h(X))^2\right]\right)^{1/2}$, to evaluate model's performance[2]. In the following, we note a linear regressor h as follows;

$$h(\mathbf{X}) = \alpha \cdot X_1 + \beta \cdot X_2 + \gamma \cdot X_3 \tag{4.9}$$

where for the particular example, we consider $d = 3$ and $m = 2$. First, the optimal model $h^\star$, *i.e.* the model that minimizes the adaptability error $R_S(h) + R_T(h)$, is;

$$h^\star(\mathbf{X}) = \alpha^\star \cdot X_1 + (1 - \alpha^\star) \cdot (X_2 + X_3) \tag{4.10}$$

with $\alpha^\star = \frac{1}{\sigma^2+1}$, *i.e.* $\beta^\star = \gamma^\star = 1 - \alpha^\star$.

Second, a model that minimizes the source risk $h_S = \arg\min_{h\in\mathcal{H}} R_S(h)$ is;

$$h_S(\mathbf{X}) := \alpha^\star \cdot X_1 + (1 - \alpha^\star) \cdot X_2 + \gamma \cdot X_3 \text{ for all } \gamma \in \mathbb{R}. \tag{4.11}$$

Similarly, a model that minimizes the target risk $h_T = \arg\min_{h\in\mathcal{H}} R_T(h)$ is $h_T(\mathbf{X}) := \alpha^\star \cdot X_1 + \beta \cdot X_2 + (1 - \alpha^\star) \cdot X_3$ for all $\beta \in \mathbb{R}$. Thus, minimizing the source risk is insufficient for exhibiting the relation $\gamma = 1 - \alpha^\star$. We investigate how the principle of invariance can adapt the model to the target distribution *i.e.* recovering the relation $\gamma = 1 - \alpha^\star$.

Enforcing invariance

In this section, we study under which conditions enforcing invariance of a representation φ improves the model in the target domain, *i.e.* recovering the relation $\gamma = 1 - \alpha^\star$. We first motivate our interest into learning domain invariant representation by showing that the analysis from (Ben-David et al. 2010a), presented in

[1] We do not assume the access of a well-suited tokenizer that can deal with spelling.

[2] For the completeness of the exposition, we mention that the theory presented in Section 3.2 remains valid when using the L^2 loss; $R_T(g\varphi) \leq R_S(g\varphi) + \sup_{(g,g')\in\mathcal{G}^2}\{R_T(g'\varphi, g\varphi) - R_S(g'\varphi, g\varphi)\} + \inf_{g\in\mathcal{G}}\{R_T(g\varphi) + R_S(g\varphi)\}$ since it only requires that the risk verifies the triangular inequality, *i.e.* $R(h,h') \leq R(h,h'') + R(h'',h')$ where we note $R_D(h) := (\mathbb{E}_D[(Y - h(X))^2])^{1/2}$ and $R_D(h,h') := (\mathbb{E}_D[(h'(X) - h(X))^2])^{1/2}$, which the case for $R(\cdot)$.

equation 3.24, is not useful since the distribution discrepancy is unbounded. We note in the following $\mathbb{V}_S[U]$ the variance of the random variable U.

Proposition 4.1.2 (Equation 3.24 is no useful)**.** *Given the structural equation model from Example 4.1.1 and* $\delta_{\mathcal{H}} := \sup_{(h,h')\in\mathcal{H}^2} R_T(h,h') - R_S(h,h')$*, then* $\delta_{\mathcal{H}} = \infty$*.*

Proof. We set $h(\mathbf{X}) := \hat{\alpha}\cdot X_1 + \beta\cdot X_2$ and $h'(\mathbf{X}) := \hat{\alpha}\cdot X_1$, thus $h(\mathbf{X}) - h(\mathbf{X}') = \beta\cdot X_2$, thus $R_T(h,h') - R_S(h,h') = |\beta|\mathbb{V}_S[X_2]^{1/2} = |\beta|(2\sigma^2+1)^{1/2}$ that tends to $+\infty$ when β tends to $+\infty$. □

Now that we have motivated our need to learn a domain invariant representation, we show that this is still insufficient. Indeed, through our example, we show that invariance in φ for an invariant regressor $h = g\varphi$ may be insufficient to recover the optimal regressor in the target domain, *i.e.* recovering the relation $\gamma = 1-\alpha^\star$. In particular, we consider the two following representations φ_1 and φ_2;

- $\varphi_1(\mathbf{X}) := (\hat{\alpha}\cdot X_1 + \beta\cdot X_2 + \gamma\cdot X_3, 0)$,
- $\varphi_2(\mathbf{X}) := (\hat{\alpha}X_1, \beta\cdot X_2 + \gamma\cdot X_3)$.

and we fix $g = (1,1)$. We show that by choosing carefully the values of α,β and γ, one can achieve invariance and a minimal source risk, *i.e.* equal to $R_S(h^\star)$. However, invariance is insufficient to guarantee a minimal target risk, *i.e.* equal to $R_T(h^\star)$. Namely, we show that $\varphi_1(\mathbf{X})$ is insufficient to recover reliably the relation $\gamma = 1-\alpha^\star$.

Proposition 4.1.3 (Invariance on $\varphi_1(\mathbf{X})$ mis-specifies γ)**.** *Given the structural equation model from Example 4.1.1,* $\mathcal{H}$ *the set of linear regressor and we note* $h(\mathbf{X}) = (1,1)\cdot\varphi_1(\mathbf{X})$ *with* $\varphi_1(\mathbf{X}) := (\alpha\cdot X_1 + \beta\cdot X_2 + \gamma\cdot X_3, 0)$*, enforcing invariance on* $\varphi_2(\mathbf{X})$ *leads to* $\gamma = 1-\alpha^\star$ *or* $\gamma = -\frac{1+(2\sigma^2-1)\alpha^\star}{1+2\sigma^2}$*.*

Proof. First, minimizing the source error sets $\alpha = \alpha^\star$ and $\beta = 1-\alpha^\star$. Second, $\alpha^\star\cdot X_1 + \beta\cdot X_2 + \gamma\cdot X_3$ is a gaussian variable with zero mean, hence, it only requires to align variance of the first dimension of $\varphi_1(X)$. Such variance in the source domain is $V_S := \mathbb{V}_S[\alpha^\star\cdot X_1 + (1-\alpha^\star)\cdot X_2 + \gamma\cdot X_3] = \mathbb{V}_S[\alpha^\star\cdot X_1 + (1-\alpha^\star)\cdot(X_1+\epsilon_1+\epsilon_{\sigma^2})]$ where $\epsilon_1\sim\mathcal{N}(0,1)$ and $\epsilon_{\sigma^2}\sim\mathcal{N}(0,\sigma^2)$ which are mutually independent from X_1, thus $V_S = \mathbb{V}_S[X_1 + (1-\alpha^\star)(\epsilon_1+\epsilon_{\sigma^2})] = \sigma^2 + (1-\alpha^\star)^2(\sigma^2+1)$. Similarly, in the target domain is $V_T := \mathbb{V}_T[\alpha^\star\cdot X_1 + (1-\alpha^\star)\cdot X_2 + \gamma\cdot X_3] = \mathbb{V}_S[\alpha^\star\cdot X_1 + \gamma\cdot(X_1 + \epsilon_1+\epsilon_{\sigma^2})] = (\alpha^\star+\gamma)^2\sigma^2 + \gamma^2(\sigma^2+1)$. Thus, the constraint $V_S = V_T$ exhibits two values of γ; $\gamma = 1-\alpha^\star$ or $\gamma = -\frac{1+(2\sigma^2-1)\alpha^\star}{1+2\sigma^2}$ by computing the real square root of the polynomial function in γ; $\gamma\mapsto(\alpha^\star+\gamma)^2\sigma^2+\gamma^2(\sigma^2+1)-\sigma^2+(1-\alpha^\star)^2(\sigma^2+1)$. □

Proposition 4.1.4 (Invariance on $\varphi_2(\mathbf{X})$ learns the optimal linear regressor)**.** *Given the structural equation model from Example 4.1.1,* $\mathcal{H}$ *the set of linear regressor and we note* $h(\mathbf{X}) = (1,1)\cdot\varphi_2(\mathbf{X})$ *with* $\varphi_2(\mathbf{X}) := (\alpha\cdot X_1, \beta\cdot X_2 + \gamma\cdot X_3)$*, enforcing invariance on* $\varphi_2(\mathbf{X})$ *leads to* $\gamma = 1-\alpha^\star$*.*

Proof. First, minimizing the source error sets $\alpha = \alpha^\star$ and $\beta = 1-\alpha^\star$. Second, $(\hat{\alpha}\cdot X_1, \beta\cdot X_2 + \gamma\cdot X_3)$ is a gaussian vector with zero mean, hence, it only requires to align covariance across domains; $\Sigma_S := \begin{pmatrix} \sigma^2 & \alpha^\star(1-\alpha^\star)\sigma^2 \\ \alpha^\star(1-\alpha^\star)\sigma^2 & (1-\alpha^\star)^2(2\sigma^2+1)\end{pmatrix}$ and $\Sigma_T := \begin{pmatrix} \sigma^2 & \alpha^\star\gamma\sigma^2 \\ \alpha^\star\gamma\sigma^2 & \gamma^2(2\sigma^2+1)\end{pmatrix}$. The constraint $\Sigma_S = \Sigma_T$ enforces $\gamma = 1-\alpha^\star$. □

Since φ_1 exhibits two (equally well) values for γ, only φ_2 is able to recover the optimal regressor in the target domain. Thus, while achieving the same trade-off between a low source risk and invariance, φ_1 and φ_2 do not generalize equally well in the target domain. Therefore, this generalization gap is reflected in the adaptability risk $\lambda_{\mathcal{G}}(\varphi_2) \leq \lambda_{\mathcal{G}}(\varphi_1)$. Crucially, it is possible to show that φ_2 has a better adaptability than φ_1 without knowing labels in the target domain. Indeed, one can show that φ_1 is reduced with respect to φ_2, which guarantees that φ_2 has a better adaptability risk than φ_1. This is of utmost importance since the UDA's difficulty lies in the fact that we cannot compute the adaptability. However, we have exhibited an example where it is still possible to know that given two representations, one has better guarantees than the other at equal source risk and invariance. In the following, we aim to extend this result through the lens of *Hypothesis Class Reduction* (HCR).

4.2 Analysis of Hypothesis Class Reduction

4.2.1 Hypothesis Class Reduction Error

We have provided the intuition that if a representation ψ is reduced with respect to φ, *i.e.* $\mathcal{G}\psi \subset \mathcal{G}\varphi$ and noted $\psi \prec \varphi$, we have a better control on the adaptability error, thus better guarantee of a successful adaptation. In particular, we have provided an example where *Hypothesis Class Reduction* (HCR) allows to identify the optimal model. However, the reduction is simply a relation between two representations and is not, as yet, quantifiable, *i.e.* "by how much is ψ reduced with respect to φ?". Intuitively, ψ is reduced with respect to φ means that each hypothesis built from ψ is also an hypothesis built from φ. Thus, if we build labels $Y' = g'\psi(X)$ for some $g \in \mathcal{G}$, it exists $g \in \mathcal{G}$ such that $Y' = g\varphi$ resulting to a null error $\mathrm{Err}_p(g\varphi, g'\psi)$ for any distribution of inputs p, in particular $\mathrm{Err}_S(g\varphi, g'\psi) + \mathrm{Err}_T(g\varphi, g'\psi) = 0$. In particular, $\inf_{g\in\mathcal{G}} \mathrm{Err}_S(g\varphi, g'\psi) + \mathrm{Err}_T(g\varphi, g'\psi) = 0$. This consideration motivates us to introduce the HCR error;

Definition 4.2.1 (Hypothesis Class Reduction Error)**.** *Let two representations $\varphi, \psi \in \Phi^2$ and $\mathcal{G}$ a set of classifiers. The Hypothesis Class Reduction error is defined as;*

$$\gamma_{\mathcal{G}}(\varphi, \psi) := \sup_{g'\in\mathcal{G}} \left\{ \inf_{g\in\mathcal{G}} \left[\mathrm{Err}_S(g\varphi, g'\psi) + \mathrm{Err}_T(g\varphi, g'\psi) \right] \right\} \tag{4.12}$$

The HCR error draws inspiration from the ability to fit any labels $Y' = g'\psi(X)$ from an hypothesis $g\varphi$, hence, defining it as a supremum over $g' \in \mathcal{G}$. Thus, the HCR error is a proxy that quantifies the relation of reduction. Crucially, if $\psi \prec \varphi$, then $\gamma_{\mathcal{G}}(\varphi, \psi) = 0$ while $\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi)$. The main contribution of this chapter is to generalize this result by proving that $\gamma_{\mathcal{G}}(\varphi, \psi) = 0$ implies that φ has a lower adaptability error than ψ, *i.e.* $\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi)$. Note that this is non-trivial result since $\gamma_{\mathcal{G}}(\varphi, \psi) = 0$ does not imply $\psi \prec \varphi$.

It is worth noting that, unlike adaptability, the HCR error $\gamma_{\mathcal{G}}(\varphi, \psi)$ does not involve target labels making it appealing for UDA where labels of target data are absent.

4.2.2 Bounding adaptability error with Hypothesis Class Reduction error

As aforementioned, the relation of HCR is immediatly related to the fundamental trade-off described in 4.1.1; if $\psi \prec \varphi$ then $\delta_{\mathcal{G}}(\psi) \leq \delta_{\mathcal{G}}(\varphi)$ while $\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi)$. We now extend the result concerning the adaptability error to the HCR error $\gamma_{\mathcal{G}}(\varphi, \psi)$,

that quantifies the HCR relation; $\gamma_{\mathcal{G}}(\varphi,\psi) \implies (\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi))$. We first start by proving the following result;

Proposition 4.2.1 (Witnessing reduction). *Let $\psi \in \Phi$, then:*

$$\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi,\varphi) + \gamma_{\mathcal{G}}(\varphi,\psi) \tag{4.13}$$

where $\lambda_{\mathcal{G}}(\psi,\varphi) := \min\{\lambda_{\mathcal{G}}(\varphi), \lambda_{\mathcal{G}}(\psi)\}$.

Proof. For the ease of reading, we introduce $\mathrm{Err}_C(h,h') = \mathrm{Err}_S(h,h') + \mathrm{Err}_T(h,h')$ and $\mathrm{Err}_C(h) = \mathrm{Err}_S(h) + \mathrm{Err}_T(h)$. First, $\forall g' \in \mathcal{G}, \mathrm{Err}_C(g\varphi) \leq \mathrm{Err}_C(g\varphi, g'\psi) + \mathrm{Err}_C(g'\psi)$. Second, we take the inf on $g \in \mathcal{G}$;

$$\begin{aligned}
\inf_{g\in\mathcal{G}} \mathrm{Err}_C(g\varphi) &\leq \inf_{g\in\mathcal{G}} \left[\mathrm{Err}_C(g\varphi, g'\psi)\right] + \mathrm{Err}_C(g'\psi) \\
&\leq \sup_{g'\in\mathcal{G}} \left\{ \inf_{g\in\mathcal{G}} \left[\mathrm{Err}_C(g\varphi, g'\psi)\right] \right\} + \mathrm{Err}_C(g'\psi) \\
&\leq \sup_{g'\in\mathcal{G}} \left\{ \inf_{g\in\mathcal{G}} \left[\mathrm{Err}_C(g\varphi, g'\psi)\right] \right\} + \inf_{g'\in\mathcal{G}} \mathrm{Err}_C(g'\psi) \\
&\leq \gamma_{\mathcal{G}}(\varphi,\psi) + \lambda_{\mathcal{G}}(\psi)
\end{aligned}$$

The first inequality is obtained by computing the inf on $g \in \mathcal{G}$. The second by bounding $\inf_{g\in\mathcal{G}} [\mathrm{Err}_C(g\varphi, g'\psi)]$ by the supremal on $g' \in \mathcal{G}$. The third is by noting that $\inf_{g\in\mathcal{G}} \mathrm{Err}_C(g\varphi)$ does not depend on g', then it is lower than the infremal on $g' \in \mathcal{G}$ of inequality's right hand. The final inequality consists in replacing the terms by $\lambda_{\mathcal{G}}(\varphi)$, $\gamma_{\mathcal{G}}(\varphi,\psi)$ and $\lambda_{\mathcal{G}}(\psi)$ respectively. To obtain the announced result, we note that $\lambda_{\mathcal{G}}(\varphi) \leq \max\{\gamma_{\mathcal{G}}(\varphi,\psi), \gamma_{\mathcal{G}}(\varphi,\varphi)\} + \min\{\lambda_{\mathcal{G}}(\varphi), \lambda_{\mathcal{G}}(\psi)\}$ where $\gamma_{\mathcal{G}}(\varphi,\varphi) = 0$. □

In particular, one can observe it implies $\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi) + \gamma_{\mathcal{G}}(\varphi,\psi)$ since $\lambda_{\mathcal{G}}(\varphi,\psi) \leq \lambda_{\mathcal{G}}(\psi)$. Thus, if $\gamma_{\mathcal{G}}(\varphi,\psi) = 0$, we have $\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi)$. Note this result holds independently of the relation of HCR, thus providing an extension of the same result through the HCR error. We can draw interesting conclusions about the adaptability of two representations, even when there is a non-null HCR error;

Theorem 4.1 (Guarantee of better adaptability). *Let $\varphi, \psi \in \Phi$ and assume that $\gamma_{\mathcal{G}}(\varphi,\psi) \leq \gamma_{\mathcal{G}}(\psi,\varphi)$, thus φ has better guarantee about its adaptability error than the adaptability error of ψ since we can exhibit a smaller upper of the former than the latter.*

Proof. From Proposition 4.2.1, one can write;

$$\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi,\varphi) + \gamma_{\mathcal{G}}(\varphi,\psi) \tag{4.14}$$
$$\lambda_{\mathcal{G}}(\psi) \leq \lambda_{\mathcal{G}}(\varphi,\psi) + \gamma_{\mathcal{G}}(\psi,\varphi) \tag{4.15}$$

Noting that $\lambda_{\mathcal{G}}(\varphi,\psi) = \lambda_{\mathcal{G}}(\psi,\varphi)$ leads to;

$$\underbrace{\lambda_{\mathcal{G}}(\psi,\varphi) + \gamma_{\mathcal{G}}(\varphi,\psi)}_{\geq \lambda_{\mathcal{G}}(\varphi)} \leq \underbrace{\lambda_{\mathcal{G}}(\varphi,\psi) + \gamma_{\mathcal{G}}(\psi,\varphi)}_{\geq \lambda_{\mathcal{G}}(\psi)} \tag{4.16}$$

We have exhibited an upper bound of $\lambda_{\mathcal{G}}(\varphi)$ which is smaller than an upper bound of $\lambda_{\mathcal{G}}(\psi)$, thus φ has better guarantee of lower adaptability than ψ. □

	$\delta_{\mathcal{G}}(\varphi)$	$\gamma_{\mathcal{G}}(\varphi,\psi)$	$\lambda_{\mathcal{G}}(\varphi,\psi)$
φ	$\nearrow$	$\searrow$	$\searrow$
ψ		$\nearrow$	$\searrow$

Table 4.1: Behavior of error terms introduced in the bound from Theorem 4.2 with respect to the relation of Hypothesis Class Reduction (HCR). The notation $\nearrow$ means the quantity increases with the HCR relation, *i.e.* $\delta_{\mathcal{G}}(\varphi) \nearrow \varphi$ means for $\varphi_1, \varphi_2 \in \Phi$ such that $\varphi_1 \prec \varphi_2$, we have $\delta_{\mathcal{G}}(\varphi_1) \leq \delta_{\mathcal{G}}(\varphi_2)$. For a given $\varphi \in \Phi$, $\gamma_{\mathcal{G}}(\varphi,\psi) \nearrow \psi$ means for $\psi_1, \psi_2 \in \Phi$ such that $\psi_1 \prec \psi_2$, we have $\gamma_{\mathcal{G}}(\varphi,\psi_1) \leq \gamma_{\mathcal{G}}(\varphi,\psi_2)$.

Note Theorem 4.1 does not prove that if $\gamma_{\mathcal{G}}(\varphi,\psi) \leq \gamma_{\mathcal{G}}(\psi,\varphi)$ implies $\lambda_{\mathcal{G}}(\varphi) \leq \lambda_{\mathcal{G}}(\psi)$. Nevertheless, Theorem 4.1 indicates we should promote φ if $\gamma_{\mathcal{G}}(\varphi,\psi) \leq \gamma_{\mathcal{G}}(\psi,\varphi)$.

4.2.3 A new bound

Based on the HCR error $\gamma_{\mathcal{G}}(\varphi,\psi)$, we can provide a new upper bound of the target risk;

Theorem 4.2 (Bounding the target error with Hypothesis Class Reduction)**.** *For a given $\varphi \in \Phi$ and $g \in \mathcal{G}$, there is for all $\psi \in \Phi$:*

$$\mathrm{Err}_T(g\varphi) \leq \mathrm{Err}_S(g\varphi) + \delta_{\mathcal{G}}(\varphi) + \gamma_{\mathcal{G}}(\varphi,\psi) + \lambda_{\mathcal{G}}(\varphi,\psi) \tag{4.17}$$

Proof. This is a direct application of Proposition 4.2.1. □

Let us analyze this new bound. On the one hand, compared to the bound (Ben-David et al. 2010a), this bounds provides a better control on the intractable part of the inequality ($\lambda_{\mathcal{G}}(\varphi,\psi)$) since $\lambda_{\mathcal{G}}(\varphi,\psi) \leq \lambda_{\mathcal{G}}(\varphi)$. On the other hand, this is paid at cost $\gamma_{\mathcal{G}}(\varphi,\psi)$ that overestimates the risk of bad adaptability since $\lambda_{\mathcal{G}}(\varphi) \leq \inf_{\psi\in\Phi}\{\gamma_{\mathcal{G}}(\varphi,\psi) + \lambda_{\mathcal{G}}(\varphi,\psi)\}$. However, this bound brings fruitful arguments; considering two representations φ and ψ that have the same source risk, *i.e.* $\inf_{g\in\mathcal{G}} \mathrm{Err}_S(g\varphi) = \inf_{g\in\mathcal{G}} \mathrm{Err}_S(g\psi)$, and the same distribution divergence, *i.e.* $\delta_{\mathcal{G}}(\varphi) = \delta_{\mathcal{G}}(\psi)$, thus, φ has better theoretical guarantees if $\gamma_{\mathcal{G}}(\varphi,\psi) < \gamma_{\mathcal{G}}(\psi,\varphi)$, which a direct instantiation of Theorem 4.1. Additionally, we provide an overview of the behavior of the bound with respect to the relation of HCR in Table 4.1.

We are now ready to elucidate Example 4.1.1;

Example 4.2.1. *We consider the structural equation model from Example 4.1.1 with $\varphi_1(\mathbf{X}) := (\alpha\cdot X_1 + \beta\cdot X_2 + \gamma\cdot X_3, 0)$ and $\varphi_2(\mathbf{X}) := (\alpha\cdot X_1, \beta\cdot X_2 + \gamma\cdot X_3)$. As presented above φ_1 is reduced with respect to φ_2 ($\varphi_1 \prec \varphi_2$). We show that the reduction error allows to identify the better guarantees of φ_2 compared to φ_1. Indeed, $\gamma_{\mathcal{G}}(\varphi_1,\varphi_2) = +\infty$ while $\gamma_{\mathcal{G}}(\varphi_2,\varphi_1) = 0$ where $\mathcal{G} = \mathbb{R}^2$. The proof is simply obtained by considering $g = (g_1, g_2) \in \mathbb{R}^2$ and $g' = (0, g_2') \in \mathbb{R}^2$, resulting to $\inf_{g\in\mathcal{G}}[\mathrm{Err}_S(g\varphi, g'\psi) + \mathrm{Err}_T(g\varphi, g'\psi)] = g_2'^2(\beta^2+\gamma^2) \to +\infty$ when $w \to +\infty$ if both $\beta \neq 0$ and $\gamma \neq 0$. Conversely, $g = (g_1, g_2) \in \mathbb{R}^2$ and $g' = (g_1, 0) \in \mathbb{R}^2$, resulting to $\inf_{g\in\mathcal{G}}[\mathrm{Err}_S(g\varphi, g'\psi) + \mathrm{Err}_T(g\varphi, g'\psi)] = 0$. Thus, φ_2 has a better adaptabilty than φ_1.*

4.3 Applications

We provide, at the theoretical level, two applications of the *Hypothesis Class Reduction* (HCR) error. We first describe the case of a linear regression from a deep representation of inputs. The HCR error allows to prove the insight of (Chen et al. 2019c), reviewed in Section 3.4, that penalizes high singular values to improve transferability of representations. Furthermore, we describe a generic algorithm to boost domain invariant representations to improve their transferability.

4.3.1 Theoretical justification of (Chen et al. 2019c)

We establish connections between *Batch Spectral Penalization* (BSP) (Chen et al. 2019c) and the HCR error. The work (Chen et al. 2019c) had a significant impact onto our understanding of the trade-off between invariance and transferability, and suggests to improve transferability of representations using the Batch Spectral Penalization;

$$L_{\text{BSP}}(\varphi) := \sum_{i=1}^{k} \sigma_{i,S}(\varphi) + \sigma_{i,T}(\varphi) \tag{4.18}$$

where $\sigma_D^2(\varphi)$ is the eigen-value of the matrix $\mathbb{E}_D[\varphi(X)\varphi(X)^T]$ for $D \in \{S, T\}$. We have reviewed this work in Section 3.4.

To derive an analytical expression of HCR error, we study the bound from Theorem 4.2 in the particular case of least square linear regressor[3]. Importantly, our discussion focuses on the practical computation of the regressor when provided with a Ridge penalization;

$$g_\eta := \arg\min_{g \in \mathcal{G}} R(g\varphi)^2 + \eta ||g||^2 \tag{4.19}$$

where $R(g\varphi) := (\mathbb{E}[(Y - g\varphi(X))])^{1/2}$. Thus, our analysis is invalid beyond the scope of ridge regression but still provides interesting insights. Our exposition will be in three stages;

1. We modify slightly the theoretical bound from Theorem 4.2 to the case of a Ridge regression. In particular, we show that for two given representations $\varphi, \psi \in \Phi$ such that $\psi \prec \varphi$, $\gamma_{\mathcal{G}}(\varphi, \psi)$ may be different from zero, that will be the starting point of our discussion.
2. We work on a lower bound of the HCR risk that allows to derive a closed form of the ridge regressor.
3. We focus on the limit of HCR risk when the penalization is pushed to 0.

First, we mention that linear regression is performed, in practice, with a penalization that controls the norm of the model's parameters. We will consider the case of the ridge that allows deriving analytical expression of the learning objective;

$$\min_{g \in \mathcal{G}} R(g\varphi)^2 + \eta \cdot ||g||^2 \tag{4.20}$$

where $\eta > 0$ reflects the strength of the penalization. We will see this practical computation of the classifier leads us to fruitful insights. First, we revisit Theorem 4.2 in a context of Ridge regression, *i.e.* when all the infremum operation are obtained from a the penalized objective from Equation 4.20;

[3]Note that the bound remains valid when replacing Err by R where $R(g\varphi) := \mathbb{E}[(Y - g\varphi(X))^2]^{1/2}$

Theorem 4.3 (Bounding the target risk for Ridge Regression). *For two representations $\varphi, \psi \in \Phi$ and for $\eta > 0$, we note;*

$$g_\eta^\gamma := \arg\min_{g \in \mathcal{G}_1} R_{C'}(g\varphi, g'\varphi) + \eta \cdot ||g||^2 \tag{4.21}$$

$$g_\eta^\lambda := \arg\min_{g \in \mathcal{G}_1} R_{C'}(g\varphi) + \eta \cdot ||g||^2 \tag{4.22}$$

and $\mathcal{G}_1 := \{g \in \mathcal{G} : ||g|| \leq 1\}$. We assume that $g_\eta^\gamma, g_\eta^\lambda \in \mathcal{G}_1$. We have;

$$R_T(g\varphi) \leq R_S(g\varphi) + \delta_{\mathcal{G}_1}(\varphi) + \sup_{g \in \mathcal{G}_1} \left\{ R_C(g_\eta^\gamma \varphi, g'\varphi) \right\} + R_C(g_\eta^\lambda \varphi) \tag{4.23}$$

Proof. The proof follows the proof of Theorem 4.2. Note that $\sup_{g \in \mathcal{G}_1}$ is obtained under the assumption that $g_\eta^\gamma, g_\eta^\lambda \in \mathcal{G}_1$. Additionally, we apply the theorem in the particular case where $\psi = \varphi$. □

Let us describe this new bound. Here, the HCR risk is $\sup_{g \in \mathcal{G}_1} \left\{ R_C(g_\eta^\gamma \varphi, g'\varphi) \right\}$, and deviates from the original HCR error since the set of classifiers where the sup is computed differs from the set of classifiers where the inf is computed; $\mathcal{G}_1$ for the former while we use a penalized optimization for the latter. In particular, such HCR risk does not verify $\varphi \prec \psi$ implies a null HCR risk. This reflects the phenomenon where the classifier obtained from ridge regression is sub-optimal due to the penalization. Thus, we focus on the particular case where $\psi = \varphi$. Nevertheless, we will study this term in the limit where $\eta \to 0^+$, *i.e.* when the ridge regressor tends to the optimal regressor.

Second, to allow a closed form of the regressor obtained by ridge regression, we prove a lower bound of the HCR risk;

Proposition 4.3.1. *For two representations $\varphi, \psi \in \Phi$, we have for all $\eta > 0$;*

$$\sup_{g \in \mathcal{G}_1} \left\{ R_C(g_\eta^\gamma \varphi, g'\varphi) \right\} \geq \sup_{g \in \mathcal{G}_1} \sqrt{2} R_{C'}(g_\eta \varphi, g'\psi) \tag{4.24}$$

where $p_{C'} = \pi p_S + (1 - \pi) p_T$ where π is a Bernoulli variable such that $p(\pi = 1) = \frac{1}{2}$[4] *and;*

$$g_\eta := \arg\inf_{g \in \mathcal{G}} R_{C'}(g\varphi, g'\psi)^2 + \eta \cdot ||g||^2 \tag{4.25}$$

Proof. We first note that $R_S(g\varphi, g'\psi) + R_T(g\varphi, g'\psi) \geq \left(R_S(g\varphi, g'\psi)^2 + R_T(g\varphi, g'\psi)^2 \right)^{1/2}$. Now $R_S(g\varphi, g'\psi)^2 + R_T(g\varphi, g'\psi)^2 = \mathbb{E}_S[(g\varphi(X) - g'\psi(X)^2] + \mathbb{E}_T[(g\varphi(X) - g'\psi(X)^2] = 2\mathbb{E}_{C'}[(g\varphi(X) - g'\psi(X)^2]$. The supremum and inferemum operation conserve the inequality, leading to the stated result. □

Finally, we prove a result that allows to discuss the connection with (Chen et al. 2019c);

Proposition 4.3.2. *Under the assumption of Theorem 4.3;*

$$\sup_{g \in \mathcal{G}_1} \left\{ R_C(g_\eta^\gamma \varphi, g'\varphi) \right\} \geq \sqrt{2} \left(1 - \frac{\sigma^2_{C',-1}(\varphi)}{\sigma^2_{C',-1}(\varphi) + \eta} \right) \sigma_{C',-1}(\varphi) \tag{4.26}$$

[4] It is worth noting it differs from $R_C(g, g'\psi) = R_S(g\varphi, g'\psi) + R_T(g\varphi, g'\psi)$.

where $\sigma^2_{C',-1}(\varphi)$ *is the smallest eigen-value of* $\mathbb{E}_{C'}[\varphi(X)\varphi(X)^\top]$.

Proof. We first start by deriving a closed form of g_η given $Y = \varphi(X)^\top g'$, which a classical result of Ridge regression;

$$\arg\inf_{g\in\mathcal{G}} R_{C'}(g\varphi, Y) + \eta \cdot ||g||^2 = \underbrace{\left(\mathbb{E}\left[\varphi(X)\varphi(X)^\top\right] + \eta I_m\right)^{-1}}_{\in\mathbb{R}^{m\times m}} \underbrace{\mathbb{E}\left[\varphi(X)Y\right]}_{\in\mathbb{R}^{m\times 1}} \in \mathbb{R}^{m\times 1} \tag{4.27}$$

Now, by noting $M := \left(\mathbb{E}_{C'}\left[\varphi(X)\varphi(X)^\top\right] + \eta I_m\right)^{-1}\mathbb{E}_{C'}\left[\varphi(X)^\top\varphi(X)^\top\right]$we note that;

$$\varphi(X)^\top g' - \varphi(X)^\top g = \varphi(X)^\top g' - \varphi(X)^\top M g' = V(X)g' \tag{4.28}$$

where $V(X) = \varphi(X)^\top(I_m - M)$. Second, we note that;

$$M = P^{-1}\text{Diag}\left(1 - \frac{\sigma_{C'}(\varphi)^2}{\sigma_{C'}(\varphi)^2 + \eta}\right)P \tag{4.29}$$

where P is the diagonalization matrix of $\mathbb{E}_{C'}[\varphi(X)\varphi(X)^\top]$. Finally, we note that

$$R_{C'}(g_\eta\varphi, g'\psi)^2 = g'^\top\mathbb{E}_{C'}\left[V(X)^\top V(X)\right]g' \tag{4.30}$$

thus, $\arg\sup_{g'\in\mathcal{G},||g'||=1} R_{C'}(g_\eta\varphi, g'\psi)^2$ is the vector of $\mathbb{R}^m$ associated with the highest eigen-value of $\mathbb{E}_{C'}\left[V(X)^\top V(X)\right]$. Thus, noting ζ an eigen-value of $\mathbb{E}_{C'}\left[V(X)^\top V(X)\right]$, we have $\sup_{g\in\mathcal{G}_1} R_{C'}(g_\eta\varphi, g'\psi) \geq \sqrt{\zeta}$. In particular,

$$\sup_{g\in\mathcal{G}_1} R_{C'}(g_\eta\varphi, g'\psi) \geq \sqrt{\left(1 - \frac{\sigma^2_{C',-1}(\varphi)}{\sigma^2_{C',-1}(\varphi) + \eta}\right)^2 \sigma^2_{C',-1}(\varphi)} \tag{4.31}$$

which, when combined with Proposition 4.3.1, leads to the stated result. □

Through Proposition 4.3.2, we have succeeded in relating the eigen-values of $\mathbb{E}_{C'}[\varphi(X)\varphi(X)^\top]$ with the HCR risk. In particular, our lower bound depends on η which quantifies the strength of the penalization. Note that;

$$\lim_{\eta\to 0^+} \sqrt{2}\left(1 - \frac{\sigma^2_{C',-1}(\varphi)}{\sigma^2_{C',-1}(\varphi) + \eta}\right)\sigma_{C',-1}(\varphi) = 0 \tag{4.32}$$

which is consistent with a null HCR risk for an unpenalized regression since $\varphi \prec \varphi$. Now, let us provide a finer expression of the limit when $\eta \to 0^+$;

$$\left(1 - \frac{\sigma^2_{C',-1}(\varphi)}{\sigma^2_{C',-1}(\varphi) + \eta}\right)\sigma_{C',-1}(\varphi) \sim_{\eta\to 0^+} \frac{\eta}{\sigma_{C',-1}(\varphi)} \tag{4.33}$$

Thus, the HCR risk is directly related to $\frac{1}{\sigma'_C(\varphi)}$, *i.e.* the higher the smallest eigen-value, the lower will be the HCR risk. This theoretical result confirms the insight from BSP that aims to increase the smaller eigen-values of $\mathbb{E}_{C'}[\varphi(X)\varphi(X)^\top]$ in order to prevent the predominance of higher eigen-values.

4.3.2 Boosting Invariant Representations

We present a general algorithm for ensembling invariant representations with theoretical guarantees of improving the resulting adaptability. The main idea is that given two representations φ and ψ, the representation χ;

$$\chi : x \mapsto \varphi(x) \oplus \psi(x), \tag{4.34}$$

where $\oplus$ is the concatenation operator[5], has a better adaptability than both φ and ψ. We note this mechanically increases the dimensionality of the representation.

Proposition 4.3.3 (Concatenation of Invariant Representations). *Let $(\varphi, \psi) \in \Phi^2$, then;*

$$\lambda_{\mathcal{G}_2}(\varphi \oplus \psi) \leq \lambda_{\mathcal{G}_2}(\varphi, \psi) \tag{4.35}$$

where $\lambda_{\mathcal{G}}(\varphi, \psi) = \min\{\lambda_{\mathcal{G}}(\varphi), \lambda_{\mathcal{G}}(\psi)\}$, where $\mathcal{G}_2$ is a set of classifiers from $\mathcal{Z}^2$ to $\mathcal{Y}$.

Proof. We show that, $\lambda_{\mathcal{G}_2}(\varphi \oplus \psi) \leq \lambda_{\mathcal{G}}(\varphi)$, the second inequality is obtained by the symmetry of $\oplus$. First, we observe that $\lambda_{\mathcal{G}_2}(\varphi \oplus \mathbf{0}) \leq \lambda_{\mathcal{G}}(\varphi)$ where $\mathbf{0}$ is a the null vector of $\mathbb{R}^m$. Then, noting that $\mathcal{G}_2 \circ (\varphi \oplus \mathbf{0}) \subset \mathcal{G}_2 \circ (\varphi \oplus \psi)$, we obtain $\lambda_{\mathcal{G}_2}(\varphi \oplus \psi) \leq \lambda_{\mathcal{G}}(\varphi)$. □

This improvement of adaptability is paid at cost that χ may not remain invariant, even if both φ and ψ are invariant. However, a sufficient condition, here the independence between φ and ψ, ensures that χ remains invariant if both φ and ψ are invariant;

Proposition 4.3.4 (A sufficient condition for enforcing invariance of χ). *Let $(\varphi, \psi) \in \Phi^2$ such that both φ and ψ are invariant and independent for both the source and target distributions, then $\varphi \oplus \psi$ is invariant. In particular;*

$$\delta_{\mathcal{G}_2}(\varphi \oplus \psi) = 0. \tag{4.36}$$

Proof. φ and ψ are independent for both the source and target distributions, *i.e.* $p_D(\varphi(X), \psi(X)) = p_D(\varphi(X))p_D(\psi(X))$ for $D \in \{S, T\}$. φ and ψ are invariant so; $p_S(\varphi(X), \psi(X)) = p_S(\varphi(X))p_S(\psi(X)) = p_T(\varphi(X))p_T(\psi(X)) = p_T(\varphi(X), \psi(X))$, then $\varphi \oplus \psi$ is invariant. □

Building an algorithm that allows, given φ, to learn ψ which is domain invariant and independent from φ is a challenging problem that we leave for future works. In the following, we assume such an algorithm exists and we prove it allows to improve transferability of representations. We hope through this theoretical result to motivate future works in that direction.

Definition 4.3.1 (An algorithm for learning independent invariant representations). *An algorithm, noted `IndRep`, for learning independent invariant representations, takes as inputs a representation class Φ, source labelled data $\mathcal{D}_S$, target unlabelled data $\mathcal{U}_T$, a representation φ and returns a representation ψ that is;*

- *invariant i.e. $p_S(\psi(X)) = p_T(\psi(X))$;*
- *independent of φ i.e. $p_D(\varphi(X), \psi(X)) = p_D(\varphi(X))p_D(\psi(X))$ for $D \in \{S, T\}$.*

[5] If $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^{d'}$ then $x \oplus y \in \mathbb{R}^{d+d'}$.

We are now ready to describe the main result of this section that consists in building an algorithm that boosts transferability by ensembling of (independent) invariant representations. It is straightforward in form, given a representation φ, `IndRep` returns a representations ψ, thus we set $\varphi \leftarrow \varphi \oplus \psi$ and we iterate the process.

Theorem 4.4 (Boosting of domain invariant representations)**.** *Algorithm 1 improves transferability of domain invariant representations, i.e. noting;*

$$\varphi_i = \texttt{BIR}(\Phi, i, \mathcal{D}_S, \mathcal{U}_T), \ \textit{for} \ i \in \mathbb{N}^\star, \tag{4.37}$$

- $\delta_{\mathcal{G}_i}(\varphi_i) = \delta_{\mathcal{G}_{i+1}}(\varphi_{i+1}) = 0$,
- $\lambda_{\mathcal{G}_{i+1}}(\varphi_{i+1}) \leq \lambda_{\mathcal{G}_i}(\varphi_i)$.

where $\mathcal{G}_i$ is a set of classifiers from $\mathcal{Z}^i$ to $\mathcal{Y}$.

Algorithm 1 Boosting Invariant Representations (`BIR`).

Inputs:

- A representation class Φ,
- An integer N,
- Source labelled data $\mathcal{D}_S$,
- Target unlabelled data $\mathcal{U}_T$,
- An algorithm for learning invariant independent representation `IndRep` according to Definition 4.3.1).

Output: φ a representation.

$\varphi \leftarrow \texttt{IndRep}(\mathbf{0}, \mathcal{D}_S, \mathcal{U}_T)$
for $i \in \{1, ..., N\}$ **do**
  $\psi \leftarrow \texttt{IndRep}(\varphi, \mathcal{D}_S, \mathcal{U}_T)$
  $\varphi \leftarrow \varphi \oplus \psi$
end for
Return: φ

5 Representations and Weights

Contents

In Chapter 4, we have introduced a new error term, called the *Hypothesis Class Reduction* (HCR) error, in the influential theory of (Ben-David et al. 2010a). We have shown that the HCR error embodies the risk of information deletion in the representation space to achieve invariance. Thus, it provides a new criterion to identify, among invariant representations, the representation with the lowest adaptability error, thus assuring a better adaptation.

However, in some scenario of adaptation, it may not exist an invariant representation with a small adaptability error. Indeed, if it exists a significant shift between the source and the target distributions of labels, *i.e.* $p_S(Y) \neq p_T(Y)$, a situation referred to as *label shift* (see Section 3.2), we can not achieve invariance and a small adaptability error (Zhao et al. 2019), as presented in Section 3.4.

Alongside the paradigm of invariance, the situation of *label shift* has been the subject of abundant literature, namely through *Importance Sampling* (see Section 3.2.3) that aims to weight the contribution of the source sample in the classification loss (Quinonero-Candela et al. 2009). In particular, the assumption of *Covariate Shift* $p_T(y|x) = p_S(y|x)$ allows to reduce the problem of adaptation to the problem of computing the density ratio of inputs $w(x) = p_T(x)/p_S(x)$ (Shimodaira 2000; Huang et al. 2007). Although IS seems natural, the covariate shift assumption is not sufficient to guarantee successful adaptation (Ben-David et al. 2010b). Moreover, for high dimensional data such as texts or images, the shift between $p_S(x)$ and $p_T(x)$ results from non-overlapping supports (D'Amour et al. 2021) leading to unbounded weights (Johansson, Sontag, and Ranganath 2019).

Historically, the Invariance-based and Importance Sampling-based lines of study have developed independently. To our knowledge, no theory offers a unified view of

these two paradigms. This chapter, therefore, aims to construct such a theory; given a representation φ and some weights w in the representation space, what guarantee can we obtain from aligning a *weighted source* distribution with the target distribution $w(z)p_S(z) \approx p_T(z)$? Indeed, 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?

We introduce a new bound of the target error which incorporates both weights and domain invariant representations. Two new terms are introduced. The first is an *invariance term* which promotes alignment between a weighted source distribution of representations and the target distribution of representations. The second, named *transferability term*, involves labelling functions from both source and target domains. We show that weights allow to design an interpretable generalization bound where transferability and invariance errors are well-characterized in the bound.

Chapter 5 is based on the publication (Bouvier et al. 2020b) in an international conference;

> Robust Domain Adaptation: Representations, Weights and Inductive Bias
> *Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami and Céline Hudelot*,
> European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Ghent (Belgium), Online, 2020.

and covers Sections 2 and 3 of (Bouvier et al. 2020b).

5.1 Preliminaries

5.1.1 Overall Strategy

Our strategy is to express both the *transferability* and *invariance* as a supremum using Integral Probability Measure (IPM) computed on a large critic class which will be the set of measurable functions. Pragmatically, this reflects the situation where the domain discriminator has infinite capacity to discriminate the source from the target domain. On the one hand, relying on a large critic class leads to an over-pessimistic estimation of the different terms involved when bounding this error. On the other hand, this deliberate choice allows to exploit symmetries in the critic class leading to drastic simplifications in error terms. We will rely on this simplification to better understand, through more interpretable error terms, the problem of transferability of domain invariant representations. In particular, we deviate from the analysis of Chapter 4, where we emphasized the case of linear classifier namely in Section 4.3.1, by considering the case of infinite capacity classifiers.

Notations. We fix a representation $\varphi \in \Phi$ where Φ is the set of representations. We note $\mathcal{M}(\mathcal{Z}, P(\{0,1\}^C))$ the set of measurable function from $\mathcal{Z}$ to $P\left(\{0,1\}^C\right)$, where $P\left(\{0,1\}^C\right) = \{y \in \{0,1\}^C, \sum_{c=1}^C y_c = 1\}$, *i.e.* the set of one-hot vectors of $\mathbb{R}^C$. $\mathbf{Y}$ is the one-hot encoded version of Y, *i.e.* $\mathbf{Y} \in P(\{0,1\}^C)$ such that $\mathbf{Y}_Y = 1$[1]. For $z \in \mathcal{Z}$ and $g \in \mathcal{G}$, we note $\mathbf{g}(z)$ the vector of $\mathbb{R}^C$ where $\mathbf{g}(z)_{g(z)} = 1$ and 0 otherwise.

The set of infinite classifiers $\mathcal{G}$ is the set of measurable functions from $\mathcal{Z}$ to $\mathcal{Y}$, in particular;

$$\{z \mapsto \mathbf{g}(z), g \in \mathcal{G}\} \subset \mathcal{M}(\mathcal{Z}, P(\{0,1\}^C)) \tag{5.1}$$

We note the *labelling classifier* from representation $g_D := \arg\min_{g \in \mathcal{G}} \mathrm{Err}_D(g\varphi)$ for $D \in \{S, T\}$. We introduce an important tool of this chapter, noted $\mathcal{M}(\mathcal{Z}, [-1,1]^C)$, which is the set of measurable functions from $\mathcal{Z}$ to $[-1,1]^C$. Crucially, we will rely on the fact that $\mathcal{M}(\mathcal{Z}, P(\{0,1\}^C)) \subset \mathcal{M}(\mathcal{Z}, [-1,1]^C)$. For the ease of reading, we note $\mathcal{F}_C := \mathcal{M}(\mathcal{Z}, [-1,1]^C)$ and $\mathcal{F} := \mathcal{M}(\mathcal{Z}, [-1,1])$, the set of measurable functions from $\mathcal{Z}$ to $[-1,1]$. Besides, we define the *labelling function* $\mathbf{f}_D : z \mapsto \mathbb{E}_D[\mathbf{Y}|Z = z] \in \mathcal{F}_C$ for $D \in \{S, T\}$, where $\mathbf{Y}$ is the one-hot encoded version of Y, *i.e.* $\mathbf{Y} \in P(\{0,1\}^C)$ such that $\mathbf{Y}_Y = 1$. Thus one should differentiate from the labelling classifier g and the labelling function $\mathbf{f}$; given $z \in \mathcal{Z}$, $\mathbf{f}(z)_y$ is the probability that the representation z has label y.

Main arguments of the proofs. For the theoretical development of this Chapter, we will use recurrent arguments during the proofs that we describe here;

- **(A1) We express the error as a L^2 norm.** Given $g, g' \in \mathcal{G}$, we observe that;

$$\mathbb{I}(g(z) \neq g'(z)) = \frac{1}{2}||\mathbf{g}(z) - \mathbf{g}'(z)||^2_{\mathbb{R}^C} = \frac{1}{2}\sum_{c=1}^{C}\left(\mathbf{g}(z)_c - \mathbf{g}'(z)_c\right)^2 \tag{5.2}$$

  where $||\cdot||^2_{\mathbb{R}^C}$ is the L^2 norm of $\mathbb{R}^C$. Indeed, when $g(z) = g'(z)$ then $\mathbf{g}(z) = \mathbf{g}'(z)$ leading to $\frac{1}{2}||\mathbf{g}(z) - \mathbf{g}'(z)||^2_{\mathbb{R}^C} = 0$, when $g(z) \neq g'(z)$ then $||\mathbf{g}(z) - \mathbf{g}'(z)||^2_{\mathbb{R}^C} = 2$ since $\mathbf{g}(z)$ and $\mathbf{g}'(z)$ takes the value 1 in different dimensions.

[1] Note that we deviate from the setup of Chapter 3 since we consider the case of multi-class classification.

- **(A2) We practice a pessimistic bounding strategy.** More specifically, we observe that $\mathcal{M}(\mathcal{Z}, P(\{0,1\}^C)) \subset \mathcal{F}_C$ the set of measurable functions from $\mathcal{Z}$ to $[-1,1]^C$. To illustrate how this property will be exploited in practice, let us consider the following simple example;

$$\sup_{(g,g')\in\mathcal{G}} \mathbb{E}\left[\mathbb{I}(g(z) \neq g(z')\right] \leq \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}\left[\frac{1}{2}||\mathbf{f}(z) - \mathbf{f}(z')||^2\right] \tag{5.3}$$

  Such bounding strategy simply derives from $\text{Err}(g,g') = \mathbb{E}\left[\mathbb{I}(g(z) \neq g(z')\right] = \mathbb{E}\left[\frac{1}{2}||\mathbf{g}(z) - \mathbf{g}(z')||^2\right]$ and noting $\{z \mapsto \mathbf{g}(z), g \in \mathcal{G}\} \subset \mathcal{M}(\mathcal{Z}, P(\{0,1\}^C)) \subset \mathcal{F}_C$.

- **(A3) We exploit symmetries in $\mathcal{F}_C$.** It leads to drastic simplification during the derivation of the bounds. More specifically;
  - (A.3.1) $\{\frac{1}{2}(\mathbf{f} - \mathbf{f}'), (\mathbf{f}, \mathbf{f}') \in \mathcal{F}_C^2\} \subset \mathcal{F}_C$, *i.e.* the difference between two measurable functions of $\mathcal{F}_C$, with a scaling factor of $\frac{1}{2}$, is a measurable function. This brings back a supremum on $\mathcal{F}_C^2$ to a supremum on $\mathcal{F}_C$.
  - (A.3.2) $\{\frac{1}{C}||\mathbf{f}||^2, \mathbf{f} \in \mathcal{F}_C\}$, *i.e.* the norm of a measurable function in $\mathcal{F}_C$, with a scaling factor of $\frac{1}{C}$, is a measurable function in $\mathcal{F}$. This brings back a supremum of a $||\mathbf{f}||^2_{\mathbb{R}^C}$ for $\mathbf{f} \in \mathcal{F}_C$ as a supremum on $\mathcal{F}$.
  - (A.3.3) $\{\frac{1}{C}\mathbf{f} \cdot \mathbf{f}', (\mathbf{f}, \mathbf{f}') \in \mathcal{F}_C^2\} \subset \mathcal{F}$, *i.e.* the scalar product between two measurable functions in $\mathcal{F}_C$, with a scaling factor of $\frac{1}{C}$, is a measurable function in $\mathcal{F}$. This brings back a supremum of a $||\mathbf{f} \cdot \mathbf{f}'||^2_{\mathbb{R}^C}$ for $(\mathbf{f}, \mathbf{f}') \in \mathcal{F}_C^2$ as a supremum on $\mathcal{F}$.
- **(A4) We exploit an Integral Probability Metric.** An *Integral Probability Metric* (IPM, (Müller 1997)) is a tool to compare two distributions. The bigger the IMP, the more dissimilar are the two distributions. More specifically, we observe that for two distributions $p_S(Z)$ and $p_T(Z)$ on $\mathcal{Z}$, $p_S = p_T$ if and only if;

$$\text{IPM}(p_S, p_T; \mathcal{F}) := \sup_{f\in\mathcal{F}} \{\mathbb{E}_S[f(Z)] - \mathbb{E}_T[f(Z)]\} = 0 \tag{5.4}$$

By relating the error with a L^2 norm of a measurable function that takes values in $[-1,1]^C$ (argument A1), we will rely on a large class of critics ($\mathcal{F}_C$ and $\mathcal{F}$), from argument A2, which enables symmetries (argument A3), leading to a drastic simplification of bounding terms. Thus, despite being over-pessimistic, our analysis allows us to derive simple, thus more interpretable, terms when bounding the target risk. We will refer to $\mathcal{F}_C$ and $\mathcal{F}$ as critic classes.

5.1.2 Two errors as IPMs

We introduce here two important tools that will guide our analysis:

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

$$\text{INV}(\varphi) := \sup_{f\in\mathcal{F}} \{\mathbb{E}_T[f(Z)] - \mathbb{E}_S[f(Z)]\} \tag{5.5}$$

- TSF(φ), named *transferability error*, which 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 $\mathbf{g}_D(Z) \cdot \mathbf{f}(Z)$, where $\mathbf{f} \in \mathcal{F}_C$ and $\mathbf{g}_D(Z) \cdot \mathbf{f}(Z)$ is the scalar product $\mathbf{g}_D(Z)$ between $\mathbf{f}(Z)$, for $D \in \{S, T\}$, where g_D is the best classifier in domain D;

$$\text{TSF}(\varphi) := \sup_{\mathbf{f} \in \mathcal{F}_C} \{\mathbb{E}_T[\mathbf{g}_T(Z) \cdot \mathbf{f}(Z)] - \mathbb{E}_S[\mathbf{g}_S(Z) \cdot \mathbf{f}(Z)]\} \tag{5.6}$$

5.2 Invariance and Transferability

5.2.1 A new bound of the target risk

Using INV(φ) and TSF(φ), we can provide a new bound of the target risk:

Proposition 5.2.1. *$\forall g \in \mathcal{G}$ and $\forall \varphi \in \Phi$:*

$$\text{Err}_T(g\varphi) \leq \text{Err}_S(g\varphi) + 3C \cdot \text{INV}(\varphi) + \text{TSF}(\varphi) + \text{Err}_T(g_T\varphi) \tag{5.7}$$

To prove such a result, we first prove the following lemma:

Proposition 5.2.2. *$\forall g \in \mathcal{G}$:*

$$\text{Err}_T(g\varphi) \leq \text{Err}_S(g\varphi) + \delta_{\mathcal{F}_C}(\varphi) + \text{Err}_T(g_S\varphi, g_T\varphi) + \text{Err}_T(g_T\varphi) \tag{5.8}$$

where $\delta_{\mathcal{F}_C}(\varphi) := \sup_{(\mathbf{f},\mathbf{f}') \in \mathcal{F}_C^2} \mathbb{E}_T\left[\frac{1}{2}||\mathbf{f}\varphi(X) - \mathbf{f}'\varphi(X)||^2\right] - \mathbb{E}_S\left[\frac{1}{2}||\mathbf{f}\varphi(X) - \mathbf{f}'\varphi(X)||^2\right]$.

Proof. We first observe the following triangular inequalites:

$$\begin{aligned}
\text{Err}_T(g\varphi) &\leq \text{Err}_T(g_T\varphi) + \text{Err}_T(g\varphi, g_T\varphi) \\
&\leq \text{Err}_T(g_T\varphi) + \text{Err}_T(g\varphi, g_S\varphi) + \text{Err}_S(g_S\varphi, g_T\varphi) \\
&\leq \text{Err}_T(g_T\varphi) + \text{Err}_S(g\varphi, g_S\varphi) + \text{Err}_T(g\varphi, g_S\varphi) - \text{Err}_S(g\varphi, g_S\varphi) + \text{Err}_S(g_S\varphi, g_T\varphi)
\end{aligned}$$

We note that;

$$\begin{aligned}
&|\text{Err}_T(g\varphi, g_S\varphi) - \text{Err}_S(g\varphi, g_S\varphi)| \\
&\quad = \left|\mathbb{E}_S\left[\frac{1}{2}||\mathbf{g}\varphi(X) - \mathbf{g}_S\varphi(X)||^2\right] - \mathbb{E}_T\left[\frac{1}{2}||\mathbf{g}\varphi(X) - \mathbf{g}_S\varphi(X)||^2\right]\right| \quad \text{(from A1)} \\
&\quad \leq \sup_{(\mathbf{f},\mathbf{f}') \in \mathcal{F}_C^2} \left|\mathbb{E}_S\left[\frac{1}{2}||\mathbf{f}\varphi(X) - \mathbf{f}'\varphi(X)||^2\right] - \mathbb{E}_T\left[\frac{1}{2}||\mathbf{f}\varphi(X) - \mathbf{f}'\varphi(X)||^2\right]\right| \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(from A2)} \\
&\quad = \delta_{\mathcal{F}_C}(\varphi)
\end{aligned}$$

and we use the property of conditional expectation $\text{Err}_S(g\varphi, g_S\varphi) \leq \text{Err}_S(g\varphi)$. □

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

Proposition 5.2.3. $\delta_{\mathcal{F}_C}(\varphi) \leq 2C \cdot \text{INV}(\varphi)$.

Proof. First, we note that $(\mathbf{f} - \mathbf{f}') = 2\mathbf{f}'' \in \mathcal{F}_C$ for some $\mathbf{f}'' \in \mathcal{F}_C$, thus using A.3.1, $\delta_{\mathcal{F}_C}(\varphi) \leq \sup_{\mathbf{f}'' \in \mathcal{F}_C} \mathbb{E}_S\left[\frac{1}{2}||2\mathbf{f}''\varphi(X)||^2\right] - \mathbb{E}_T\left[\frac{1}{2}||2\mathbf{f}''\varphi(X)||^2\right] = 2\sup_{\mathbf{f} \in \mathcal{F}_C} \mathbb{E}_S\left[||\mathbf{f}\varphi(X)||^2\right] - \mathbb{E}_T\left[||\mathbf{f}\varphi(X)||^2\right]$. Second, we note that for $\mathbf{f} \in \mathcal{F}_C$, $||\mathbf{f}||^2 = Cf$ for some $f \in \mathcal{F}$, thus

using A.3.2, $\delta_{\mathcal{F}_C}(\varphi) \leq 2\sup_{f\in\mathcal{F}} \mathbb{E}_S[Cf\varphi(Z)] - \mathbb{E}_T[Cf\varphi(Z)]$, leading to the stated result. □

Third, we bound $\mathrm{Err}_T(g_S\varphi, g_T\varphi)$.

Proposition 5.2.4. $\mathrm{Err}_T(g_S\varphi, g_T\varphi) \leq C \cdot \mathrm{INV}(\varphi) + \mathrm{TSF}(\varphi)$.

Proof. First, $\mathrm{Err}_T(g_S\varphi, g_T\varphi) = \mathbb{E}_T\left[\frac{1}{2}||\mathbf{g}_S\varphi - \mathbf{g}_T\varphi||^2\right]$ from A1. We note $\Delta = \mathbf{g}_T - \mathbf{g}_S$ and we omit φ for the ease of reading

$$\begin{aligned}
2\mathrm{Err}_T(g_S, g_T) &= \mathbb{E}_T\left[||\Delta||^2\right] && \text{(from A1)}\\
&= \mathbb{E}_T\left[(\mathbf{g}_T - \mathbf{g}_S)\cdot\Delta\right]\\
&= \mathbb{E}_T[\mathbf{g}_T\cdot\Delta] - \mathbb{E}_T[\mathbf{g}_S\cdot\Delta]\\
&= (\mathbb{E}_T[\mathbf{g}_T\cdot\Delta] - \mathbb{E}_S[\mathbf{g}_S\cdot\Delta]) + (\mathbb{E}_S[\mathbf{g}_S\cdot\Delta] - \mathbb{E}_T[\mathbf{g}_S\cdot\Delta])
\end{aligned}$$

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

$$\begin{aligned}
\mathbb{E}_S[\mathbf{g}_S\cdot\Delta] - \mathbb{E}_T[\mathbf{g}_S\cdot\Delta] &\leq \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[\mathbf{f}\cdot(\mathbf{g}_T - \mathbf{g}_S)] - \mathbb{E}_T[\mathbf{f}\cdot(\mathbf{g}_T - \mathbf{g}_S)]\\
&&& (\mathbf{g}_D \in \mathcal{F}_C \text{ and A2})\\
&\leq \sup_{(\mathbf{f},\mathbf{f}',\mathbf{f}'')\in\mathcal{F}_C^3} \mathbb{E}_S[\mathbf{f}\cdot(\mathbf{f}' - \mathbf{f}'')] - \mathbb{E}_T[\mathbf{f}\cdot(\mathbf{f}' - \mathbf{f}'')]\\
&&& (\mathbf{g}_D \in \mathcal{F}_C \text{ and A2})\\
&\leq 2\left(\sup_{(\mathbf{f},\mathbf{f}')\in\mathcal{F}_C^2} \mathbb{E}_S[\mathbf{f}\cdot\mathbf{f}'] - \mathbb{E}_T[\mathbf{f}\cdot\mathbf{f}']\right) && \text{(from A.3.1)}\\
&\leq 2C\left(\sup_{f\in\mathcal{F}} \mathbb{E}_S[f] - \mathbb{E}_T[f]\right) && \text{(from A.3.3)}\\
&= 2C\cdot\mathrm{INV}(\varphi)
\end{aligned}$$

   The inequalities above are obtained as follows. We first use the fact that $\mathbf{g}_D$ is in $\mathcal{F}_C$, then we use the argument of A.2. The second inequality is obtained using similar arguments. Third, we use A.3.1. Finally, we use A.3.3, thus we identify $\mathrm{INV}(\varphi)$.

2. Second, we relate $\mathbb{E}_T[\mathbf{g}_T\cdot\Delta] - \mathbb{E}_S[\mathbf{g}_S\cdot\Delta]$ to $\mathrm{TSF}(\varphi)$;

$$\begin{aligned}
\mathbb{E}_T[\mathbf{g}_T\cdot\Delta] - \mathbb{E}_S[\mathbf{g}_S\cdot\Delta] &\leq \sup_{(\mathbf{f},\mathbf{f}')\in\mathcal{F}_C^2} \mathbb{E}_T[\mathbf{g}_T\cdot(\mathbf{f} - \mathbf{f}')] - \mathbb{E}_S[\mathbf{g}_S\cdot(\mathbf{f} - \mathbf{f}')]\\
&&& (\mathbf{g}_D \in \mathcal{F}_C)\\
&\leq 2\sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_T[\mathbf{g}_T\cdot\mathbf{f}] - \mathbb{E}_S[f\mathbf{g}_S\cdot\mathbf{f}] = \mathrm{TSF}(\varphi)\\
&&& \text{(from A.2.1)}
\end{aligned}$$

   The inequalities above are obtained as follows. We first use the fact that $\mathbf{g}_D \in \mathcal{F}_C$ for $D \in \{S, T\}$. Second, we use A.3.1, finally we identify $\mathrm{TSF}(\varphi)$.

leading to the stated result. □

Here two IPMs are involved to compare representations (INV(φ) and TSF(φ)). A new term, $\text{Err}_T(g_T\varphi)$, reflects the level of noise when fitting labels from representations using an infinite capacity classifier class. In particular, if both $\text{Err}_T(g_T\varphi) = 0$ and $\text{Err}_S(g_S\varphi) = 0$, then;

$$\text{TSF}(\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_T[\mathbf{Y}\cdot\mathbf{f}(Z)] - \mathbb{E}_S[\mathbf{Y}\cdot\mathbf{f}(Z)] \tag{5.9}$$

Our bound does not evade from the fundamental trade-off in UDA (Ben-David et al. 2010a) as described in Section 3.4.2. Indeed, a new trade-off now operates;

Proposition 5.2.5 (A new trade-off)**.** *Let two representations $\varphi, \psi \in \Phi^2$, two representations such that $\mathcal{F}\circ\varphi \subset \mathcal{F}\circ\psi$ and $\mathcal{G}\circ\varphi \subset \mathcal{G}\circ\psi$, then;*

$$\text{INV}(\varphi) \leq \text{INV}(\psi) \text{ while } \text{Err}_T(g_T^\psi\psi) \leq \text{Err}_T(g_T^\varphi\varphi) \tag{5.10}$$

where $g_D^\varphi := \arg\min_{g\in\mathcal{G}} \text{Err}_D(g\varphi)$, and $g_D^\psi := \arg\min_{g\in\mathcal{G}} \text{Err}_D(g\psi)$ for $D \in \{S, T\}$.

Proof. First, $\text{INV}(\varphi) \leq \text{INV}(\psi)$ a property of the supremum applied to $\mathcal{F}\circ\varphi \subset \mathcal{F}\circ\psi$. Second, $\text{Err}_T(g_T^\psi\psi) \leq \text{Err}_T(g_T^\varphi\varphi)$ is a property of the infremum applied to $\mathcal{G}\circ\varphi \subset \mathcal{G}\circ\psi$. □

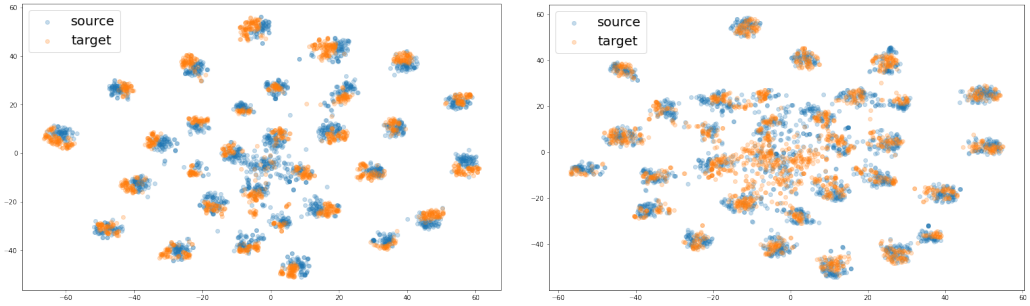
Bounding the target risk using IPMs has two advantages. First, it allows to better control the invariance / transferability trade-off since $\text{Err}_T(\mathbf{f}_T\varphi) \leq \lambda_\mathcal{G}(\varphi)$. This is paid at the cost of $2C\cdot\text{INV}(\varphi) \geq \delta_\mathcal{G}(\varphi)$. Second, $\text{Err}_T(g_T\varphi)$ 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, which is then a similar role than adaptability from Theorem 3.3. To illustrate the interest of our new transferability error, we provide visualisation of representations (Fig. 5.1) when trained to minimize the adaptability error $\lambda_\mathcal{G}(\varphi)$ from Theorem 3.4 and the transferability error TSF(φ) from Proposition 5.2.1.

5.2.2 A detailed view on the property of tightness

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

Proposition 5.2.6 (Tightness of Invariance and Transferability)**.** $\text{INV}(\varphi) = \text{TSF}(\varphi) = 0$ *if and only if* $p_T(Z) = p_S(Z)$ *and* $g_S = g_T$*. Furthermore, if additionally* $\text{Err}_S(g_S\varphi) = \text{Err}_T(g_T\varphi) = 0$*, then* $p_S(Y, Z) = p_T(Y, Z)$*.*

Proof. First, $\text{INV}(\varphi) = 0$ implies $p_T(Z) = p_S(Z)$ which is a direct application of A4. Now $\text{TSF}(\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)] - \mathbb{E}_T[\mathbf{g}_T(Z)\cdot\mathbf{f}(Z)] = \mathbb{E}_S[\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)] - \mathbb{E}_S[\mathbf{g}_T(Z)\cdot\mathbf{f}(Z)] = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[(\mathbf{g}_S - \mathbf{g}_T)(Z)\cdot\mathbf{f}(Z)]$. For the particular choice of $\mathbf{f} = \frac{1}{2}(\mathbf{g}_S - \mathbf{g}_T)$ leads to $\mathbb{E}_S[||\mathbf{g}_S - \mathbf{g}_T||^2]$ then $\mathbf{g}_S = \mathbf{g}_T$, p_S, then p_T, almost surely. Now, if we assume that $\text{Err}_S(g_S\varphi) = \text{Err}_T(g_T\varphi) = 0$, then $\text{TSF}(\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_T[\mathbf{Y}\cdot\mathbf{f}(Z)] - \mathbb{E}_S[\mathbf{Y}\cdot\mathbf{f}(Z)] = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_T[\mathbf{f}_T(Z)\cdot\mathbf{f}(Z)] - \mathbb{E}_S[\mathbf{f}_S(Z)\cdot\mathbf{f}(Z)]$ which leads to $\mathbf{f}_s = \mathbf{f}_T$, p_S, then p_T, almost surely, following similar argument. Noting that $\mathbb{E}_D[\mathbf{Y}|Z = z]_c = p_D(Y = c|Z = z)$ for $c \in \{1, ..., C\}$, leading to $p_T(Y|Z) = p_S(Y|Z)$, then $p_S(Y, Z) = p_T(Y, Z)$ when combined with $p_S(Z) = p_T(Z)$. □



(a) $\lambda_{\mathcal{G}}(\varphi)$ adaptability from Theorem 3.3. Inside class clusters, source and target representations are separated.

(b) TSF(φ) transferability from bound 5.2.1 (contribution). Inside class clusters, source and target representations are not distinguishable

Figure 5.1: t-SNE (Maaten and Hinton 2008) visualisation of representations when trained to minimize (a) adaptability error $\lambda_{\mathcal{G}}(\varphi)$ from Theorem 3.4, (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.

Two important points should be noted:

1. INV$(\varphi) = 0$ ensures that $p_S(z) = p_T(z)$, using (A4). Similarly, TSF$(\varphi) = 0$, when combined with INV$(\varphi) = 0$, leads to $g_S = g_T$. Additionally, one can show that TSF$(\varphi) \geq$ INV(φ) noting that $\mathbf{g}_D(Z) \cdot \mathbf{f}(Z) = f(z)$ when $\mathbf{f}(z) = (f(z), ..., f(z))$ for $f \in \mathcal{F}$.

2. Second, in the particular situation of tightness where we obtain the equality $p_S(Y, Z) = p_T(Y, Z)$, it also implies that $p_S(Y) = p_T(Y)$. Therefore, in the context of label shift (when $p_S(Y) \neq p_T(Y)$), such contradiction shows that INV(φ) and TSF(φ) can not be null simultaneously. This bound highlights the fact that representations alone can not address UDA in complex settings such as the label shift one.

5.3 The role of Weights

5.3.1 Reconciling weights and representations

Based on the interesting observations from (Johansson, Sontag, and Ranganath 2019; Zhao et al. 2019), that we reviewed in Section 3.4, and following the line of study that proposed to relax invariance using weights (Cao et al. 2018; Zhang et al. 2018; You et al. 2019; Wu et al. 2019), 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 Proposition 5.3.1, we obtain a new bound of the target risk incorporating both weights and representations:

Proposition 5.3.1. $\forall g \in \mathcal{G}, \forall w : \mathcal{Z} \to \mathbb{R}^+$ *such that* $\mathbb{E}_S[w(Z)] = 1$*:*

$$\mathrm{Err}_T(g\varphi) \leq \mathrm{Err}_{w\cdot S}(g\varphi) + 3C \cdot \mathrm{INV}(w, \varphi) + \mathrm{TSF}(w, \varphi) + \mathrm{Err}_T(g_T\varphi)$$

where;

$$\mathrm{INV}(w, \varphi) := \sup_{f \in \mathcal{F}} \{\mathbb{E}_T[f(Z)] - \mathbb{E}_S[w(Z)f(Z)]\} \tag{5.11}$$

$$\mathrm{TSF}(w, \varphi) := \sup_{\mathbf{f} \in \mathcal{F}_C} \{\mathbb{E}_T[\mathbf{g}_T(Z) \cdot \mathbf{f}(Z)] - \mathbb{E}_S[w(Z)\mathbf{g}_S(Z) \cdot \mathbf{f}(Z)]\} \tag{5.12}$$

5.4 Analysis of tightness

As for the previous Proposition 5.2.1, the property of tightness *i.e.* when invariance and transferability are null simultaneously, leads to interesting observations:

Proposition 5.4.1. $\mathrm{INV}(w, \varphi) = \mathrm{TSF}(w, \varphi) = 0$ *if and only if* $w(z) = \frac{p_T(z)}{p_S(z)}$ *and* $f_S = f_T$*. Furthermore, if additionally* $\mathrm{Err}_S(g_S\varphi) = \mathrm{Err}_T(g_T\varphi) = 0$*, then* $p_S(Y|Z) = p_T(Y|Z)$*.*

Proof. It follows exactly the same proof than Proposition 5.2.1 changing the source distribution into the source weighted distribution. Thus, we obtain $w(Z)p_S(Z) = p_T(Z)$ which leads to $w(Z) = \frac{p_T(Z)}{p_S(Z)}$ when $p_S(Z) \neq 0$. Additionally, when $\mathrm{Err}_S(g_S\varphi) = \mathrm{Err}_T(g_T\varphi) = 0$, we have $p_{w\cdot S}(Y, Z) = p_T(Y, Z)$, where $p_{w\cdot S}(Y, Z) = p_{w\cdot S}(Z)p_{w\cdot S}(Y|Z) = w(Z)p_S(Z)p_S(Y|Z)$. From the equality $w(Z)p_S(Z)p_S(Y|Z) = p_T(Y, Z)$ and noting $w(Z)p_S(Z) = p_T(Z)$ which implies $p_S(Y|Z) = p_T(Y, Z)/p_T(Z) = p_T(Y|Z)$. □

This proposition means that the nullity of invariance error, *i.e.* $\mathrm{INV}(w, \varphi) = 0$, 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.* $\mathrm{TSF}(w, \varphi) = 0$, implies that labelling functions, $\mathbf{f} : z \mapsto \mathbb{E}[Y|Z = z]$, are conserved across domains. Furthermore, the equality $p_T(Y|Z) = p_T(Y|Z)$ interestingly resonates with a recent line of work called *Invariant Risk Minimization* (IRM) (Arjovsky et al. 2019) that we reviewed in Section 3.3.3. Incorporating weights in the bound thus brings two benefits:

1. First, it raises the inconsistency issue of invariant representations in presence of label shift. Indeed, tightness is not conflicting with label shift.
2. $\mathrm{TSF}(w, \varphi)$ and $\mathrm{INV}(w, \varphi)$ have two disctinct roles: the former promotes domain invariance of representations while the latter controls whether aligned representations share the same labels across domains.
3. Given a representation, one can achieve make the invariance term null by choosing weights as follows; $w(z) = p_T(z)/p_S(z)$ (Proposition 5.4.1). Such a property will motivate our first contribution dedicated to applications of this thesis (Chapter 7). Note that choosing specific weights do not guarantee to make the transferability term null.

5.5 From IPM to Domain Adversarial Objective

Our analysis builds the invariance and the transferability term as an *Integral Probability Measure* (IPM) through a supremum on the measurable function. To adapt

this term in a context of domain adversarial learning, we recall the connections with f−divergence for comparing distributions, where domain adversarial loss is a particular instance. This connection is motivated by the furnished literature on adversarial learning and follows the work (Bottou et al. 2018) where such connections are established in a context of generative modelling. This section is then an informal attempt to transport our theoretical analysis, which holds for IPM, to f−divergence. Given f a function defined on $\mathbb{R}^+$, continuous and convex, the f−divergence between two distributions p and q: $\mathbb{E}_p[\mathsf{f}(p/q)]$, is null if and only if $p = q$. Interestingly, f−divergence admits an '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 (Ganin and Lempitsky 2015) is a particular instance of f−divergence. Then, we informally transports our analysis on IPM distance to domain adversarial loss. More precisely, we define:

$$\mathrm{INV}_{\mathrm{adv}}(w, \varphi) := \log(2) - \sup_{d \in \mathcal{D}} \mathbb{E}_S[w(Z)\log(d(Z))] + \mathbb{E}_T[\log(1 - d(Z))] \tag{5.13}$$

$$\mathrm{TSF}_{\mathrm{adv}}(w, \varphi) := \log(2) - \sup_{\mathbf{d} \in \mathcal{D}_C} \mathbb{E}_S[w(Z)\mathbf{Y} \cdot \log(\mathbf{d}(Z))] + \mathbb{E}_T[\mathbf{Y} \cdot \log(1 - \mathbf{d}(Z))] \tag{5.14}$$

where $\mathcal{D}$ is the well-established domain discriminator from $\mathcal{Z}$ to $[0, 1]$, and $\mathcal{D}_C$ is the set of *label domain discriminator* from $\mathcal{Z}$ to $[0, 1]^C$. For practical applications that we will develop in III, we will rely on the following losses;

$$L_{\mathrm{INV}}(w, \varphi, d) := \mathbb{E}_S[w(Z)\log(d(Z))] + \mathbb{E}_T[\log(1 - d(Z))] \tag{5.15}$$

$$L_{\mathrm{TSF}}(w, \varphi, g, \mathbf{d}) := \mathbb{E}_S[w(Z)\mathbf{Y} \cdot \log(\mathbf{d}(Z))] + \mathbb{E}_T[g(Z) \cdot \log(1 - \mathbf{d}(Z))] \tag{5.16}$$

where $g \in \mathcal{G}_{\mathrm{p}}$ where $\mathcal{G}_{\mathrm{p}}$ is the set of probability classifiers, *i.e.* a subset of measurable functions from $\mathcal{Z}$ to $\mathcal{P}(\mathcal{Y})$. Although we intend to guarantee as much as possible harmony in the notations, the notations may deviate slightly in the next chapters for the purpose of the exposition.

6 The Role of Inductive Bias

Contents

Chapter 5 provides a unified vision of representations and weights for Unsupervised Domain Adaptation. We have introduced two terms, *invariance* (INV) and transferability (TSF), that both involve weights and representations. In particular, we have shown that one can minimize the invariance term by choosing weights as the ratio of target and source distributions of representations.

However, the transferability term involves labels in the target domain, which are absent in a scenario of UDA. Indeed, following the notation of Chapter 5, the transferability term is expressed (under some assumptions) as;

$$\mathrm{TSF}(w, \varphi) := \sup_{\mathbf{f} \in \mathcal{F}_C} \mathbb{E}_S \left[w(Z) \mathbf{Y} \cdot \mathbf{f}(Z) \right] - \underbrace{\mathbb{E}_T \left[\mathbf{Y} \cdot \mathbf{f}(Z) \right]}_{\text{involves target labels}} \tag{6.1}$$

Therefore, computing this term is the remaining difficulty for improving transferability of domain invariant representations.

Chapter 6 focuses on finding an approximation of the transferability term. The natural idea is to approximate the transferability term by replacing target labels with predictions. Following the notation of Chapter 5, we introduce the approximation of the transferability term based on predicitions in the target domain;

$$\widehat{\mathrm{TSF}}(w, \varphi) := \sup_{\mathbf{f} \in \mathcal{F}_C} \mathbb{E}_S \left[w(Z) \mathbf{Y} \cdot \mathbf{f}(Z) \right] - \mathbb{E}_T \left[\hat{\mathbf{Y}} \cdot \mathbf{f}(Z) \right] \tag{6.2}$$

where $\hat{Y}$ are predictions in the target domain. Incorporating predictions when learning domain invariant representations has been an influential strategy resulting to significant improvements of adaptation (Long et al. 2018) that we reviewed in Section 3.4. Such a strategy hopes to create a virtuous loop improving predictions by injecting the predicted labels in the transferability term. In turn, the predictions are used to recompute a better approximation of the transferability error and iterate the process.

However, there is a lack of theoretical understanding about why such a strategy is viable. In this Chapter, we show that naively replacing target labels with the source labelling classifier builds an approximation of the transferability term that behaves similarly to the invariance term. Thus, such a strategy suffers from a lack of theoretical ground. Nevertheless, empirical evidence highlights such a phenomenon operates (Long et al. 2018), demonstrating the incompleteness of the theoretical analysis. Our contribution is to identify the inductive bias of the learner as the key ingredient for initiating a virtuous feedback loop, with theoretical evidence. Pragmatically, inductive bias means that we are able to identify a model that performs better than the best source model.

Chapter 6 is organized as follows. We first show that naively injecting predicting target labels to approximate the transferability term builds a term that behaves similarly than invariance. It demonstrates this naive strategy does not find a viable theoretical support. Second, we review some classical inductive bias in Machine Learning and Deep Learning, as well as the role of inductive bias for preventing from overfitting and the risk of distribution shift. Third, we define formally the inductive bias in Unsupervised Domain Adaptation and derive a new bound of the target risk when inductive bias is available. In particular, we show that incorporating predicted labels in the transferability, as described in Equation 6.2, is compatible with a virtuous feedback loop.

The theoretical evidence that inductive bias is responsible of virtuous loop, hence promotes more transferable domain invariant representations, shifts our understanding of Unsupervised Domain Adaptation. We can speculate on where the remaining effort should be made; building stronger inductive bias to incorporate in the mature paradigm of learning domain invariant representations. This approach will motivate contributions of Part III.

Chapter 6 is based on the publication (Bouvier et al. 2020b) in an international conference;

> Robust Domain Adaptation: Representations, Weights and Inductive Bias
> *Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami and Céline Hudelot*,
> European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Ghent (Belgium), Online, 2020.

and covers Section 4 of (Bouvier et al. 2020b).

6.1 The role of predicted labels

The *transferability* term presented in Chapter 5 reflects if aligned representations correspond to similar classes. Thus, this new term is of interest to improve the transferability of domain invariant representations while remaining intractable since it involves target labels. A natural idea consists to replace target labels with the model prediction. Here, transferability improves predictions, and predictions improve the approximation of the transferability term, and so on, a phenomenon that we refer to as a *virtuous feedback loop*. In this section, we show this strategy is not supported theoretically. Second, we recall that previous prior works already incorporate model prediction for adaptation with some empirical success. We show that these works also suffer from a lack of theoretical support. Thus, a phenomenon escapes our theoretical description of UDA, which we will explore in the next section.

6.1.1 Approximated transferability error

We conduct formally the analysis when weights are absent since it is straightforward to extrapolate when they are present. The transferability term defined in Chapter 5 is defined as;

$$\mathrm{TSF}(\varphi) := \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S\left[\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)\right] - \mathbb{E}_T\left[\mathbf{g}_T(Z)\cdot\mathbf{f}(Z)\right] \tag{6.3}$$

where φ is representation in Φ, $\mathcal{F}_C$ is the set of measurable functions from the representation space $\mathcal{Z}$ to $[-1,1]^C$ and f_D is the labelling classifier in domain $D \in \{S,T\}$. We refer the reader to the Chapter 5 for more details about the transferability term and additional notations. In particular, we have related the error in the target domain to the transferability term and the invariance term $\mathrm{INV}(\varphi) = \sup_{f\in\mathcal{F}} \mathbb{E}_T[f(Z)] - \mathbb{E}_S[f(Z)]$. As aforementioned, the transferability term involves target labels, thus remains intractable in a scenario of UDA. We suggest to provide an approximation of the transferability term, named *approximated transferability*, where we simply replace the target labels with model prediction;

$$\widehat{\mathrm{TSF}}(\varphi, g_S) := \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S\left[\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)\right] - \mathbb{E}_T\left[\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)\right] \tag{6.4}$$

Although such strategy seems natural, we show that the approximated transferability behaves similarly than the invariance term;

Proposition 6.1.1. *Replacing the target labels in the transferability term with model prediction is equivalent to optimize the invariance error, i.e. ;*

$$\widehat{\mathrm{TSF}}(\varphi, f_S) \leq C\cdot\mathrm{INV}(\varphi) \tag{6.5}$$

Proof. We note that $\mathbf{g}_S\cdot\mathbf{f} = Cf$ for some $f\in\mathcal{F}$ using A.3.3 (See Chapter 5), thus $\widehat{\mathrm{TSF}}(\varphi,\mathbf{g}_S) := \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S\left[\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)\right] - \mathbb{E}_T\left[\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)\right] \leq C\sup_{f\in\mathcal{F}} \mathbb{E}_T[f(Z)] - \mathbb{E}_S[f(Z)]$ leading to the stated result. □

Let us discuss the implication of such result.

1. It proves that minimizing the invariance term minimizes the approximated transferability error naturally. We recall that invariance of representations is

incompatible with the scenario of label shift, as described in Section 3.4. Therefore, using the approximated term as a proxy of the transferability term will not improve, at least theoretically, the transferability of domain invariant representations.

2. It highlights that transferability term reflects the discrepancy between $\mathbf{g}_S$ and $\mathbf{g}_T$. When replacing $\mathbf{g}_T$ by $\mathbf{g}_S$ in the target expectation, the transferability term loses such a property.

6.1.2 Connections with Conditional Domain Adaptation Network

In this section, we conduct a discussion about Conditional Domain Adaptation Network (CDAN) (Long et al. 2018) which aims to improve *Domain Adversarial Neural Networks* (DANN) (Ganin and Lempitsky 2015) by leveraging model's output. From this point of view, it thus represents a strategy very similar to the approximation of the transferability error replacing the target labels by the model's predictions. We prove that CDAN suffers from the same lack of theoretical guarantee than the one described in Proposition 6.1.1.

In this section, we consider $g \in \mathcal{G}_{\mathrm{p}}$. CDAN aims to align across domains the distribution of the couple of variable $(\hat{Y}, Z)$ where $\hat{Y} = g_S\varphi(X)$ represents the model prediction expressed as the probability of a class. Notably, it is performed by exposing the tensor product between $\hat{Y}$ and Z to a discriminator in order to align conditional $Z|\hat{Y}$. Importantly, CDAN leads to a substantial improvement of performances compared to DANN (Ganin and Lempitsky 2015). We refer the reader to Section 3.4 for more details about CDAN.

Proposition 6.1.2. *For a given representation $\varphi \in \Phi$;*

$$L_{\mathrm{DANN}}(\varphi) = L_{\mathrm{CDAN}}(\varphi) \tag{6.6}$$

where;

$$L_{\mathrm{DANN}}(\varphi) = \inf_{d \in \mathcal{D}} -\mathbb{E}_S[\log(d(Z))] - \mathbb{E}_T[\log(1 - d(Z))] \tag{6.7}$$

$$L_{\mathrm{CDAN}}(\varphi) = \inf_{d \in \mathcal{D}_\otimes} -\mathbb{E}_S[\log(d(Z \otimes Y))] - \mathbb{E}_T[\log(1 - d(Z \otimes Y))] \tag{6.8}$$

where $\mathcal{D}$ and $\mathcal{D}_\otimes$ are infinite capacity set of discriminators of $\mathcal{Z}$ and $\mathcal{Z} \otimes \mathcal{Y}$ respectively.

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 $L_{\mathrm{CDAN}}(\varphi) \leq L_{\mathrm{DANN}}(\varphi)$. 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 $L_{\mathrm{CDAN}}(\varphi) \geq L_{\mathrm{DANN}}(\varphi)$ finishing the proof. □

This proposition follows a key assumption that we are in a context of infinite capacity discriminators of both $\mathcal{Z}$ and $\mathcal{Y} \otimes \mathcal{Z}$, which is a similar setting that considering the set of measurable functions when developing the theory of invariance and transferability from Chapter 5. The theoretical equivalence between the DANN loss (L_{DANN}) and CDAN loss (L_{CDAN}) is in accordance to the Proposition 6.1.1. However, as described in Section 3.4, we observe a significant empirical improvement

of CDAN and DANN. Thus, there is a missing brick in the theory that we aim to overcome in the following section through the lens of *Inductive Bias*.

6.2 Inductive Bias

When replacing the target labels in the transferability term with f_S, the approximated transferability behaves similarly than invariance (See Proposition 6.1.1). It results from the fact that labelling functions in each term of the expectation, source and target expectations respectively, are the same. Therefore, to go beyond the invariance regime, our intuition is to replace the classifier g_T by some classifier $\tilde{g}$ that we will specify later. More formally, we now consider the following approximated error based on $\tilde{g}$;

$$\widehat{\text{TSF}}(\varphi, \tilde{g}) := \sup_{\mathbf{f} \in \mathcal{F}_C} \mathbb{E}_S \left[\mathbf{g}_S(Z) \cdot \mathbf{f}(Z) \right] - \mathbb{E}_T \left[\tilde{\mathbf{g}}(Z) \cdot \mathbf{f}(Z) \right] \tag{6.9}$$

Seemingly, the best function $\tilde{\mathbf{g}}$ for approximating the transferability term $\text{TSF}(\varphi)$ is $\mathbf{g}_T$. We recall that g_T is out of reach in Unsupervised Domain Adaptation. Nevertheless, what happens if $\tilde{g}$ tends to get closer to g_T than g_S is to g_T? More formally, what is the guarantee behind the approximated transferability error based on $\tilde{g}$ if we are able to build a function $\tilde{g}$ such that;

$$\text{Err}_T(\tilde{g}\varphi) < \text{Err}_T(g_S\varphi) \tag{6.10}$$

Intuitively, we assume that we are able to build a classifier $\tilde{g}$ that performs better in the target domain than the source labelling function. We refer to this phenomenon as our *Inductive Bias* of the adaptation problem. In the following, we provide an overview of the ubiquitous role of Inductive Bias in Machine Learning, from preventing overfitting (Section 6.2.2) and to address out-of-distribution generalization (Section 6.2.3). The Section 6.3 shows that the assumption of inductive bias formulated in Equation 6.10 is sufficient to guarantee that a positive feedback loop occurs to improve transferability of domain invariant representations.

6.2.1 Historical overview

Inductive Bias is the set of assumptions enforced into a learner to empower generalization. The *Occam's razor* is an iconic example of inductive bias; it states that among models that explain equally well the training data, we should promote the simplest one. In particular, it is even sometimes better to not have a null error on training data if we could have a small one with a much simpler model as presented in Figures 6.1(a) and 2.7. The Section 3.1.3 provides a formal depiction of such principle through the lens of Structural Risk Minimization (SRM) that controls the complexity of the hypothesis class through its VC dimension.

The search for solid inductive bias has been a crucial question around the development of Machine Learning. In the following, we provide some examples of influential inductive bias. Hence, this depiction does not give an exhaustive overview of inductive bias in Machine Learning but has only an illustration purpose.

- **Conditional independence:** Naives Bayes model corresponds to assume features independence making it one of the simplest models (Zhou et al. 2004). As a result, one can learn the correlation between a feature's coordinate with the

label independently from other features, resulting in a computationally efficient algorithm. Despite the fierce independence assumption, the Naive Bayes model remains a strong baseline, even when features are highly correlated.

- **Maximal Margin:** Maximal margin corresponds to assume that the model which exhibits the best generalization is the one that separates the data with the maximal margin. When data from two different classes are linearly separable, it may exist an infinite number of hyperplans that split the data with a null error. Which one should be promoted? Among all the hyperplans that equally split the training data, the *Maximal Margin* principle isolates the separator that maximises the margin between two classes, as presented in Figure 6.1(b). This principle has led to a computationally efficient algorithm, named *Support Vector Machine* (SVM), with generalisation virtues observed in practice and a natural extension to kernels (Cortes and Vapnik 1995).

- **Minimal Features:** Minimal features correspond to assume that the model which exhibits the best generalization is the one built upon the minimal number of features. The model focuses on a subset of features that contributes the most to the predictions by rejecting features responsible for a marginal performance improvement on the training data. The principle of Minimal Features can be achieved by penalization of model's parameter where a typical example is the Ridge (L^2) and Lasso (L^1) penalization, or by method for features selection.

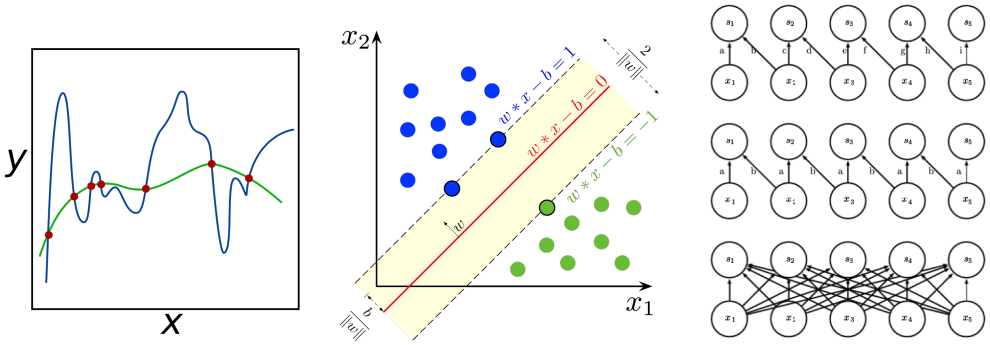
- **Semi-Supervised Learning:** To learn a well-performing classifier from small labelled data and large unlabelled data, *Semi-Supervised Learning* (SSL) leverages various inductive bias as presented in Section 3.1.4. Such inductive biases include the cluster assumption, the smoothness assumption, the manifold assumption or the minimal entropy principle (Grandvalet and Bengio 2004). Crucially, inductive bias in SSL focuses on assumption enforced on unlabelled data.

- **Deep Learning:** Convolutional Neural Networks (ConvNets) (LeCun, Bengio, and Hinton 2015) is the first successful application of an inductive bias enforced in neural nets inspired by the brain functioning, here the visual cortex as presented in Figure 6.1(c). Recurrent neural networks for sequential data, *e.g.* texts or audio data, draw inspiration from the memory to develop a mechanism to propagate information through long time frames (Hochreiter and Schmidhuber 1997). More recently, *Natural Language Processing* has made significant progress by moving from recurrent networks to networks that apply the attention mechanism (Vaswani et al. 2017). From the mathematical point of view, developing Deep Learning is closely related to the finding of new inductive bias for regularizing an over-parametrized model.

6.2.2 Preventing from overfitting

One can mathematically formulate the principle of inductive bias as restricting the hypothesis space $\mathcal{H}$ to $\tilde{\mathcal{H}} \subset \mathcal{H}$ to empower generalization. Pragmatically, we *bias* the hypothesis class to more plausible models. For instance, for neural networks applied to image processing, $\tilde{\mathcal{H}}$ may be Convolutional Neural Networks. Formally, given a distribution on $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{D}$[1] that contains n IID realizations from p, minimizing the

[1]Note that the notation $\mathcal{D}$ deviates in this chapter from the set of discriminators. See the list of symbols.



(a) Occam's razor (b) Maximal separation (c) Convolutional layer

Figure 6.1: Illustration of some popular inductive bias in Machine Learning. (a) The Occam's razor promotes the simpler model among those that explain equally well the data. *Image from* `en. wikipedia. org/ wiki/ Regularization_ ( mathematics)` . (b) Support Vector Machines (SVM) implement a classifier that maximizes the margin between classes. *Image from* `en. wikipedia. org/ wiki/ Support-vector_ machine` . (c) A convolutional layer is inspired by the visual cortex and implements parameter sharing and local interaction. *Image from the Deep Learning book* `www. deeplearningbook. org/` *(Goodfellow, Bengio, and Courville 2016).*

empirical error $\widehat{\text{Err}}_{\mathcal{D}}(h)$ for $h \in \tilde{\mathcal{H}}$ leads to a overall better solution on p compared to minimizing the empirical error $\widehat{\text{Err}}_{\mathcal{D}}(h)$ for $h \in \mathcal{H}$;

$$\text{Err}_p(\tilde{h}^\star) \leq \text{Err}_p(h^\star) \quad \text{where} \quad \begin{cases} \tilde{h}^\star = \arg\min_{h \in \tilde{\mathcal{H}}} \widehat{\text{Err}}_{\mathcal{D}}(h) \\ h^\star = \arg\min_{h \in \mathcal{H}} \widehat{\text{Err}}_{\mathcal{D}}(h) \end{cases} \tag{6.11}$$

We refer to the section 3.1.3 where we detail the formal principle of *Structural Risk Minimization* for a wider development of this principle.

6.2.3 Out-of-distribution generalization

In the previous section, we have detailed the role of inductive bias to prevent from overfitting. Formally, inductive bias bridges the gap between the empirical distribution $\hat{p}_{\mathcal{D}} = \sum_{(x,y) \in \mathcal{D}} \delta_{(x,y)}$ and the underlying (targeted) distribution p, where $\mathcal{D}$ is a set of IID samples from p. Similarly, inductive bias is a crucial ingredient to generalize from a source to a different target distribution, respectively noted p_S and p_T. Following the formal statement of inductive bias in a context of overfitting (Equation 6.11), we would like to observe ideally a similar statement for out-of-distribution generalization;

$$\text{Err}_T(\tilde{h}^\star) \leq \text{Err}_T(h^\star) \quad \text{where} \quad \begin{cases} \tilde{h}^\star = \arg\min_{h \in \tilde{\mathcal{H}}} \text{Err}_S(h) \\ h^\star = \arg\min_{h \in \mathcal{H}} \text{Err}_S(h) \end{cases} \tag{6.12}$$

Note that we leave out the difficulty of generalization from a finite set of samples; we focus on generalizing from a source to a target distribution when provided with infinite data from both distributions p_S and p_T. Inductive bias is crucial in the theory of learning from different domains (Section 3.2);

$$\text{Err}_T(h) \leq \text{Err}_S(h) + \delta_{\mathcal{H}} + \lambda_{\mathcal{H}} \tag{6.13}$$

Biasing the hypothesis space to more plausible hypotheses $\tilde{\mathcal{H}}$ falls into the fundamental trade-off of maintaining transferability of domain invariant representations that we have deeply investigated in Section 3.4;

Proposition 6.2.1 (Inductive Bias). *Let $\tilde{\mathcal{H}} \subset \mathcal{H}$, then;*

$$\delta_{\tilde{\mathcal{H}}} \leq \delta_{\mathcal{H}} \quad \textit{while} \quad \lambda_{\tilde{\mathcal{H}}} \geq \lambda_{\mathcal{H}} \tag{6.14}$$

However, we still have poor knowledge about the role of inductive bias when learning domain invariant representations.

6.3 Theoretical Aspects

6.3.1 Inductive Bias

Motivations

Our proposition follows the intuitive idea that;

$$g_S := \arg\min_{g \in \mathcal{G}} \mathrm{Err}_S(g\varphi), \tag{6.15}$$

i.e. the best source classifier, is not necessarily the best target classifier, *i.e.* ;

$$g_S \neq \arg\min_{g \in \mathcal{G}} \mathrm{Err}_T(g\varphi). \tag{6.16}$$

For instance, a well suited regularization, noted $\Omega_T(g)$ that involves unlabelled target data may improve target performance[2]. In Chapter 8, we will discuss the role of the cluster assumption during adaptation. Thus, noting;

$$\tilde{g} := \arg\min_{g \in \mathcal{G}} \mathrm{Err}_S(g\varphi) + \lambda \cdot \Omega_T(g), \tag{6.17}$$

we hope to obtain;

$$\mathrm{Err}_T(\tilde{g}\varphi) < \mathrm{Err}_T(g_S\varphi), \tag{6.18}$$

if Ω_T is well-chosen, that we refer to as *Inductive Bias* (IB). Formally, given a representation $\varphi \in \Phi$, we assume it exists a process IB, called Inductive Bias of representations, that builds an hypothesis $\tilde{h}^\varphi = \mathsf{IB}(\varphi)$ that reduces the error in the target domain with guarantee;

Definition 6.3.1 ($\beta-$Inductive Bias). *Given a representation $\varphi \in \Phi$, a classifier $g \in \mathcal{G}$ where we note $\mathcal{H} = \mathcal{G}\Phi$, an Inductive Bias* IB, *at level $0 < \beta \leq 1$, builds $\tilde{h}^\varphi := \mathsf{IB}(\varphi) \in \mathcal{H} = \mathcal{G}\Phi$ such that;*

$$\sup_{\varphi \in \Phi} \left\{ \frac{\mathrm{Err}_T\left(\tilde{h}^\varphi\right)}{\mathrm{Err}_T(g_S\varphi)} \right\} = \beta \tag{6.19}$$

where $g_S = \arg\min_{g \in \mathcal{G}} \mathrm{Err}_S(g\varphi)$. We say the inductive design is $\beta-$strong when $\beta < 1$ and weak when $\beta = 1$.

The definition of IB introduces β that embodies the strength of the inductive bias. Note that β is a condition that involves the whole representation class Φ, thus does not depend on a particular φ, which is a strong assumption. The closer to 1 is β, the less improvement we can expect using the inductive classifier $\tilde{h}^\varphi$.

[2] See our dicussion on Semi-Supervised Learning in Section 3.1.4

For illustration purpose, we provide two simple instantiations of IB. The former conserves the representation while the latter modifies it. We will refer to the former as an Inductive Bias of the classifier.

Example 6.3.1 (Inductive Bias of the classifier)**.** *Let a regularization Ω, the inductive design of the classifier consists in building $\mathsf{IB}(\varphi)$ by conserving the representation. Formally, i.e. $\tilde{h}^{\varphi} = \tilde{g}\varphi$ where*

$$\tilde{g} = \arg\min_{g \in \mathcal{G}} \mathrm{Err}_S(g\varphi) + \lambda \cdot \Omega(g\varphi) \tag{6.20}$$

for some $\lambda > 0$. For instance, $\Omega(g\varphi)$ could be the entropy loss, i.e. it promotes classifier resulting in small predictions entropy (Grandvalet and Bengio 2005).

As an illustration of an inductive bias that modifies the representation φ, we use a simple gradient descent update according to a regularization Ω, *e.g.* predictions entropy (Grandvalet and Bengio 2005).

Example 6.3.2. *Let a regularization Ω, one can build $\mathsf{IB}(\varphi)$ by modifying the representation, i.e. $\tilde{h}^{\varphi} = g_S\tilde{\varphi}$ where*

$$\tilde{\varphi} = \varphi - \lambda \cdot \nabla_{\varphi}\Omega(g_S, \varphi) \tag{6.21}$$

for some $\lambda > 0$.

To conclude, the assumption of inductive bias allows identifying a model that performs strictly better than the best source classifier from representations.

6.3.2 The role of Inductive Bias in Adaptation

Main result

We now study the impact of the inductive bias of a classifier in Proposition 5.3.1. Thus, we introduce the *inductive transferability error*:

$$\widehat{\mathrm{TSF}}(\varphi, \tilde{h}^{\varphi}) := \sup_{\mathbf{f} \in \mathcal{F}_C} \left\{ \mathbb{E}_T \left[\tilde{\mathbf{h}}^{\varphi}(X) \cdot \mathbf{f}(Z) \right] - \mathbb{E}_S[\mathbf{g}_S(Z) \cdot \mathbf{f}(Z)] \right\} \tag{6.22}$$

where $\tilde{h}^{\varphi} = \mathsf{IB}(\varphi)$. We show the presence of an inductive design IB leads to a bound of the target risk where transferability error is free of target labels:

Proposition 6.3.1 (Guarantee of Adaptation in presence of Inductive Bias)**.** *Let $\varphi \in \Phi$ and a $\beta-$strong Inductive Bias IB;*

$$\mathrm{Err}_T(\tilde{h}^{\varphi}) \leq \rho \left(\mathrm{Err}_S(g_S\varphi) + 3C \cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^{\varphi}) + \mathrm{Err}_T(g_T\varphi) \right) \tag{6.23}$$

where $\tilde{h}^{\varphi} = \mathsf{IB}(\varphi)$ and $\rho := \frac{\beta}{1-\beta}$.

Proof. First, we reuse Proposition 5.2.2 with a new triangular inequality involving the inductive design $\tilde{h}^{\varphi} := \mathsf{IB}(\varphi)$:

$$\mathrm{Err}_T(g\varphi) \leq \mathrm{Err}_S(g\varphi) + \delta_{\mathcal{F}_C}(\varphi) + \mathrm{Err}_T(g_S\varphi, \tilde{h}^{\varphi}) + \mathrm{Err}_T(\tilde{h}^{\varphi}, g_T\varphi) + \mathrm{Err}_T(g_T\varphi)$$

where $\mathrm{Err}_T(\tilde{h}^{\varphi}, g_T\varphi) \leq \mathrm{Err}_T(\tilde{h}^{\varphi})$. Now, following previous proofs of Chapter 5 (proofs of Propositions 5.2.3 and 5.2.4), we can show that: $\mathrm{Err}_T(g_S\varphi, \tilde{h}^{\varphi}) \leq C \cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^{\varphi})$. Then, $\mathrm{Err}_T(g\varphi) \leq \mathrm{Err}_S(g\varphi) + 3C \cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^{\varphi}) + \mathrm{Err}_T(\tilde{h}^{\varphi}) + \mathrm{Err}_T(g_T\varphi)$. This bound is true for any g and in particular for $g_S =$

$\arg\min_{g\in\mathcal{G}} \mathrm{Err}_S(g\varphi)$, then;

$$\mathrm{Err}_T(g_S\varphi) \leq \mathrm{Err}_S(g_S\varphi) + 3C\cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^\varphi) + \mathrm{Err}_T(\tilde{h}^\varphi) + \mathrm{Err}_T(g_T\varphi) \quad (6.24)$$

then the assumption of $\beta-$strong inductive design is $\mathrm{Err}_T(\tilde{h}^\varphi) \leq \beta\mathrm{Err}_T(g_S\varphi)$ which leads to;

$$\mathrm{Err}_T(g_S\varphi) \leq \mathrm{Err}_S(g_S\varphi) + 3C\cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^\varphi) + \beta\mathrm{Err}_T(g_S\varphi) + \mathrm{Err}_T(g_T\varphi) \quad (6.25)$$

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

$$\mathrm{Err}_T(g_S\varphi) \leq \frac{1}{1-\beta}\left(\mathrm{Err}_S(g_S\varphi) + 3C\cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^\varphi) + \mathrm{Err}_T(g_T\varphi)\right) \quad (6.26)$$

And finally, using $\mathrm{Err}_T(\tilde{h}^\varphi) \leq \beta\mathrm{Err}_T(g_S\varphi)$;

$$\mathrm{Err}_T(\tilde{h}^\varphi) \leq \frac{\beta}{1-\beta}\left(\mathrm{Err}_S(g_S\varphi) + 3C\cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^\varphi) + \mathrm{Err}_T(g_T\varphi)\right) \quad (6.27)$$

finishing the proof. □

We can provide a similar version of this theorem when using weights in the source domain as described in Chapter 5;

Proposition 6.3.2 (Guarantee of Adaptation in presence of Inductive Design (Source weighted version)). *Let $\varphi \in \Phi$ and $w : \mathcal{Z} \to \mathbb{R}^+$ such that $\mathbb{E}_S[w(z)] = 1$ and a $\beta-$strong Inductive Design* IB*;*

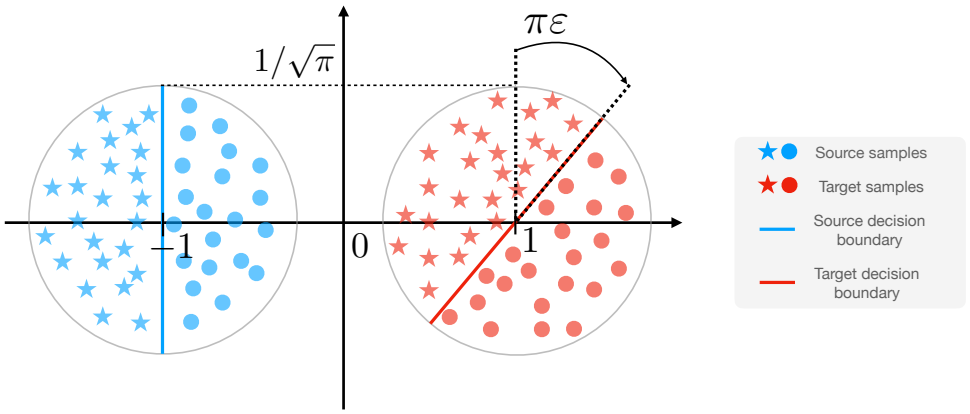
$$\mathrm{Err}_T(\tilde{h}^\varphi) \leq \rho\left(\mathrm{Err}_{w\cdot S}(g_{w\cdot S}\varphi) + 3C\cdot \mathrm{INV}(w,\varphi) + \widehat{\mathrm{TSF}}(w,\varphi,\tilde{h}^\varphi) + \mathrm{Err}_T(g_T\varphi)\right) \quad (6.28)$$

where $\tilde{h}^\varphi = \mathsf{IB}(\varphi)$, $\rho := \frac{\beta}{1-\beta}$ and

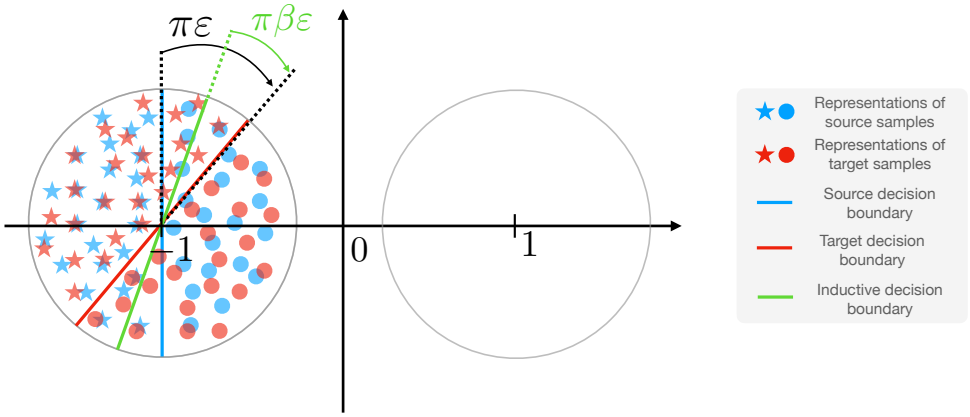
$$\widehat{\mathrm{TSF}}(w,\varphi,\mathsf{IB}) := \sup_{\mathbf{f}\in\mathcal{F}_C}\left\{\mathbb{E}_T\left[\tilde{\mathbf{h}}^\varphi(X)\cdot\mathbf{f}(Z)\right] - \mathbb{E}_S[w(Z)\mathbf{g}_S(Z)\cdot\mathbf{f}(Z)]\right\} \quad (6.29)$$

When provided with inductive bias and relying on the bound from Proposition 9.3.1, the target labels are only involved in $\mathrm{Err}_T(g_T\varphi)$. This term reflects the level of noise when fitting labels from representations, thus one can reasonably assume it is a small error. Therefore, we move beyond the difficulty of achieving a small combined error (a small adaptability error), a condition required for the success of adaptation in the influential theory (Ben-David et al. 2010a), to simply achieving a small error in the target domain with an ideal classifier, which is arguably a much weaker condition. In particular, the transferability term 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.

Seemingly, breaking the dependence of the bound to target labels is paid at cost of a $\rho = \beta/(1-\beta)$ that explodes ($\rho \to +\infty$) when the inductive bias becomes weaker ($\beta \to 1^-$). Besides, β is unknown (only assumed to be smaller than 1) and is a factor of all other terms of the bound, making the bound intractable. More precisely, the bound of the target error is a sum of tractable terms (assuming that $\mathrm{Err}_T(g_T\varphi)$ is negligible) scaled by an unknown factor ρ. When assuming $\mathrm{Err}_T(g_T\varphi) = 0$, two things should be noted;



(a) Input data



(b) Representations of data

Figure 6.2: Illustration of example 6.3.3. (a): Source and target data do not overlap. (b): A representation φ makes invariant distributions. The source classifier achieves an error ε on the target data while the inductive classifier achieves an error $\beta\varepsilon$ on the target data. The area where the source and the inductive classifiers are different is $(1-\beta)\varepsilon$, resulting to an inductive transferability error of $(1-\beta)\varepsilon$.

- The bound from Proposition 9.3.1 is not helpful if one wants to have a tractable upper bound of the target risk. Indeed, $\mathrm{Err}_S(g_S\varphi) + 3C \cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^\varphi)$ is tractable but we do not know how to relate it to $\mathrm{Err}_T(\tilde{\varphi})$ since ρ is unknown.
- The bound from Proposition 9.3.1 is helpful if one wants to minimize the target error by gradient descent. Indeed, if the bound is not vacuous[3], that is $\rho\left(\mathrm{Err}_S(g_S\varphi) + 3C \cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^\varphi) + \mathrm{Err}_T(g_T\varphi)\right)$ is a tight upper bound of $\mathrm{Err}_T(\tilde{h}^\varphi)$, one can use the gradient of $\mathrm{Err}_S(g_S\varphi) + 3C \cdot \mathrm{INV}(\varphi) + \widehat{\mathrm{TSF}}(\varphi, \tilde{h}^\varphi)$ as a proxy of the direction of the gradient of $\mathrm{Err}_T(\tilde{h}^\varphi)$.

To illustrate the effect of inductive bias, we provide a toy example;

Example 6.3.3 (A toy example of Inductive Design). *We build an artificial dataset where the principle of invariance achieves an error of $\varepsilon \in (0,1)$. To this purpose, we consider a two dimensional data (x_1, x_2) uniformly sampled on a disk with radius $\frac{1}{\sqrt{\pi}}$ centered in -1 and 1 in the source and target domains respectively, as presented in Figure 6.2(a). We assume that the data is linearly separable in both domains by the lines $x_1 = -1$ and*

[3] We recall it is impossible to test this assumption in practice without labels in the target domain.

$x_2 = \text{cotan}(\pi\varepsilon)(x_1 - 1)$ *in the source and target domains, respectively. We consider the representation;*

$$\varphi(x_1, x_2) := \begin{cases} (x_1, x_2) & \text{if } (x_1+1)^2 + (x_2+1)^2 \leq \frac{1}{\pi} \\ (x_1 - 2, x_2 - 2) & \text{if } (x_1-1)^2 + (x_2-1)^2 \leq \frac{1}{\pi} \end{cases} \tag{6.30}$$

as presented in Figure 6.2(b). Then, one can observe that;

$$\begin{cases} \text{Err}_T(g_S\varphi) & = \varepsilon \\ \text{Err}_S(g_S\varphi) & = 0 \\ \text{INV}(\varphi) & = 0 \\ \text{TSF}(\varphi) & = \varepsilon \\ \text{Err}_T(g_T\varphi) & = 0 \end{cases} \tag{6.31}$$

Indeed, since $x_2 = -1$ *separates the data in the source domain then* $\text{Err}_S(g_S\varphi) = 0$*, while* $\text{Err}_T(g_S\varphi) = \varepsilon$ *results from an area calculation of two disk portions of angle* $\pi\varepsilon$*.* $\text{INV}(\varphi) = 0$ *is null since source and target data representations have the same distribution.* $\text{TSF}(\varphi) = \varepsilon$ *is achieved with the critic function;*

$$\mathbf{f} : (x_1, x_2) \mapsto \begin{pmatrix} 2\mathbb{I}(x_1 > -1) - 1 \\ 2\mathbb{I}(x_1 < -1) - 1 \end{pmatrix} \tag{6.32}$$

Now we assume it exists an Inductive Bias $\tilde{h}^\varphi = \mathsf{IB}(\varphi)$ *that is* $\beta-$*strong, i.e.* $\text{Err}_T(\tilde{h}^\varphi) \leq \beta \cdot \text{Err}_T(g_S\varphi)$*. Similarly, we observe that* $\widehat{\text{TSF}}(\varphi, \tilde{h}^\varphi) = (1-\beta)\varepsilon$*, leading to;*

$$\text{Err}_T(\tilde{h}^\varphi) \leq \frac{\beta}{1-\beta} \cdot \widehat{\text{TSF}}(\varphi, \tilde{h}^\varphi) = \beta\varepsilon \tag{6.33}$$

Two important points should be noted:

1. *The bound is not vacuous, even tight, since* $\text{Err}_T(\tilde{h}^\varphi) = \frac{\beta}{1-\beta} \cdot \widehat{\text{TSF}}(\varphi, \tilde{h}^\varphi)$*.*
2. *The inequality* $\text{Err}_T(\tilde{h}^\varphi) \leq \beta \cdot \text{Err}_T(g_S\varphi)$ *involves two terms that are unknown (*β *and* $\text{Err}_T(g_S\varphi)$*). In the inequality* $\text{Err}_T(\tilde{h}^\varphi) \leq \frac{\beta}{1-\beta} \cdot \widehat{\text{TSF}}(\varphi, \tilde{h}^\varphi)$ *bounds* $\text{Err}_T(\tilde{h}^\varphi)$ *with a tractable term* $\widehat{\text{TSF}}(\varphi, \tilde{h}^\varphi)$ *while* $\frac{\beta}{1-\beta}$ *remains unknown.*

Our analysis reveals an essential condition when aiming to improve the transferability of invariant representation; it is sufficient to have an inductive bias. From the practical perspective, this means we can replace the target labels in the transferability term with the model's prediction, as long as we enforce additional assumptions during learning. For instance, Chapter 8 will implement the cluster assumption when learning domain invariant representations, resulting in a drastic improvement of performances in the target domain.

Part III

From Theory to Practice: Some Implementations of Adaptation

7 The challenge of Label Shift

Contents

Part II explores a theoretical view of the problem of adaptation with an emphasis on learning more transferable domain invariant representations. In particular, we study in Chapter 4 the role of the *Hypothesis Class Reduction* to achieve a minimal adaptability error. In Chapter 5, we unify weights and representations, introducing two terms, *invariance* and *transferability*. Crucially, we prove that invariance is achieved by carefully choosing weights, following the influential line of study based on the *covariate shift* assumption (Quinonero-Candela et al. 2009) (see Section 3.2). However, the transferability term remains intractable since it involves the target labels. Thus, we have theoretically isolated the role of inductive bias to provide an estimation of transferability error in Chapter 6.

Part III is dedicated to implementing various inductive biases for improving the transferability of domain invariant representations. The present Chapter focuses on the problem of *Label Shift*, *i.e.* where the distribution of labels shifts across domains $p_T(Y) \neq p_S(Y)$. In Section 3.4, more specifically the theoretical analysis from (Zhao et al. 2019) presented in Equation 3.69, we have brought elements showing that the invariance principle is bound to fail in this case. We show how the theoretical development from Chapter 5 can address the problem of label shift.

More specifically, we will use both weights and representations to address this challenging problem of label shift, as motivated in Chapter 5. There are two ingredients. First, we design weights to achieve invariance in the representation space between a weighted source domain and the target domain. As a result, one must minimize the transferability error to improve performance in the target domain. To this purpose, we rely as a second step on a *weak* inductive bias, *i.e.* the transferability error is approximated by replacing target labels with by the prediction provided by the model. We recall that despite empirical evidence that such strategy is viable (Long et al. 2018), there is no theoretical ground for this approach as described in Section 6.1. Our approach results into a bi-level optimization that involves both weights and

representations. Through various experiments, we demonstrate that the theory developed in Chapters 5 and 6 leads to a significant improvement of transferability of domain invariant representations in the challenging scenario of label shift.

Chapter 7 is organized as follows. First, we provide a theoretical analysis on the interaction between the design of weights, that we will refer to as the *inductive bias* of weights, and property of invariance of representations. Additionally, we will define the notion of *weak* inductive bias of weights. Second, we expose the optimization problem that combines both weights and invariant representations through a domain adversarial objective. Finally, we conduct experiments on two standard benchmarks of Unsupervised Domain Adaptation demonstrating that our two inductive biases, on the classifier and weights respectively, improve adaptation in a context of label shift.

Chapter 7 is based on the publication (Bouvier et al. 2020b) in an international conference;

> Robust Domain Adaptation: Representations, Weights and Inductive Bias
> *Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami and Céline Hudelot*,
> European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Ghent (Belgium), Online, 2020.

and covers Sections 4, 5 and 6 of (Bouvier et al. 2020b).

7.1 Preliminaries

7.1.1 Introduction

The situation of *label shift* is characterized by the difference between the source and target distributions of labels;

$$p_T(Y) \neq p_S(Y) \tag{7.1}$$

In Section 3.4, we have reviewed the theoretical analysis from (Zhao et al. 2019) proving that invariant representations are bound to fail in situation of label shift. Such failure results from the fact that invariance of representations is too strong a constraint. Pragmatically, enforcing invariance when label shift is observed aligns source and target samples that do not belong to the same, hence degrades model's performance in the target domain. Previous works (Cao, Long, and Wang 2018; Wu et al. 2019; You et al. 2019; Cao et al. 2018) relax invariance through a source weighted distribution, *i.e.* transforming the source distribution in the representation $p_S(z)$ to $w(z)p_S(z)$ where $\mathbb{E}_S[w(z)] = 1$. More details about this strategy are provided in Section 3.4 through an example of *Partial Domain Adaptation* (Cao et al. 2018). Formally, learning invariant representations between a source weighted domain and a target domain enforces the equality;

$$w(z)p_S(z) = p_T(z) \tag{7.2}$$

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?

7.1.2 Theoretical analysis

In Chapter 3 (Section 3.4), we have presented the problem of *Partial Domain Adaptation* (PDA) (Cao et al. 2018), as a particular instance of the problem of label shift in Unsupervised Domain Adaptation. In particular, the work (Cao et al. 2018) designs weights based on the prediction, *i.e.* w is not directly a function from the representation z but from prediction $\hat{Y} = g_S\varphi(X)$. Similarly, weights have also been studied in (Long et al. 2018) by developing *entropy conditioning*, that is designing weights based on the entropy of predictions. More precisely, weights are designed as entropy conditioning; $w(z) \propto 1 + e^{-v}$ where $v = -\sum_{1\leq c\leq C} g_{S,c}\log(g_{S,c})$ is predictions entropy. In the following, we analyze how the design of weights, that we will refer to the *inductive bias of weights*, enforces new property of invariance of representations. To this purpose, we formalize this bias as a dependence of weights with respect to some transformation of representations $z' = \psi(z)$, for some function ψ, *i.e.* w depends on z', that we note $w(z')$ by notation abuse. For instance, in the particular case of (Cao et al. 2018), weights are built from predictions, *i.e.* $\psi(z) = g_S(z)$, while entropy conditioning (Long et al. 2018) builds weights from entropy, *i.e.* $\psi(z) = -\sum_{1\leq c\leq C} g_{S,c}(z)\log(g_{S,c}(z))$.

Proposition 7.1.1 (Inductive bias of weights 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:*

$$w(z') = \frac{p_T(z')}{p_S(z')} \quad \textit{and} \quad p_S(z|z') = p_T(z|z') \tag{7.3}$$

Furthermore, if additionally $\mathrm{Err}_S(g_S\varphi) = \mathrm{Err}_T(g_T\varphi) = 0$, *then* $p_S(Y|Z) = p_T(Y|Z)$ *and* $P_S(Y|Z') = p_T(Y|Z')$.

Proof. See Appendix B. □

The literature that uses weights for improving transferability of domain invariant representations (Long et al. 2018; Cao et al. 2018; You et al. 2019; Wu et al. 2019) builds weights through heuristics, which we call *inductive bias of weights*. Proposition 7.1.1, therefore, clarifies the interaction of these heuristics with the resulting domain invariant representations. In particular, this choice imposes new invariance conditions on Z' and $Z|Z'$ where $Z' = \psi(Z)$. As an illustration, when building weights from prediction, our proposition shows that $Z|\hat{Y}$ is domain invariant which can be compared to the assumption of *Generalized Label Shift*, *i.e.* $Z|Y$ is invariant, that has been studied in a later work (Combes et al. 2020b).

7.1.3 A weak Inductive Bias of Weights

In a similar spirit than Chapter 6 where we have distinguished the strong and the weak inductive biases of representations, discussing their theoretical properties, we will conduct a similar discussion for the inductive bias of weights. We refer to a *weak* inductive bias of weights when we do not rely on an heuristic when building weights, *i.e.* ψ is the identity function. Crucially, under the assumption that the target distribution is included in the support of the source distribution, one can control naturally the invariance term by choosing a well-suited value of weights w, *i.e.* leading to $\mathrm{INV}(w, \varphi) = 0$. Indeed, $w^\star(\varphi) = \arg\min_w \mathrm{INV}(w, \varphi)$ has a closed form when given an optimal domain discriminator $d^\star$;

$$d^\star(z) := \frac{p_S(z)}{p_S(z) + p_T(z)}, \tag{7.4}$$

setting $w^\star(z) := (1 - d^\star(z))/d^\star(z) = p_T(z)/p_S(z)$ leads to $w(z)p_S(z) = p_T(z)$ and finally $\mathrm{INV}(w^\star(\varphi), \varphi) = 0$. Thus, given a representation φ, one can determine weights $w^\star$ that achieve invariance, as long as one has access to an optimal discriminator.

From the practical perspective, it is unlikely to obtain easily an optimal discriminator, especially at an early stage of learning. Thus, using exactly the closed form $w^\star(z)$ may degrade the estimation of the transferability term. We overcome this issue by building a *weight relaxation* technique by building $\tilde{w}_d$ which are smoothly pushed to $w^\star$ during training. This is done using temperature relaxation in the sigmoid output of the domain discriminator;

$$w_d^\tau(z) := \frac{1 - \sigma\left(\tilde{d}(z)/\tau\right)}{\sigma\left(\tilde{d}(z)/\tau\right)} \tag{7.5}$$

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

7.2 Algorithm

7.2.1 Label Shift Robust Adaptation

In this section, we expose a new learning procedure which relies on two weak inductive biases on both weights and the classifier. We recall some notations $\mathcal{G}_{\mathrm{p}}$ is a subset of measurable functions from $\mathcal{Z}$ to $\mathcal{P}(\mathcal{Y})$, *i.e.* $g\varphi(X)$ is a vector of probabilities where $g\varphi(X)_c$ is the (predicted) probability of X to belong to c for $c \in \{1, \cdots, C\}$. 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 Unsupervised Domain Adaptation):

$$\begin{cases} g^\star, \varphi^\star & = \arg \min_{g \in \mathcal{G}_{\mathrm{p}}, \varphi \in \Phi} L_c(g\varphi, w(\varphi)) + \lambda \cdot \widehat{\mathrm{TSF}}(w(\varphi), \varphi, g) \\ & \text{such that } w(\varphi) = \arg\min_{w} \mathrm{INV}(w, \varphi) \end{cases} \tag{RUDA}$$

where $\lambda > 0$ is a trade-off parameter and $L_c(g\varphi, w(\varphi)) := \mathbb{E}_S[-w(\varphi(X))\mathbf{Y} \cdot \log(g\varphi)]$.

7.2.2 Overall objective

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:

$$L_{\mathrm{INV}}(\theta_d|\theta_\varphi) = \frac{1}{n_S}\sum_{i=1}^{n_S} -\log(d(z_{S,i})) + \frac{1}{n_T}\sum_{i=1}^{n_T} -\log(1 - d(z_{T,i})) \tag{7.6}$$

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. The latter, noted $\mathbf{d}$, maps representations to the label space $[0, 1]^{\mathrm{C}}$ in order to obtain a proxy of the transferability error expressed as a domain adversarial objective:

$$\begin{aligned} L_{\mathrm{TSF}}(\theta_\varphi, \theta_{\mathbf{d}}|\theta_d, \theta_g) = \inf_{\mathbf{d}} \Bigg\{ & \frac{1}{n_S}\sum_{i=1}^{n_S} -w_d(z_{S,i})g(z_{S,i}) \cdot \log(\mathbf{d}(z_{S,i})) \\ & + \frac{1}{n_T}\sum_{i=1}^{n_T} -g(z_{T,i}) \cdot \log(1 - \mathbf{d}(z_{T,i})) \Bigg\} \end{aligned} \tag{7.7}$$

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:

$$L_c(\theta_g, \theta_\varphi|\theta_d) = \frac{1}{n_S}\sum_{i=1}^{n_S} -w_d(z_{S,i})y_{S,i} \cdot \log(g(z_{S,i})) \tag{7.8}$$

Finally, the optimization is then expressed as follows:

$$\begin{cases} \theta_\varphi & = \arg\min_{\theta_\varphi} L_c(\theta_g, \theta_\varphi|\theta_d) + \lambda \cdot L_{\mathrm{TSF}}(\theta_\varphi, \theta_{\mathbf{d}}|\theta_d, \theta_g) \\ \theta_g & = \arg\min_{\theta_g} L_c(\theta_g, \theta_\varphi|\theta_d) \\ \theta_d & = \arg\min_{\theta_d} L_{\mathrm{INV}}(\theta_d|\theta_\varphi) \end{cases} \tag{7.9}$$

Losses are minimized by stochastic gradient descent (SGD) where in practice $\inf_d$ and $\inf_{\mathbf{d}}$ are gradient reversal layers (Ganin and Lempitsky 2015). The trade-off

Algorithm 2 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.
  $\theta_d \leftarrow \theta_d - \eta_t \nabla_{\theta_d} L_{\text{INV}}(\theta_d | \theta_\varphi; \mathcal{B}_S, \mathcal{B}_T)$
  $\theta_{\mathbf{d}} \leftarrow \theta_{\mathbf{d}} - \eta_t \nabla_{\theta_{\mathbf{d}}} L_{\text{TSF}}(\theta_g, \theta_\varphi, \theta_{\mathbf{d}} | \theta_d, \tau_t)$
  $\theta_\varphi \leftarrow \theta_\varphi - \eta_t \nabla_{\theta_\varphi} \left(L_c(\theta_g, \theta_\varphi | \theta_d, \tau_t) - \alpha_t L_{\text{TSF}}(\theta_\varphi, \theta_{\mathbf{d}} | \theta_g, \theta_d, \tau_t) \right)$
  $\theta_g \leftarrow \theta_g - \eta_t \nabla_{\theta_g} L_c(\theta_g, \theta_\varphi | \theta_d, \tau_t)$
  $t \leftarrow t + 1$
end while

parameter λ is pushed from 0 to 1 during training. We provide an implementation in `Pytorch` (Paszke et al. 2019a) based on (Long et al. 2018), as exposed in Algorithm 2. For more details about the losses introduced in the present section, we refer to Section 5.5.

7.3 Experiments

7.3.1 Setup

Datasets

We investigate two digits datasets; **MNIST** and **USPS**), with 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. We studied the transfer tasks **A**→**W**, **W**→**A**, **A**→**D**, **D**→**A**, **D**→**W** and **W**→**D**.

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, the images that from classes 16 to 31 are duplicated 5 times. (refered as $5 \times [16 \sim 31]$), creating a label shift situation without removing samples in the dataset. 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 (Ganin and Lempitsky 2015) and CDAN (Long et al. 2018). 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

	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**

Table 7.1: Accuracy (%) on the **Office-31** dataset.

datasets, we report a weighted version of CDAN (entropy conditioning CDAN+E Long et al. 2018). For the label shifted datasets, we report IWAN (Zhang et al. 2018), a weighted DANN where weights are learned from a second discriminator, and CDAN$_w$ a weighted CDAN where weights 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 (He et al. 2016). We used the same hyper-parameters than (Long et al. 2018) which were selected by importance weighted cross-validation (Sugiyama, Krauledat, and MÃžller 2007). The trade-off parameter λ is smoothly pushed from 0 to 1 as detailed in (Long et al. 2018). 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: $\tau = \tau_{\text{min}} + 2(\tau_{\text{max}} - \tau_{\text{min}})/(1 + \exp(-\alpha p))$ where $\tau_{\text{max}} = 5, \tau_{\text{min}} = 1$, $p \in [0, 1]$ is the training progress as described in Section 7.1.3. In all experiments, α is set to 5, except for $5\% \times [0 \sim 5]$ where $\alpha = 15$.

7.3.2 Results

Unshifted datasets

On both **Office-31** (Table 7.1) and **Digits** (Table 7.2), RUDA performs similarly than CDAN. Simply performing the scalar product allows to achieve results obtained by multi-linear conditioning (Long et al. 2018). This presents a second advantage; when domains exhibit a large number of classes, our approach does not need to leverage a random layer as it is prescribed in (Long et al. 2018). 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. It demonstrates the need to design stronger inductive bias to improve adaptation in this setting.

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 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 (Johansson, Sontag, and Ranganath 2019; Zhao et al. 2019). 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).

7.3.3 Ablation

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**).

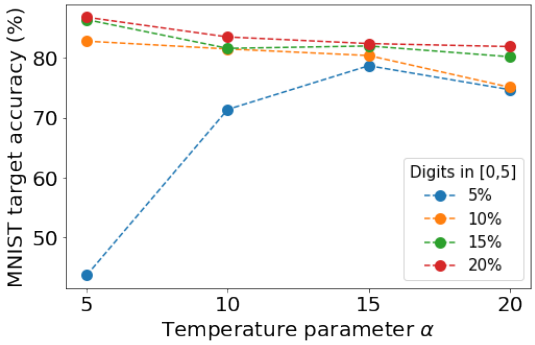


Figure 7.1: Effect of α.

Do we really need 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 these 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.

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**

Table 7.2: Accuracy (%) on the **Digits** dataset.

8 Target Consistency

Contents

Part II isolates the role of *inductive bias*, particularly in Chapter 6, to improve transferability of domain invariant representations. Part III is dedicated to their implementations. We have studied in Chapter 6 that two weak inductive biases, on both weights and representations, improve adaptation in the challenging context of label shift. This result validates our interest in building stronger inductive bias to improve the transferability of domain invariant representations.

In Section 6.2.1 of Chapter 6, we have provided a historical overview of the influential role of inductive bias in developing Machine Learning. In particular, *Semi-Supervised Learning* (SSL) has developed various inductive biases for learning a model from a handful of labelled data and abundant unlabelled data. We have reviewed the paradigm of SSL in 3.1.4. We recall that SSL differs from UDA by assuming that unlabelled data, *i.e.* the target data, is sampled from the same distribution as labelled data, *i.e.* the source data. Indeed, UDA focuses on the distribution shift between the source and the target data while SSL focuses on the problem of learning from small labelled samples.

An interesting difference between these two approaches is how the research and development of the methods have been historically conducted. In UDA, the algorithms are derived from a strong theoretical foundation, notably through the works of (Cortes et al. 2008) and (Ben-David et al. 2010a). In the case of SSL, the algorithms are derived by enforcing assumptions (*e.g.* the cluster assumption) that take advantage of the data structure. Thus, two very different visions of learning build the UDA and SSL paradigms. The present chapter bridges this gap through the lens of inductive bias.

Chapter 8 aims to provide a new understanding of the transferability of representations through the prism of the cluster assumption, a well-known SSL paradigm. The cluster assumption states that if samples are in the same cluster in the input space,

they are likely to be of the same class, as presented in Section 3.1.4. When enforced on unlabeled samples, the model benefits from a significant gain in generalization (Chapelle, Scholkopf, and Zien, Eds. 2009; Sohn et al. 2020; Xie et al. 2020) and robustness (Carmon et al. 2019; Hendrycks et al. 2020). In particular, we rely on the inductive bias enforced by the cluster assumption into the model to minimize the inductive transferability term that we have developed in Chapter 6. More precisely, target labels in the transferability term are replaced by the prediction from a classifier $g\varphi$ that verifies the cluster assumption. Formally, such classifier is obtained through a regularization that promotes the consistency of predictions to an input perturbation;

$$\mathrm{Err}_T(g\varphi) = \mathrm{Err}_T(g\varphi\tau) \tag{8.1}$$

where τ is a mapping from the input space $\mathcal{X}$ to itself that reflects a perturbation of inputs that conserves the label, *e.g.* a rotation should not modify the prediction since it does not change the semantic content of the image.

We organize the present Chapter as follows:

1. We show that domain invariance induces a significant model sensitivity to perturbations in the target domain, indicating that invariance is achieved by disregarding principles of robustness. Such evidence motivates our interest in enforcing the cluster assumption for improving the transferability of domain invariant representations.
2. We describe an algorithm for learning domain invariant representations while enforcing the cluster assumption.
3. We conduct an extensive empirical analysis on both classification and segmentation datasets to reach state-of-the-art methods based on invariant representations. Crucially, we show that enforcing the cluster assumption brings strong robustness to the model, a property that is not verified when learning domain invariant representations.

Chapter 8 is based on the publication and an oral presentation in a French national conference;

> Target Consistency for Domain Adaptation: when Robustness meets Transferability,
> *Yassine Ouali, Victor Bouvier, Myriam Tami, Céline Hudelot*,
> Conférence sur l'apprentissage Automatique (CAp), Saint-Étienne, 2021.

Yassine Ouali and Victor Bouvier contributed equally.

8.1 Invariance, Transferability and the Cluster Assumption

Estimating the lack of transferability of domain invariant representations is a difficult problem since it requires target labels that are unavailable in a scenario of UDA. Therefore, other tools need to be leveraged. Following the insight from (Shu et al. 2018), we hypothesize that violation of the cluster assumption in the target domain is a strong indicator of a lack of transferability. In such a case, a classifier that is not optimal in both domains exhibit a substantial sensitivity in the target domain to small input perturbations. Indeed, the violation of the cluster assumption is characterized by a decision boundary localized in high density regions of the target input space. To this purpose, we study a model through two analysis of robustness; the *Jacobian Norm* of the model (Novak et al. 2018) in Section 8.1.1 and the *Fourier Analysis* (Yin et al. 2019) in Section 8.1.2.

8.1.1 Sensitivity in the Target Domain

Jacobian norm as a proxy of generalization

We analyze the robustness of a model trained to minimize the source risk, through its sensitivity to small perturbations in the input space. We follow (Novak et al. 2018) and compute the mean Jacobian norm as a proxy of the generalization at the level of individual target samples, and as a measure of the local sensitivity of the model on target examples:

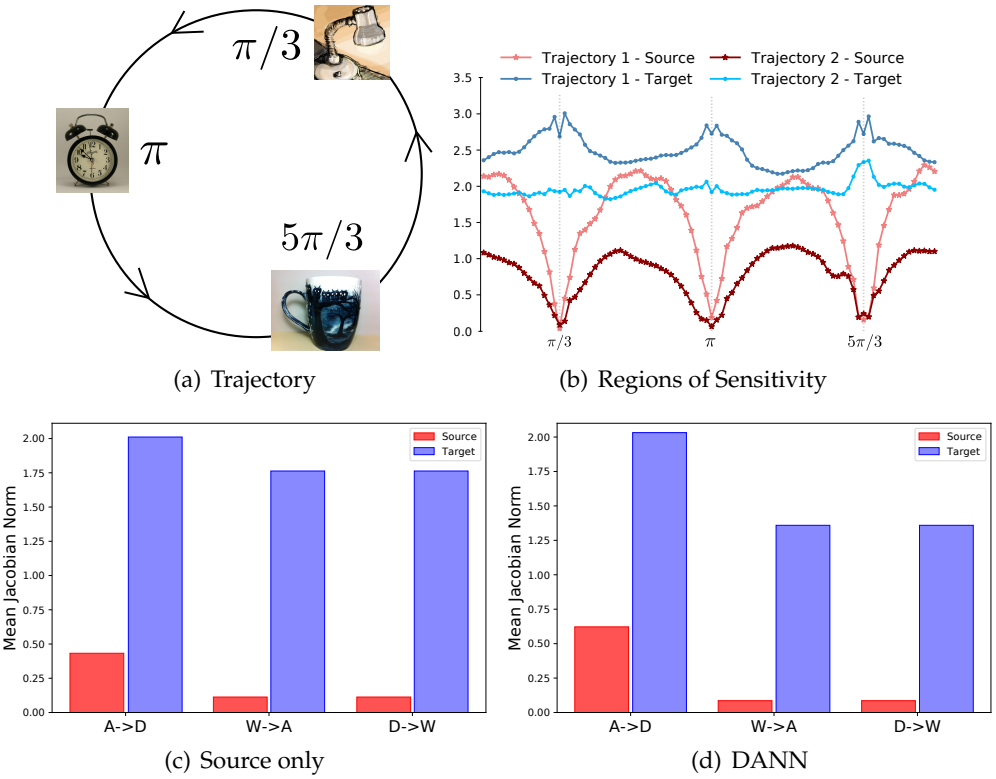
$$\mathbb{E}_T \left[\| \mathsf{J}(X) \|_F \right] \tag{8.2}$$

where $\mathsf{J}_{ij}(x) = \partial \hat{y}_i / \partial x_j$ is the Jacobian matrix, $\| \mathsf{J} \|_F$ is the Frobenius norm, and $\hat{y}_i$ is the output class probability for class i. For comparison, the source domain's sensitivity can be computed similarly over source instances. By language abuse, we will refer to sensitivity in source and target domains as source and target sensitivity, respectively.

Results

The results obtained on 3 transfer tasks from Office-31 ($\mathbf{A} \rightarrow \mathbf{D}$, $\mathbf{W} \rightarrow \mathbf{A}$, $\mathbf{D} \rightarrow \mathbf{W}$) are shown in Figures 8.1(c) and 8.1(d). As suspected, the target sensitivity is significantly higher compared to the source sensitivity. Importantly, when enforcing invariance of representations with Domain Adversarial Neural Networks (DANN (Ganin and Lempitsky 2015)), sensitivity in the target domain decreases (for tasks $\mathbf{W} \rightarrow \mathbf{A}$ and $\mathbf{D} \rightarrow \mathbf{W}$) while remaining significantly higher than the source sensitivity. This validates our concern on non-conservative domain adaptation: even after features alignment, the resulting classifier still violates the cluster assumption in the target domain. To further investigate the regions of sensitivity, we examine the function's behavior on and off the data manifold as it approaches and moves away from three anchor points. To this end, following (Novak et al. 2018), we analyze the behavior of the model near and away from target and source data. The protocol considers three data points x_1, x_2 and x_3 and we report the model sensitivity when doing a smooth transformation from x_1 to x_2, from x_2 to x_3 and from x_3 to x_1. We will study two cases: when x_1, x_2 and x_3 are of the same class *versus* when they do not belong to the same, through two types of trajectories:

1. an ellipse passing through three data points of different classes as illustrated in Figure 8.1(a),



(a) Trajectory (b) Regions of Sensitivity

(c) Source only (d) DANN

Figure 8.1: Sensitivity Analysis. (a) An illustration of the circular trajectory passing through three images of different classes. (b) Jacobian norm of source (**D**) and target (**A**) as the input traverses two elliptical trajectories: *Trajectory 1*: different classes. *Trajectory 2*: same classes, for a ResNet-50 (He et al. 2016) trained on source only. (c) and (d) The mean Jacobian norm on target and source domains of a ResNet-50 when trained on source only and with a DANN objective on three Office-31 tasks.

2. an ellipse passing through three data points of the same class.

Since linear combinations of images from the same class are likely to look like a realistic image, the second trajectory is expected to traverse overall closer to the data manifold. Figure 8.1(b) shows the obtained results. We observe that, according to the Jacobian norm, the model's sensitivity in the vicinity of target data is comparable to its sensitivity off the data manifold. Inversely, the model remains relatively stable in the neighborhood of source data and becomes unstable only away from them, further confirming our hypothesis.

8.1.2 Fourier Analysis

Formulation

To further examine the lack of target robustness of domain invariant based adaptation, we investigate a common hypothesis in robust deep learning (Hendrycks et al. 2020), where the lack of robustness is attributed to spurious high-frequency correlations that exist in the source data, that are not transferable to target data. To this end, we follow (Yin et al. 2019), and measure the model error after injecting an additive noise at different frequencies in the spectral representation of the image. Concretely, we resize all of the data to 96×96 images, we then add, at each iteration, 96×96 Fourier basis vector corresponding to an additive noise at a given frequency,

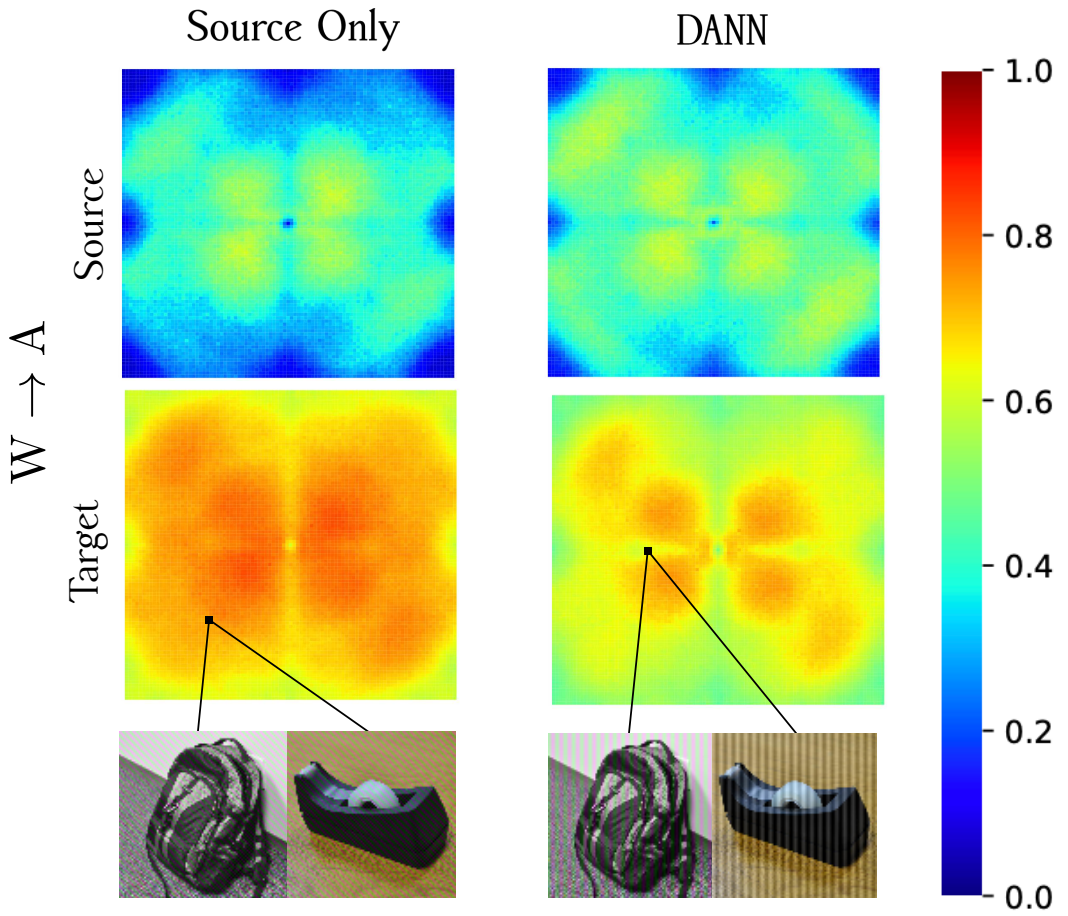


Figure 8.2: Heatmaps of two ResNet-50 where the former is trained on the source domain while the latter incorporates the target unlabelled data with DANN (Ganin and Lempitsky 2015). This corresponds in a matrix of pixels of size 96×96 where each coordinate corresponds to a basis vector in the Fourier space as presented in Equation 8.3, where we provide two examples of perturbed images. The intensity of the heatmap indicates the error rate, as presented in Equation 8.3, the lower the better. As suspected both models exhibit a lack of robustness in the target domain while being robust in the source domain. Interestingly, DANN improves the robustness in the target domain while remaining sensitive to perturbations.

and record the model error over either source or target data when such basis vector is added to each image individually. Formally, given an image $x \in \mathbb{R}^{d_1 \times d_2}$, we note $u = \mathbf{F}(x) \in \mathbb{C}^{d_1 \times d_2}$ the 2D Discrete Fourier Transform of x and $\mathbf{F}^{-1}(u)$ the Inverse Discrete Fourier Transform of u, that coincides with x *i.e.* $x = \mathbf{F}^{-1}(\mathbf{F}(x))$. Given an hypothesis h, we are then interested in the robustness of h when applied to $\tilde{x} = \mathbf{F}^{-1}(\tilde{u})$, where u is a noisy version of u, measured by;

$$\mathbb{E}\left[\mathbb{I}\left(h(\mathbf{F}^{-1}(\tilde{U})) \neq Y\right)\right] \text{ where } \tilde{U}_{i,j} = \begin{cases} U_{i,j} + \mathcal{N}(0, \sigma^2) \text{ if } (i,j) = (\tilde{i}, \tilde{j}) \\ U_{i,j} \text{ otherwise.} \end{cases} \tag{8.3}$$

where $\tilde{i}$ and $\tilde{j}$ are uniformly sampled on $\{1, ..., d_1\}$ and $\{1, ..., d_2\}$ respectively.

Analysis

The Figure C.1 shows the Fourier sensitivity heatmaps on source and target, for a ResNet-50 trained with two objectives; a model only trained on source data (source

only) and a model that uses unlabelled target data with DANN (Ganin and Lempitsky 2015). Each pixel of 96×96 heatmaps shows the error of the model when the inputs are perturbed by a single Fourier basis vector, in which the error corresponding to low-frequency noise is shown in the center, and high frequencies are away from the center. We observe that the model is highly robust on source across frequencies and the different objectives, but becomes quite sensitive to high-frequency perturbations on target when trained on source only or with a DANN objective. Again, this validates our concern that adaptation through invariant representations is insufficient to guarantee robustness in the target domain, which correlates strongly with the generalisation gap.

8.2 Algorithm

8.2.1 Consistency Regularization

To promote a more robust model and mitigate target sensitivity, we regularize the model predictions to be invariant to a set of perturbations applied to the target inputs. Concretely, we add to the objective function an additional Target Consistency term:

$$L_{\text{TC}}(\varphi, g) = L_{\text{VAT}}(\varphi, g) + L_{\text{AUG}}(\varphi, g) \tag{8.4}$$

$$= \mathbb{E}_T \left[\max_{\|r\| \leq \epsilon} ||h(X) - h(X + r)||^2 \right] + \mathbb{E}_T \left[||h(X) - h(\tilde{X})||^2 \right] \tag{8.5}$$

Similar to (Shu et al. 2018), the first term incorporates the locally-Lipschitz constraint by applying Virtual Adversarial Training (VAT) (Miyato et al. 2018) which forces the model to be consistent within the norm-ball neighborhood of each target sample. Additionally, the second term forces the model to embed a target instance x and its augmented version $\tilde{x}$ similarly to push for smooth neural network responses in the vicinity of each target data. With a carefully chosen set of augmentations, such a constraint makes sense since the semantic content of a transformed image is approximately preserved. Note that for more stable training, we follow Mean Teachers (MT) (Tarvainen and Valpola 2017) and use of an exponential moving average of the model to compute the target pseudo-labels (*i.e.* $h(x)$). Overall, L_{TC} is in-line with the cluster assumption by promoting consistency to a various set of input perturbations, thus, forcing the decision boundary to not cross high-density regions.

8.2.2 Augmentations

For visual domain adaptation, and based on the recent success of supervised image augmentations (Cubuk et al. 2019a; Lim et al. 2019; Cubuk et al. 2019b) in semi-supervised learning (Xie et al. 2020; Sohn et al. 2020) and robust deep learning (Yin et al. 2019; Hendrycks et al. 2020), we propose to use a rich set of state-of-the-art data augmentations to inject noise and enforce consistency of predictions on target domain. Specifically, we use augmentations from AutoAugment (Cubuk et al. 2019a). Upon each application, we sample a given operation o from all possible augmentations $\mathcal{O} = \{\text{equalize}, \ldots, \text{brightness}\}$. If the operation o is applicable with varying severities, we also uniformly sample the severity, and apply o to obtain the augmented target image $\tilde{x} = o(x)$. However, applying a single operation might be solved easily by a high capacity model by memorizing the specific perturbations. To overcome this, we generate more diverse augmentations

Figure 8.3: An example of the possible augmentations to be applied on a given input image.

by mixing multiple augmented images (see 8.4). We start by randomly sampling K operations from $\mathcal{O}$ and K convex coefficients α_i sampled from a Dirichlet distribution: $(\alpha_1, \ldots, \alpha_K) \sim \text{Dir}(1, \ldots, 1)$. The augmented image $\tilde{x}$ can then be obtained with an element-wise convex combination of the K augmented instances of x: $\tilde{x} = \sum_{i=1}^{K} \alpha_i o_i(x)$, impelling the model to be stable, consistent, and insensitive across a more diverse range of inputs (Zheng et al. 2016; Kannan, Kurakin, and Goodfellow 2018; Hendrycks et al. 2020).

8.2.3 Overall objective

Our model is trained by minimizing a trade-off between source Cross-Entropy (CE), the Transferability term (TSF) and Target Consistency (TC); given μ and ν two tunable hyper-parameters,

$$L(g, \varphi) := L_{\text{CE}}(g, \varphi) + \mu L_{\text{TSF}}(\varphi) + \nu L_{\text{TC}}(g, \varphi) \tag{8.6}$$

Enforcing the target consistency gives us the ability to control the trade-off between a low target sensitivity, *i.e.* a low violation of the cluster assumption and a low source risk. As described in (Shu et al. 2018), adding L_{TC} to the objective function reduces the hypothesis class $\mathcal{H}$ to only include classifiers that are robust on both

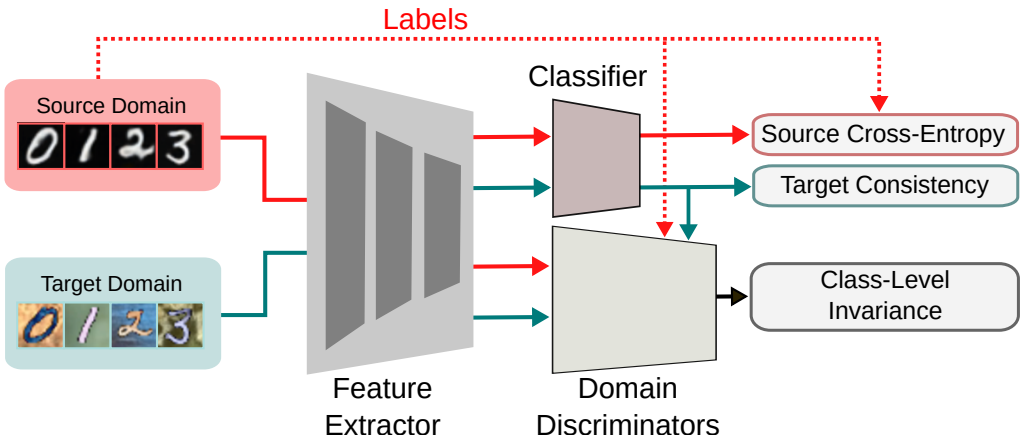


Figure 8.4: An overview of the proposed framework. In addition to training on the labeled source data, we enforce a Target Consistency (TC), imposing the cluster assumption over target data, and promoting a more robust model on the target domain. To amplify the effect of TC, we perform class-level invariance through the transferability loss while enforcing the cluster assumption, where a class specific discriminator is selected using either the source labels or the target predictions for the adversarial loss. Thus promoting positive feedback between decision boundary updates and representation alignment.

target and source domains following the trade-off described of Proposition 6.2.1. For more details about L_{TSF}, we refer to Section 5.5.

Our analysis from Chapter 6 on the role of inductive bias when learning domain invariant representations provides a new perspective. In particular, we assume that for some $\lambda > 0$;

$$\tilde{g}, \tilde{\varphi} = (g, \varphi) - \nu \cdot \nabla_{(g,\varphi)} L_{\text{TC}}(\varphi, g) \text{ leads to } \text{Err}_T(\tilde{g}\tilde{\varphi}) \leq \beta \text{Err}_T(g\varphi) \tag{8.7}$$

where $\beta < 1$, *i.e.* a strong inductive bias. Such statement follows the two examples 6.3.1 and 6.3.2 from Chapter 6.

To gain insight about this phenomenon, we consider a target sample x near the decision boundary (red squared pen in Figure 8.5) which is hard to adapt (the pen is confounded with a mug). Thus, its augmented version, $\tilde{x}$, is likely to have a different predicted class (red squared pens with a low opacity in 8.5). By enforcing TC, the model embeds x and $\tilde{x}$ similarly to incrementally push the decision boundary far from class boundaries. Such incremental change might result in correcting the predicted class label (green decision boundary in Figure 8.5). However, the underlying representations remain approximately the same, and the discriminator feedback reflects poorly this predicted labels change. Now, consider that domain invariance is achieved by leveraging one discriminator per predicted class, *i.e.* the transferability loss. The change of label due to the TC update will result in a switch of the discriminator used, subsequently reflecting the label change in the domain adversarial loss. This interaction between class-level invariance and decision boundary update is the key to the success of TC. Figure 8.5 illustrates such an interaction.

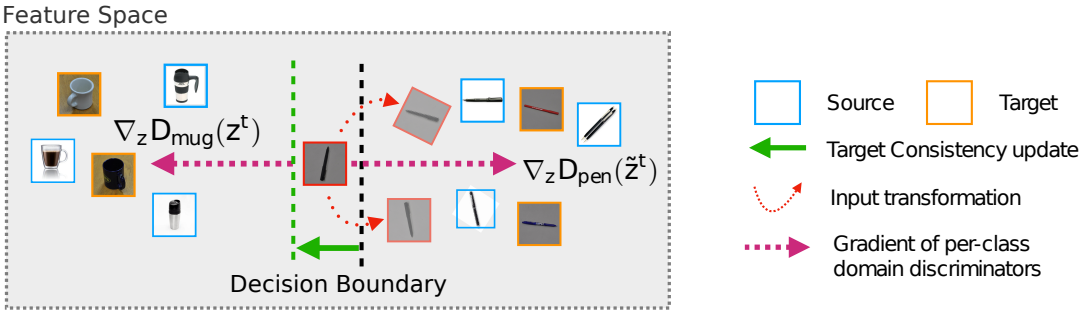


Figure 8.5: Effect of TC on the learned representations. Mugs and pens from the source (**A**) and target (**D**) domains of Office31 are pictured. The red squared pen, a target sample, is confounded with a mug due to spurious correlations *i.e.* upward orientation and black color. Input augmentations wipe out spurious correlations induced by the orientation, and the TC pushes the decision boundary to low density regions, correcting the predicted class. Before the TC update, the class-level discriminator encourages the pen to reach the high-density region of the incorrect class, *i.e.* the mug class. At this time, the class-level discriminator and TC gradients have opposite directions, indicating a negative interaction. The TC update allows the sample to cross the decision boundary. It ultimately changes the class-level discriminator, which now pushes the pen to the correct high-density region corresponding to its true class *i.e.* the pen class. At this time, the domain adversarial and TC gradients have similar directions, indicating a positive interaction. Crucially, the gradient of a vanilla domain discriminator (*i.e.* DANN) interacts poorly with the TC update since it does not modify the target representations distribution substantially. *Best viewed in color.*

8.3 Experiments

8.3.1 Setup

Datasets

Office-31 (Saenko et al. 2010) is the standard dataset for visual domain adaptation, containing 4,652 images in 31 categories divided across three domains: Amazon (**A**), Webcam (**W**), and DSLR (**D**). We use all six possible transfer tasks to evaluate our model. **ImageCLEF-DA**[1] is a dataset with 12 classes and 2,400 images assembled from three public datasets: Caltech-256 (**C**), ImageNet (**I**) and Pascal VOC 2012 (**P**), where each one is considered as separate domain. We evaluate on all possible pairs of the three domains. **Office-Home** (Venkateswara et al. 2017) is a more difficult dataset compared to Office-31, consisting of 15,00 images across 65 classes in office and home settings. The dataset consists of four widely different domains: Artistic images (**Ar**), Clip Art (**Ca**), Product images (**Pr**), and Real-World images (**Rw**). We conduct experiments on all twelve transfer tasks. **VisDA-2017** (Peng et al. 2017) presents a challenging simulation-to-real dataset, with two very distinct domains: **Synthetic**, with renderings of 3D models with different lightning conditions and from many angles; **Real** containing real-world images. We conduct evaluations on the **Synthetic** $\rightarrow$ **Real** task. For semantic segmentation experiments, we evaluate our method on the challenging **GTA5** $\rightarrow$ **Cityscapes** VisDA-2017 semantic segmentation task. The synthetic source domain is **GTA5** (Richter et al. 2016) dataset with 24,966 labeled images, while the real target domain is **Cityscapes** (Cordts et al. 2016) dataset consisting of 5,000 images. Both datasets are evaluated on the same classes, with the mean Intersection-over-Union (mIoU) metric.

[1] https://www.imageclef.org/2014/adaptation

Method	Office-31	ImageCLEF-DA	Office-Home	VisDA	VisDA (ResNet-101)
ResNet (He et al. 2016)	76.1	80.7	46.1	45.6	52.4
DANN (Ganin et al. 2016)	82.2	85.0	57.6	55.0	57.4
CDAN (Long et al. 2018)	87.7	87.7	65.8	70.0	73.7
TAT (Liu et al. 2019a)	88.4	88.9	65.8	71.9	-
BSP (Chen et al. 2019c)	88.5	-	66.3	-	75.9
TransNorm (Wang et al. 2019)	89.3	88.5	67.6	71.4	-
Ours	**89.6**	**89.5**	**69.0**	**77.5**	**79.0**

Table 8.1: Average accuracy (%) of all tasks on image classification benchmarks for UDA. We compare our approach with similar methods based on invariant representations, evaluated using the same protocol. Results are obtained with a ResNet-50 unless specified otherwise. For detailed per task results, see the Appendix.

Protocol

We follow the standard protocols for UDA (Long et al. 2017; Long et al. 2018; Chen et al. 2017). We train on all labeled source samples and all unlabeled target samples and compare the classification accuracy based on three random experiments for classification and the mIoU based on a single run for segmentation. For classification, we use the same hyperparameters as CDAN (Long et al. 2018) and adopt ResNet-50 (He et al. 2016) as a base network pre-trained on ImageNet dataset (Deng et al. 2009). As for TSF and TC hyperparameters, we use $K = 4$, $\mu = 1$ and $\nu = 10$. We note that the method performs comparatively on a wide range of hyperparameter values making it robust for practical applications. For segmentation, we follow ADVENT (Vu et al. 2019) and use the same experimental setup with Deeplab-V2 (Chen et al. 2017) as the base semantic segmentation architecture with a ResNet-101 backbone and a DCGAN discriminator (Radford, Metz, and Chintala 2016). We employ `PyTorch` (Paszke et al. 2019a) and base our code on official implementations of CDAN (Long et al. 2018) and ADVEN (Vu et al. 2019).

Losses	Avg
L_{TSF}	56.7
$+L_{\text{VAT}}$	57.1
$+L_{\text{AUG}}$	58.1
$+L_{\text{VAT}} + L_{\text{AUG}}$	58.6
$+L_{\text{VAT}} + L_{\text{AUG}}$ w/ MT	**58.9**

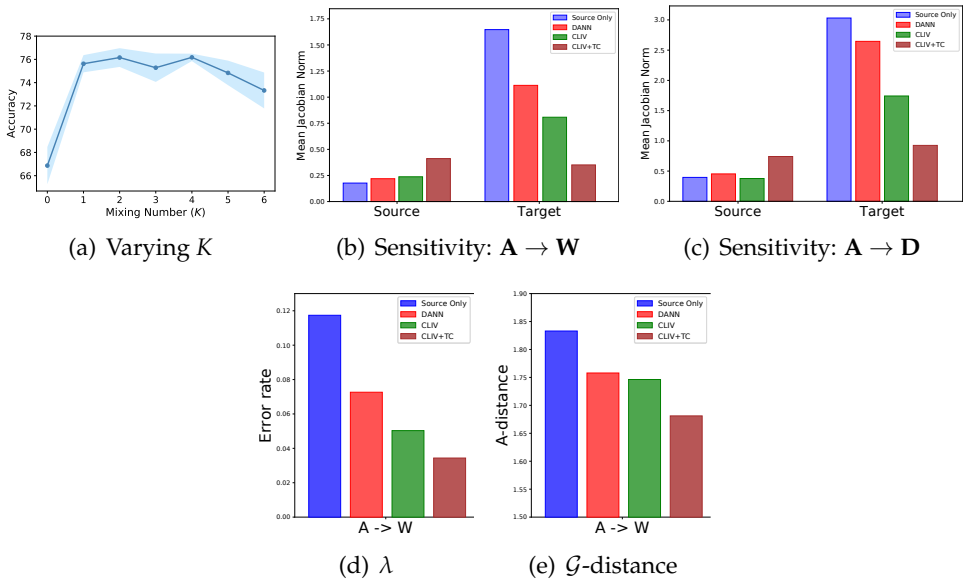
Table 8.2: Acc (%) on the 5 hardest Office-Home tasks for TC ablation.

DeepLab v2	
Method	mIoU
Adapt-SegMap (Tsai et al. 2018)	42.4
AdvEnt (Vu et al. 2019)	43.8
Ours	**44.9**

Table 8.3: mIoU on **GTA5** → **Cityscapes**.

Table 8.4: Avg Acc (%) of the 5 hardest Office-Home tasks for TC coupled with different adversarial losses.

$L_{\text{adv}} =$	L_{DANN}	L_{CDAN}	L_{TSF}
L_{adv}	47.6	53.4	56.7
$+L_{\text{VAT}}$	48.0	55.1	57.1
$+L_{\text{AUG}}$	51.3	55.7	58.1
$+L_{\text{VAT}} + L_{\text{AUG}}$	51.4	56.9	58.6
$+L_{\text{VAT}} + L_{\text{AUG}}$ w/ MT	51.0	56.0	**58.9**



(a) Varying K (b) Sensitivity: $\mathbf{A} \rightarrow \mathbf{W}$ (c) Sensitivity: $\mathbf{A} \rightarrow \mathbf{D}$

(d) λ (e) $\mathcal{G}$-distance

Figure 8.6: Analyses. (a) Accuracy on VisDA-2017 with different number of mixed augmentations K. (b) and (c) The effect of TC on the target and source sensitivity for two Office-31 tasks ($\mathbf{A} \rightarrow \mathbf{W}$ and $\mathbf{A} \rightarrow \mathbf{D}$). (d) The error λ of the ideal joint hypothesis h^{λ}. (e) $\mathcal{G}$ measure of domain discrepancy d_G in the representation space.

8.3.2 Results

For clarity and compactness, the average accuracy results of all tasks on all standard classification benchmarks for UDA are reported in Table 8.1. The proposed method outperforms previous adversarial methods on all datasets. The gains are substantial when the source and target domain are more dissimilar, as in VisDA dataset. We conjecture that this is a result of a large number of target instances available, enabling us to extract a significant amount of training signal with TC objective term to enforce the cluster assumption. Additionally, the method performs well with many categories, as it is the case for Office-Home dataset. Such gain is a result of the class-level invariance, which is empowered as the number of classes grows. We observe overall smaller improvements on Office-31 due to its limited size, and ImageCLEF-DA since the three domains are visually more similar. We further demonstrate the generality of the proposed method by conducting additional experiments on GTA5 $\rightarrow$ Cityscapes task for semantic segmentation (Table 8.3), and observe a gain of 2.5 points over the baseline Adapt-SegMap (Tsai et al. 2018), confirming the flexibility of TC and its applicability across UDA tasks. Detailed results are available in Appendix C.1.

8.3.3 Ablations

To examine the effect of each component of our proposed method, we conduct several ablations on the 5 hardest tasks on Office-Home, with and without the TC term, and with different variations of the TC loss. The results are reported in Table Table 8.2. We observe that adding a consistency term, either VAT or AUG, results in a higher accuracy across tasks, with better results when smoothing in the vicinity of each target data point within the data manifold with AUG, instead of the adversarial direction using VAT. Their combination, with Mean Teacher (MT), results in an overall more performing model. We also conduct an ablation study on the effect of varying the mixing number K to produce more diverse target images. Figure

8.6(a) shows the results. Overall, we observe a slight improvement and more stable results when K is increased, but over a certain threshold, the degree of noise becomes significant, heavily modifying the semantic content of the inputs and hurting the model's performance. Most importantly, to show the importance of coupling TC with TSF, we pair TC with DANN and CDAN losses. The obtained results in 8.4 show lower average accuracy and minimal gains when enforcing the cluster assumption in conjunction with such adversarial losses, confirming the importance of imposing class-level invariance when applying TC.

8.3.4 Analyses

Sensitivity Analysis. To investigate the impact of TC on the model sensitivity, we compare the mean Jacobian norm of models trained with various objectives (Figures 8.6(b), 8.6(c)). TC coupled with TSF, greatly improves the model's robustness on target, with a small increase in the source sensitivity. We report the robustness of our consistency loss in Figure C.1 following the Fourier analysis.

Ideal Joint Hypothesis and Distributions Discrepancy. We evaluate the performances of the ideal joint hypothesis, which can be found by training an MLP classifier on top of a frozen features extractor on source and target data with labels. Figure 8.6(d) provides empirical evidence that TC produces a better joint hypothesis h^λ, thus more transferable representations. Additionally, as a proxy measure of domain discrepancy (Ben-David et al. 2010a), we compute the $\mathcal{G}$-distance (see definition 3.2.3), defined as $d_\mathcal{G} = 2(1 - 2\varepsilon)$, with ε as the error rate of a domain classifier trained to discriminate source and target domains. Figure 8.6(e) shows that TC decreases $d_\mathcal{G}$, implying a better invariance.

Qualitative Analysis. As shown in Figure 8.7, the method produces locally consistent and globally coherent predictions for semantic segmentation.

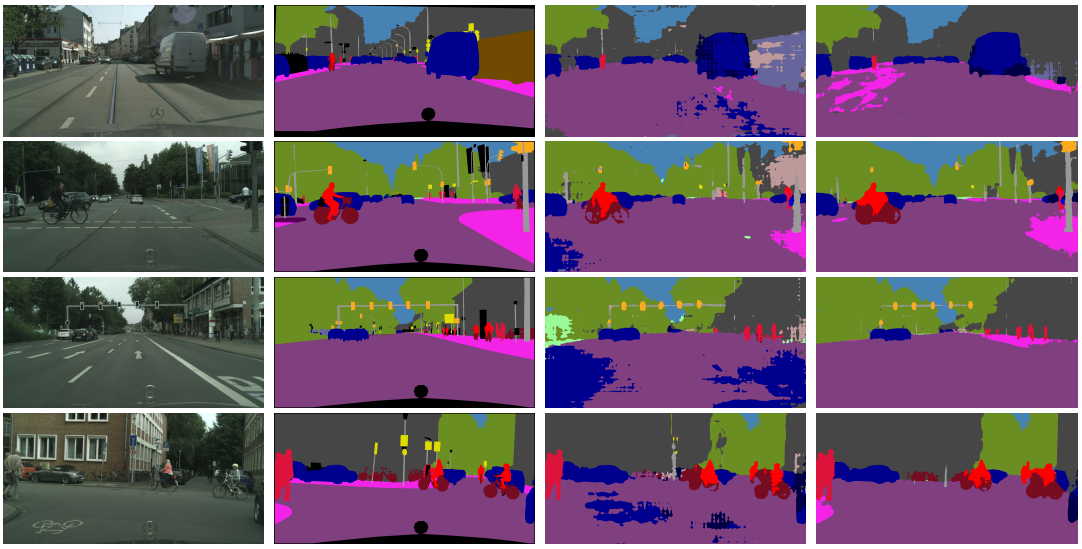


Figure 8.7: Qualitative Results on GTA5 → Cityscapes.

9 Active Domain Adaptation

Contents

Part II studies the transferability of domain invariant representations from the theoretical point of view. Chapter 5 introduces a new error term that involves target labels, making it intractable in a standard scenario of unsupervised Domain Adaptation. Nevertheless, Chapter 6 shows that one can estimate the transferability term when provided with a strong inductive bias. In Part III, we present some implementations of inductive bias to improve transferability of domain invariant representations. In particular, enforcing the consistency of predictions in the target domain improves adaptation performances substantially as presented in Chapter 8. However, most real-world use cases may not benefit from such a strong inductive bias. Indeed, our analysis about target consistency relies on the design of data augmentations that are valid only when dealing with images data.

The present chapter bridges the gap between Unsupervised Domain Adaptation to Active Learning (Settles 2009), where a user can query an *Oracle*, *i.e.* an expert, the labels of some samples in the target domain. We interpret the access to an Oracle as side information to perform adaptation, which we relate to our analysis of inductive bias of Chapter 6. Thus, we are back to building the best interaction between an adaptation algorithm and the Oracle that includes two crucial components. The first focuses on incorporating efficiently a small amount of target labelled samples with source labelled samples. The second identifies the most relevant target samples to annotate. The former is close to *Semi-Supervised Domain Adaptation* (SSDA). The latter is the subject of our contribution to *Active Domain Adaptation* (ADA). Our contribution is to select target samples that are the more susceptible to improve the transferability of representations during unsupervised domain adaptation.

We organize the present chapter as follows. We first provide an overview of Active Learning while considering similar learning paradigms, notably supervised, semi-supervised learning and adaptation. Second, we present a new criterion that allows

quantifying the lack of transferability of a target sample, defined as a norm of an embedding called **Sage** (**S**tochastic **a**dversarial **g**radient **e**mbedding). Crucially, such criterion allows deriving a new query for Active Domain Adaptation that meets the requirements of modern Active Learning by promoting the selection of a diverse set of target samples for which model's predictions are uncertain. Third, we analyse theoretically our proposal demonstrating one can interpret it as an inductive bias, as defined in Chapter 6. Finally, we conduct an empirical study of Sage for AL. In particular, we show that Sage achieves a new state-of-the-art for Active Domain Adaptation. Importantly, with a comparable labelling budget, Sage performs better than its semi-supervised counterpart while having more realistic assumptions for applications where one can not assume on target sample per class is available.

Chapter 9 is based on the publication and an oral presentation in a worshop hosted by international conference;

> Stochastic Adversarial Gradient Embedding for Active Domain Adaptation,
> *Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami and Céline Hudelot*,
> Interactive Adaptive Learning workshop (IAL),
> Colocated with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Bilbao (Basque Country), Online, 2021.

9.1 Preliminaries

Acquiring a large labelled dataset can be cost-prohibitive in most real-world applications. *Active Learning* (AL) (Settles 2009) focuses on selecting a small amount of data to annotate to achieve comparable performances that standard Supervised Learning with extensive labelled data. Thus, AL builds upon a crucial assumption that an Oracle, *e.g.* an expert, exists and can provide the label ground truth of a given sample. As a result, AL is often organized as follows; we query a subset from a pool of unlabelled data to send to the Oracle for annotation. Based on the label obtained from this subset of samples, we train a model with supervision. Using the model's information, we select a novel subset of unlabelled data and so on. We present such paradigm in Algorithm 10 and Figure 9.1.

There is an extensive literature on Active Learning (Settles 2009) that can be divided into two schools; *uncertainty* and *diversity*. The first aims to annotate samples for which the model has uncertain prediction, *e.g.* samples are selected according to their entropy (Wang and Shang 2014) or prediction margin (Roth and Small 2006), with some theoretical guarantees (Hanneke et al. 2014; Balcan, Beygelzimer, and Langford 2009). The second focuses on annotating a representative sample of the data distribution, *e.g.* the Core-Set approach (Sener and Savarese 2018) selects samples that geometrically cover the distribution. Several approaches also propose a trade-off between uncertainty and diversity, *e.g.* (Hsu and Lin 2015) that is formulated as a bandit problem. Recently, the work (**ash**) introduces Badge, a gradient embedding, which is an embedding achieving a state-of-the-art trade-off between uncertainty and diversity when performing AL with deep neural networks.

Guiding adaptation by selecting for annotation a pool of target unlabelled instances is a relatively new paradigm, referred to as *Active Domain Adaptation* (ADA). To our knowledge, only a few prior works address ADA (Chattopadhyay et al. 2013; Rai et al. 2010; Saha et al. 2011; Su et al. 2020). In particular, the recent work of Su et al. (Su et al. 2020) is the first that studies the problem of learning domain invariant representations by AL. Active Domain Adaptation exhibits fruitful similarities with the paradigm of Semi-Supervised Domain Adaptation (SSDA), for instance through the principle of Mini-Max Entropy (MME) (Saito et al. 2019). SSDA typically assumes that at least one target labelled sample represents a class, thus involving information

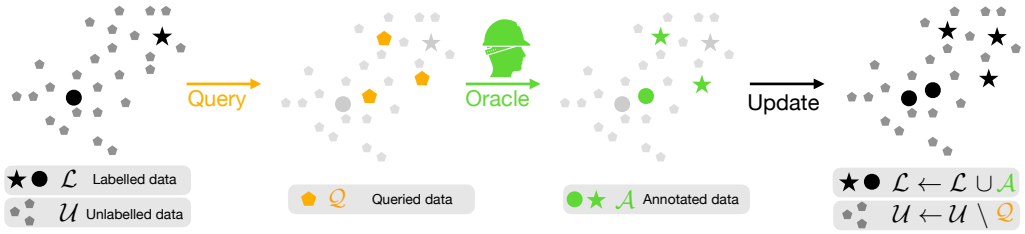


Figure 9.1: Overview of *Active Learning* (AL). AL aims to interactively annotate the smallest set of unlabelled samples to obtain the best generalization performance. AL is an iterative process that loops on three steps. The first step, called the *Query step* consists in identifying a set of samples $\mathcal{Q}$ among the unlabelled samples $\mathcal{U}$. Those samples are selected according to our belief their annotation may help to improve the model, quantified by a query function $q(x)$ for $x \in \mathcal{U}$. The design of $q(x)$ is the focus of AL. The second step consists in sending $\mathcal{Q}$ for requesting labels to build a novel set of annotated samples $\mathcal{A} := \{(x, \text{Oracle}(x)) : x \in \mathcal{Q}\}$. The third and last step, consists in updating the pool of labelled data ($\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{A}$) and the pool of unlabelled data ($\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{Q}$), then retraining a model.

	Paradigm	Source		Target	
		$\mathcal{L}$	$\mathcal{U}$	$\mathcal{L}$	$\mathcal{U}$
Passive	SSL	Small	Large		
	UDA	Large			Large
	SSDA	Large		Small	Large
Active	AL	Queried			
	SSAL	Queried	Large		
	ADA	Large		Queried	Large

Table 9.1: Positioning of Active Domain Adaptation with respect to other learning paradigms; Semi-Supervised Learning (SSL) (Chapelle, Scholkopf, and Zien 2009), Unsupervised Domain Adaptation (UDA) (Quinonero-Candela et al. 2009; Pan and Yang 2009), Semi-Supervised Domain Adaptation (SSDA) (Saito et al. 2019), Active Learning (AL) (Settles 2009), Semi-Supervised Active Learning (SSAL) (Gao et al. 2020) and Active Domain Adaptation (ADA) (Su et al. 2020). We divide learning paradigms into two categories, whether it is **Active** or **Passive**. We discriminate paradigms according to assumption about the labelled dataset and unlabelled data in term of size (small / large), acquisition (queried) and if distribution shifts (source or target).

about target labels. Therefore, SSDA is built on assumptions that are unlikely to be met in practice. We provide a positioning of ADA with respect to related paradimgs in Table 9.1.

9.2 Method

9.2.1 Motivations

Gradient-based selection, as shown in Badge (Ash et al. 2020a), is promising in AL. In contrast to Badge, which focuses on the network's predictions, we discuss the role of representations' transferability. To this purpose, we introduce, in the following, the *adversarial gradient* that reflects the lack of transferability of a target sample. From this gradient, we expose a query that efficiently incorporates the domain shift problem in ADA. Let a target sample $x \sim p_T$ with representation $z := \varphi(x) \in \mathbb{R}^m$, we start by describing the effect of annotating the sample x on the gradient descent

Algorithm 3 Active Learning

Input: A set of unlabelled target samples $\mathcal{U}$, an hypothesis class $\mathcal{H}$, a learning algorithm $\mathtt{A}$, a query q, budget b, annotation rounds r:

$\mathcal{L} \leftarrow \{\}$ ▷ Initializes the labelled samples.
$h \leftarrow \mathtt{Init}(\mathcal{H})$ ▷ Initializes model.
for r rounds of annotations **do**
  $\mathcal{Q} \leftarrow q(\mathcal{U}, h, b)$ ▷ Selects samples for annotation.
  $\mathcal{A} \leftarrow \text{Oracle}(\mathcal{Q})$ ▷ Sends samples to an Oracle.
  $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{A}$ ▷ Adds newly labelled samples.
  $\mathcal{U} \leftarrow \mathcal{U} \backslash \mathcal{Q}$ ▷ Removes newly labelled samples.
  $h \leftarrow \mathtt{A}(\mathcal{L}, \mathcal{U})$ ▷ Learns a model.
end for
Return: h

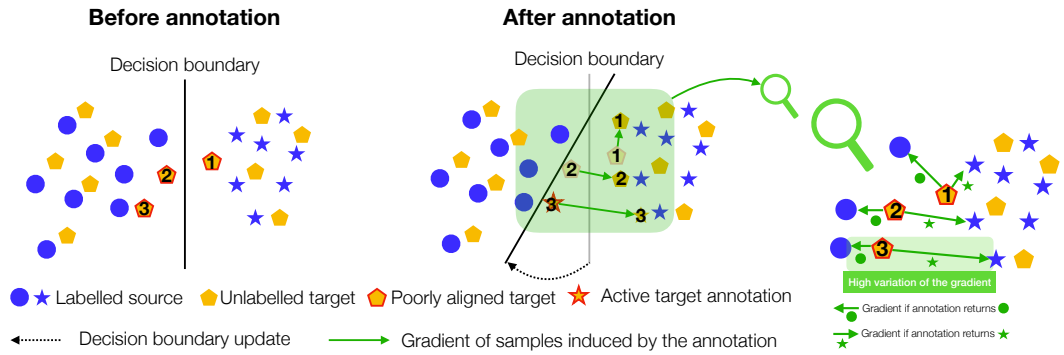


Figure 9.2: Effect of annotation of a target sample selected by Sage (*best viewed in colors*). Binary classification problem ($\bullet$ *vs* $\star$) where source samples are blue and target samples are orange. Before annotation, the class-level alignment is not satisfactory leading to a potential negative transfer (poorly aligned target samples tagged as **1**, **2** and **3**). We estimate which sample should be primarily annotated by measuring the variation of the representations' transferability gradient, before and after annotation. We observe the highest variation is obtained for target sample **3**, which is sent to an oracle. The oracle annotation returns class $\star$, validating the suspicion of negative transfer. This leads to an update of the decision boundary, which pushes **1**, **2**, and **3** into class $\star$, resulting in a better class-level alignment of representations.

update of 9.2. We define the *adversarial gradient* v_x of x as the gradient of the discriminator loss w.r.t the representation z:

$$\mathsf{v}_x := -\frac{\partial \log(1-\mathbf{d}(z))}{\partial z} \in \mathbb{R}^{C\times m}, \text{ where } \mathbf{d}(z) \in \mathcal{D}_C \tag{9.1}$$

Following the expression of the transferability loss L_{TSF} (See Chapter 7) that we recall;

$$L_{\text{TSF}}(\varphi, g, \mathbf{d}) = \mathbb{E}_S[\mathbf{Y}\cdot\log(\mathbf{d}(Z))] + \mathbb{E}_T[g(Z)\cdot\log(1-\mathbf{d}(Z))] \tag{9.2}$$

where $\varphi \in \Phi, g \in \mathcal{G}_p$ and $\mathbf{d} \in \mathcal{D}_C$, see Section 5.5. The contribution of a sample x to the gradient update (Equation 9.2), before and after its annotation, is:

$$\underbrace{\left\{\theta \leftarrow \theta - \alpha\frac{\partial z}{\partial \theta}\cdot(\hat{y}\cdot\mathsf{v}_x)\right\}}_{\textit{Before annotation}} \longrightarrow \underbrace{\left\{\theta \leftarrow \theta - \alpha\frac{\partial z}{\partial \theta}\cdot(\mathbf{y}\cdot\mathsf{v}_x)\right\}}_{\substack{\textit{After annotation}\\ y\sim\text{Oracle}(x)}}$$

where $\partial z/\partial\theta$ is the jacobian of the representations with respect to the deep network parameters θ *i.e.* $z := \varphi(x;\theta)$, $\hat{y} = h(x) := g\varphi(x;\theta), g \in \mathcal{G}_p$ is the current label estimation and α is some scaling parameter. Before the annotation, the gradient vector can be written as a weighted sum of v_x *i.e.* $\hat{y}\cdot\mathsf{v}_x \in \mathbb{R}^m$, reflecting the class probability of x. Annotating the sample x has the effect of setting, once and for all, a direction of the gradient $(y\cdot\mathsf{v}_x)$. Based on this observation, we can measure the annotation procedure's ability to learn more transferable representations by its tendency to change the path of the gradient descent *i.e.* how $y\cdot\mathsf{v}_x$ may differ with $\hat{y}\cdot\mathsf{v}_x$. We provide a high-level overview of the method in Figure 9.2.

9.2.2 Positive Orthogonal Projection (POP)

In the rest of the paper, we consider $\mathsf{v}_x \in \mathbb{R}^{C\times m}$ as a stochastic vector of $\mathbb{R}^m$ with realizations lying in $\mathcal{V}_x := \{\mathsf{v}_x^1, ..., \mathsf{v}_x^C\}$ where $\mathsf{v}_x^c = (-\partial \log(1 - \mathbf{d}(z))/\partial z)_c$. When provided the label through an oracle, *i.e.* $y \sim \text{Oracle}(x)$, we obtain $\mathsf{v}_x^y \in \mathcal{V}_x$, a realization of v_x. Before annotation, the direction of the gradient is the *mean* of v_x where $\mathcal{V}_x$ is provided with the class probability given by the classifier's output $h(x)$. More precisely, the probability of observing $\tilde{\mathsf{v}}_x = \tilde{\mathsf{v}}_x^c$ is $h(x)_c$, then, the *mean* of v_x, noted $\mathbb{E}_h[\mathsf{v}_x]$ is defined as follows:

$$\mathbb{E}_h[\mathsf{v}_x] := \mathbb{E}_{y\sim h(x)}\left[\mathsf{v}_x^y\right] = h(x) \cdot \mathsf{v}_x \in \mathbb{R}^m \tag{9.3}$$

Therefore, the tendency to modify the direction of the gradient is reflected by a high discrepancy between $\mathbb{E}_h[\mathsf{v}_x]$ and v_x^y for $y \sim \text{Oracle}(x)$. To quantify this discrepancy, we consider variations in both direction and magnitude. To find a good trade-off between these two requirements, we remove the mean direction of the gradient $\mathbb{E}_h[\mathsf{v}_x]$ to v_x by computing a *Positive* Orthogonal Projection (POP):

$$\tilde{\mathsf{v}}_x := \mathsf{v}_x - \lambda \mathbb{E}_h[\mathsf{v}_x] \tag{9.4}$$

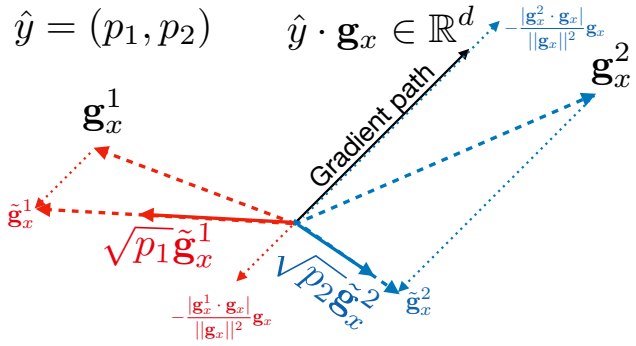
where $\lambda := |\mathsf{v}_x \cdot \mathbb{E}_h[\mathsf{v}_x]|/||\mathbb{E}_h[\mathsf{v}_x]||^2$. In the following, we motivate the use $|\mathsf{v}_x \cdot \mathbb{E}_h[\mathsf{v}_x]|$ rather than $\mathsf{v}_x \cdot \mathbb{E}_h[\mathsf{v}_x]$ for the standard orthogonal projection. On the one hand, if the annotation provides a gradient with the same direction as the expected gradient *i.e.* the annotation reinforces the prediction, $\tilde{\mathsf{v}}_x$ is null. On the other hand, if the annotation provides a gradient with an opposite direction to the expected gradient *i.e.* the annotation contradicts the prediction, the norm of $\tilde{\mathsf{v}}_x$ increases. Therefore, target samples x for which we expect the highest impact on the transferability, are those with the highest norm of $\tilde{\mathsf{v}}_x$. Since λ involves an absolute value, we refer to it as a *positive* orthogonal projection. An illustration is provided in Figure 9.3. Since $\tilde{\mathsf{v}}_x$ is stochastic, we need additional tools to define a norm operator properly on it.

9.2.3 Stochastic Adversarial Gradient Embedding (Sage)

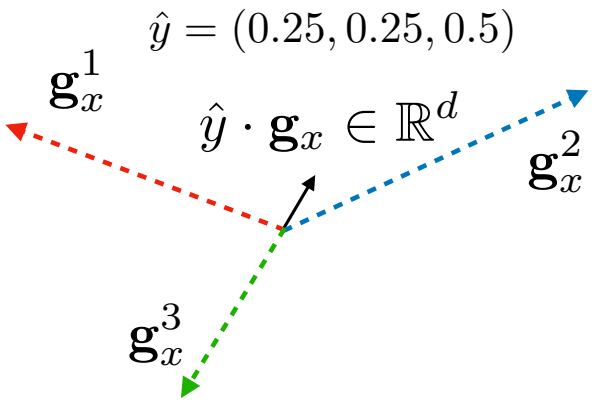
It seems natural to quantify the norm of the stochastic vector $\tilde{\mathsf{v}}_x$ as the square root of the mean of $\tilde{\mathsf{v}}_x$'s norm: $||\tilde{\mathsf{v}}_x||_h := (\mathbb{E}_{y\sim h(x)}\left[||\tilde{\mathsf{v}}_x^y||^2\right])^{1/2}$. However, given x_1 and x_2, how to quantify the discrepancy between v_{x_1} and v_{x_2}? The difficulty results from the fact that $h(x_1) \neq h(x_2)$ in general. Simply using $\mathbb{E}_{y_1\sim h(x_1), y_2\sim h(x_2)}\left[||\mathsf{v}_{x_1}^{y_1} - \mathsf{v}_{x_2}^{y_2}||^2\right])^{1/2}$ leads to an operator that returns a non-null discrepancy between x and itself if $h(x)$ is not a one-hot vector. To address this issue, we suggest to embed x, through a mapping S named *Stochastic adversarial gradient embedding* (Sage):

$$\mathsf{S}(x) := (\sqrt{h(x)_1}\tilde{\mathsf{v}}_x^1, ..., \sqrt{h(x)_C}\tilde{\mathsf{v}}_x^C) \in \mathbb{R}^{C\times m} \tag{9.5}$$

By choosing $\sqrt{h}$, we guarantee that $||\mathsf{S}(x)|| = ||\tilde{\mathsf{v}}_x||_h$ while offering a proper discrepancy between v_{x_1} and v_{x_2} with $||\mathsf{S}(x_1) - \mathsf{S}(x_2)||$. Crucially, both the norm and the distance computed on Sage do not involve the target labels, making it relevant for UDA since target labels are unknown. An illustration of Sage is provided in Figure 9.3.



(a) Sage



(b) Poor local min

Figure 9.3: (a) Visualisation of $\mathsf{S}(x) = (\sqrt{p_1}\tilde{\mathsf{v}}_x^1, \sqrt{p_2}\tilde{\mathsf{v}}_x^2)$. Here $\tilde{\mathsf{v}}_x^2 \perp (\hat{y} \cdot \mathsf{v}_x)$ since $\hat{y} \cdot \mathsf{v}_x$ and v_x^2 have a similar direction while $|\tilde{\mathsf{v}}_x^1 \cdot \mathsf{v}_x| \geq |\mathsf{v}_x^1 \cdot \mathsf{v}_x|$ since v_x^1 as a component in the opposite direction of $\hat{y} \cdot \mathsf{v}_x$. (b) Illustration of a case where the transferability loss is close to a local minimum ($\hat{y} \cdot \mathsf{v}_x \approx 0$), but the stochastic gradients (v_x^y for $y \in \{1, 2, 3\}$) have a high norm. Here, the annotation chooses one of the gradients resulting in a strong update of the model.

9.2.4 Increasing Diversity of Sage (`k-means++`)

As aforementioned, the higher the norm of $||\mathsf{S}(x)||$, the greater the expected impact of annotating sample x on the transferability of representations. A naive strategy of annotation would be to rank target samples by their Sage norm ($||\mathsf{S}(x)||$). The drawback is to acquire labels for a not IID batch from the target distribution, a problem referred to as the challenge of *diversity* in AL (Settles 2009). In certain pathological cases (e.g. the selection of very similar samples or samples of the same class), the IID violation may degrade the performance in the target domain. To label useful target samples (*i.e.* high $||\mathsf{S}(x)||$) while acquiring a representative batch of the target distribution, we follow (**Ash**) by selecting samples with high $||\mathsf{S}(x)||$ which span in various directions. This is performed using the `k-means++` initialization (Arthur and Vassilvitskii 2006). The procedure `Sage` is detailed in Algorithm 4 for a given budget b of annotation. Importantly, sampling diverse target samples with high impact on transferability results from the construction of an embedding (Sage) suitable with `k-means++`.

Algorithm 4 `Sage`$(\mathcal{U}_T, b, f, \varphi, \mathbf{d})$: Sage with diversity (`k-means++`)

Input: $\mathcal{U}_T$: Unlabelled target data, budget b, representation φ, classifier f, discriminator $\mathbf{d}$

Computes $\mathsf{S}(x_u)$ for $x_u \in \mathcal{U}_T$ ▷ Depends on both f and φ.
$\mathcal{A} \leftarrow \{\text{argmax}_{x_u \in \mathcal{U}_T} ||\mathsf{S}(x_u)||\}$ ▷ Select sample with the highest Sage norm.
while $|\mathcal{A}| < b$ **do** ▷ Apply `k-means++` on Sage embedding.
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{\underset{x_u \in \mathcal{U}_T}{\text{argmax}} \min_{x_a \in \mathcal{A}} ||\mathsf{S}(x_u) - \mathsf{S}(x_a)||\}$
end while
Return $\mathcal{A}$

9.2.5 Semi-Supervised Domain Adaptation (SSDA)

SSDA regularizer

When acquiring labels in the target domain, we are in the Semi-Supervised Domain Adaptation (SSDA) setting. To this purpose, we note $\mathcal{L}_S$ and $\mathcal{L}_T$ the sets of labelled samples from the source and the target domains, respectively. We study three strategies, referred to as $S \cup T$, $S + T$ and MME (Saito et al. 2019). They incorporate labelled samples into adaptation through an additional loss Ω, called a SSDA regularizer:

$$\Omega_{S \cup T}(f, \varphi) := L_{\mathcal{L}_S \cup \mathcal{L}_T}(f, \varphi) \tag{9.6}$$
$$\Omega_{S+T}(f, \varphi) := L_{\mathcal{L}_S}(f, \varphi) + L_{\mathcal{L}_T}(f, \varphi) \tag{9.7}$$

noting $L_{\mathcal{L}}(f, \varphi)$ the empirical cross-entropy of $f\varphi$ computed on some labelled dataset $\mathcal{L}$. Note that Ω_{S+T} gives more importance to target labelled samples compared to $\Omega_{S \cup T}$, especially in the small budget regime (*i.e.* when the budget b is such that $b \ll |\mathcal{L}_S|$). As a strong baseline exists in SSDA, we design Ω following the minimax entropy (MME) (Saito et al. 2019). Noting $H_{\mathcal{U}_T}(h) := -\frac{1}{|\mathcal{U}_T|} \sum_{x \in \mathcal{U}_T} h(x) \cdot \log h(x)$, the entropy of unlabelled samples $\mathcal{U}_T$, the MME objective is:

$$\begin{cases} \Omega_{\text{MME}}(f) & := \Omega_{S+T}(f, \varphi) - \lambda H_{\mathcal{U}_T}(f\varphi) \\ \Omega_{\text{MME}}(\varphi) & := \Omega_{S+T}(f, \varphi) + \lambda H_{\mathcal{U}_T}(f\varphi) \end{cases} \tag{9.8}$$

where $f := \sigma\left(\frac{1}{T} W \circ \ell_2\right)$ ($\ell_2(\mathbf{f}) := \mathbf{f}/||\mathbf{f}||_2$ is the L^2 normalization of features and $W \in \mathbb{R}^{C \times m}$ is a linear layer), $\lambda = 0.1$, $T = 0.05$ and σ is the softmax layer.

Training procedure

The training procedure is described in Algorithm 15. First, we train the model by UDA following the training procedure from (Bouvier et al. 2020b). Second, for a given number of iterations, we select by `Sage` (See Algorithm 4) b samples to send to the Oracle. Then, we perform UDA provided with the knowledge of newly labelled samples, that is using a SSDA regularizer Ω combined with soft-class conditioning loss L_{TSF}. We describe the gradient descent step in the following. First, given a loss L, Given a SSDA regularizer Ω (See Section 9.2.5), the gradient descent step is defined as follows, for some $\alpha > 0$:

$$(f, \varphi, \mathbf{d}) \leftarrow (f, \varphi, \mathbf{d}) - \alpha \nabla_{(f, \varphi, \mathbf{d})} \left(\hat{\Omega}(f, \varphi) + \lambda \hat{L}_{\text{TSF}}(f, \varphi)\right) \tag{9.9}$$

where for a given loss L, we note its batch-wise computation $\hat{L}$ when provided with batches of source labelled samples $\mathcal{B}_S^\ell$ from $\mathcal{L}_S$, a source labelled samples $\mathcal{B}_T^\ell$ from $\mathcal{L}_T$, a source labelled samples $\mathcal{B}_T^u$ from $\mathcal{U}_T$. Notably, $\mathcal{B}_S^\ell$ and $\mathcal{B}_T^\ell$ are involved for computing $\hat{\Omega}$ (eventually $\mathcal{B}_T^u$ for $\hat{\Omega}_{\text{MME}}$) while $\mathcal{B}_S^\ell$ and $\mathcal{B}_T^u$ are involved for computing $\hat{L}_{\text{TSF}}$.

Algorithm 5 Training procedure

Input: Labelled source samples $\mathcal{L}_S$, Unlabelled target samples $\mathcal{U}_T$, budget b, annotation rounds r, iterations n_{it}, SSDA regularizer Ω:

$\mathcal{L}_T \leftarrow \{\}, \mathcal{U}'_T \leftarrow \mathcal{U}_T$ ▷ Initializes the labelled target samples.
$f, \varphi, \mathbf{d} \leftarrow$ UDA as described in (**bouvier2020robust**) ▷ Pretraining before Active Learning.
for b rounds of annotations **do**
  $\mathcal{A} \leftarrow \texttt{Sage}(\mathcal{U}'_T, b, f, \varphi, \mathbf{d})$ ▷ Selects samples for annotation.
  $\mathcal{L} \leftarrow \text{Oracle}(\mathcal{A})$ ▷ Sends samples to an Oracle.
  $\mathcal{L}_T \leftarrow \mathcal{L}_T \cup \mathcal{L}$ ▷ Adds newly labelled samples.
  $\mathcal{U}'_T \leftarrow \mathcal{U}'_T \backslash \mathcal{A}$ ▷ Removes newly labelled samples.
  for n_{it} iterations **do**
    Sample a source labelled batch $\mathcal{B}_S^\ell$ from $\mathcal{L}_S$
    Sample a source labelled batch $\mathcal{B}_T^\ell$ from $\mathcal{L}_T$
    Sample a source labelled batch $\mathcal{B}_T^u$ from $\mathcal{U}_T$ ▷ (Not from $\mathcal{U}'_T$).
    $f, \varphi, \mathbf{d} \leftarrow$ Gradient descent update from Equation 9.9.
  end for
end for
Return: f, φ

9.3 Theoretical Analysis

9.3.1 Setup

In this section, we provide a simple example where the bound from Proposition 9.3.1 presented in Chapter 6 has a closed form. To conduct the analysis, we consider $\mathcal{X}$ as a measurable set provided with a probability measure noted p_T. We present an extension of an annotation selection to a measurable set. Selecting samples for annotation with probability budget $\tilde{b} \in [0, 1]$[1] consists in determining some measurable subset $\mathcal{B}$ such that $p_T(X \in \mathcal{B}) = \tilde{b}$. In the particular case where $p_T := \sum_{x \in \mathcal{D}_T} \delta_x$ (δ_x is the Dirac distribution in x) is an empirical distribution, determining some measurable subset $\mathcal{B}$ such that $p_T(X \in \mathcal{B}) = \tilde{b}$ consists in determining a subset of b samples of $\mathcal{D}_T$.

9.3.2 Naive Active Classifier

Given a classifier h and an annotated subset $\mathcal{B}$ (with probability b), we suggest a slight modification of the classifier h based on the annotation provided by the Oracle of $\mathcal{B}$. To this purpose, we introduce the *naive active classifier*, noted $h_{\mathcal{B}}(x)$, and defined as follows:

$$h_{\mathcal{B}}(x) = \text{Oracle}(x) \text{ if } x \in \mathcal{B}, h(x) \text{ otherwise.} \tag{9.10}$$

[1] The probability budget $\tilde{b}$ is related to the standard definition of budget b as $\tilde{b} = \frac{b}{|\mathcal{D}_T|}$.

Thus, $h_{\mathcal{B}}(x)$ returns the classifier's output $h(x)$ if x is not annotated and returns the oracle's output Oracle(x) if x is annotated.

9.3.3 A closed bound

We want to exhibit a closed form of ρ from Proposition 9.3.1 when considering the active classifier. To this purpose, we introduce the *purity* π of $\mathcal{B}$, $\pi := p_T(h_S(X) \neq \text{Oracle}(X) | X \in \mathcal{B})$. It reflects our capacity to identify misclassified target samples. With this notion, we observe that the naive classifier improves the target error; $\text{Err}_T(h_{\mathcal{B}}) \leq \text{Err}_T(h_S) - b\pi$. Put simply, the error is reduced by $b\pi$ corresponding to annotated samples for which the prediction is different from the Oracle output. The higher the budget of annotation b and the higher the purity π, the lower the target error of the naive classifier. It corresponds to $\text{Err}_T(h_S) - \tilde{b}\pi = \left(1 - \frac{\tilde{b}\pi}{\text{Err}_S(h_S)}\right) \text{Err}_S(h_S) \leq (1 - \tilde{b}\pi)\text{Err}_S(h_S)$; resulting into $\beta = (1 - \tilde{b}\pi)$. Note that $\text{Err}_T(h_S) - \tilde{b}\pi \geq 0$[2]

Proposition 9.3.1 (Guarantee of Adaptation in presence of target labelled data)**.** *Let* $\varphi \in \Phi$*;*

$$\text{Err}_T(\tilde{h}^{\varphi}) \leq \left(\frac{1}{\tilde{b}\pi} - 1\right) \left(\text{Err}_S(g_S\varphi) + 3C \cdot \text{INV}(\varphi) + \widehat{\text{TSF}}(\varphi, h_{\mathcal{B}}) + \text{Err}_T(g_T\varphi)\right) \tag{9.11}$$

The target error of the active classifier is a decreasing function of both the purity and the annotation budget and an increasing function of the transferability error. The budget b, the purity π and the transferability of representations τ are levers to improve the naive classifier target error. The budget b must be considered as a cost constraint and not as a parameter to be optimized. The purity of π is not tractable since it involves labels in the target domain. Some proxy measures, such as the entropy of predictions (Grandvalet and Bengio 2005), can provide a fair estimation of purity. However, it is known that deep nets tend to be overconfident on misclassified samples (Corbière et al. 2019). Therefore, we focus our efforts on understanding the role of active annotation in improving transferability error $\widehat{\text{TSF}}(\varphi, h_{\mathcal{B}})$.

9.4 Experiments

9.4.1 Setup

Tasks. We evaluate our approach on **Office-31** (Saenko et al. 2010), **VisDA-2017** (Peng et al. 2017) and **DomainNet** (Peng et al. 2019). Office-31 contains 4,652 images classified in 31 categories across three domains: Amazon (**A**), Webcam (**W**), and DSLR (**D**). We explore tasks **A** $\rightarrow$ **W**, **W** $\rightarrow$ **A**, **A** $\rightarrow$ **D** and **D** $\rightarrow$ **A**. We do not report results for tasks **D** $\rightarrow$ **W** and **W** $\rightarrow$ **D** since these tasks have already nearly perfect results in UDA (Long et al. 2018). For VisDA, we explore **Synthetic**: 3D models with different lighting conditions and different angles; **Real**: real-world images. We explore the **Synthetic** $\rightarrow$ **Real** task. **DomainNet** (Peng et al. 2019) is a large scale dataset with six domains and 345 classes (Clipart (**C**), Infograph (**I**), Painting (**P**), Quickdraw (**Q**), Real (**R**) and Sketch (**S**)). As **DomainNet** suffers of noisy labels, thus

[2] $\tilde{b}\pi = p_T(X \in \mathcal{B}) p_T(h_S(X) \neq \text{Oracle}(X) | X \in \mathcal{B}) = p_T(h_S(X) \neq \text{Oracle}(X), X \in \mathcal{B})$ while $\text{Err}_T(h_S) = p_T(h_S(X) \neq \text{Oracle}(X)$ and we use that for any events $\mathcal{A}, \mathcal{B}$ and for some probability $\mathbb{P}$, we have $\mathbb{P}(\mathcal{A} \cap \mathcal{B}) \leq \mathbb{P}(\mathcal{A})$.

violates the assumption of a perfect Oracle, we focus on the subset of 126 classes and the 7 tasks **R**→**C**, **R**→**P**, **P**→**C**, **C**→**S**, **S**→**P**, **R**→**S** and **P**→**R** (Saito et al. 2019).

Protocol. The standard protocol in UDA uses the same target samples during train and test phases. In AL's context, this induces an undesirable effect where sample annotation mechanically increases the accuracy. At train time, the model has access to input and label of annotated samples which are also present at test time. We suggest instead to split the target domain into a *train target domain* (samples used for adaptation and pool of data used for annotation) and *test target domain* (samples used for evaluating the model) with a ratio of 1/2. Therefore, samples from the test target domain have never been seen at train time. As a result, our protocol evaluates the model generalization in an inductive scenario. Reported results are based on 8 seeds for each method.

Budget, rounds and backbone. As the selected datasets are of different volumetry and difficulty, we used different budgets b: $b = 8$ for A→W and A→D (referred to as *easy tasks*), $b = 16$ for W→A and D→A (referred to as *medium tasks*), both $b = 16$ and $b = 128$ for VisDA (referred to as *hard tasks*). This allows to appreciate versatility of methods in *small* ($b = 8$), *medium* ($b = 16$) and *high* ($b = 128$) budget regimes. Our experiments are conducted with a 10 rounds of annotation for these tasks. Additional details for DomainNet experiments are provided in comparison with SSDA in Section 9.4.2. Our backbone is a ResNet50 (He et al. 2016) trained by 10k steps of SGD by UDA before annotation. We use DANN (Ganin and Lempitsky 2015) for AADA, MME (Saito et al. 2019) for MME based methods and TSF (**bouvier2020robust**) for TSF based methods.

Baselines **AADA** (Su et al. 2020) is the closest algorithm to Sage. AADA adapts representations by fooling a domain discriminator d trained to output 1 for source data and 0 for target data (Ganin and Lempitsky 2015) and scores target samples x; $s(x) := H(\hat{y})w(z)$ where $H(\hat{y})$ is the entropy of predictions $\hat{y}$ and $w(z) = (1 - d(z))/d(z)$. $H(\hat{y})$ brings information about uncertainty while $w(z)$ brings diversity to the score. We have reproduced the implementation of AADA. To demonstrate the effectiveness of Sage for Active DA, we report TSF with **Badge** query (Ash et al. 2020b) (**TSF+Badge**), which is the state-of-the-art query in AL. For these methods, we used $\Omega = \Omega_{S+T}$. To compare Sage with an AL method which ignores domain shift between labelled samples and queried samples, we report **Badge** with $\Omega_{S\cup T}$. Finally, to compare with SSDA approaches, we build two methods upon **MME** (Saito et al. 2019) with **Entropy** query (selection samples with highest prediction entropy (Wang and Shang 2014)), noted **MME+Entropy**, which is the most natural query for MME since it relies on max/min entropy, and with **Random** query noted **MME+Random**. We have reproduced the implementation of MME.

9.4.2 Results

Comparison with SOTA. Results are reported in Figure 9.4. First, active annotation brings substantial improvements to UDA (round 0 of annotation). This validates the effort and the focus that should be put on ADA, in our opinion. Sage outperforms the current state-of-the-art (AADA) with a comfortable margin for tasks with medium or hard difficulty, except for tasks A→D after the 5-th round. Importantly, Sage performs similarly or better than naively combining TSF with a state-of-the-art

query in AL (Badge) demonstrating that Sage takes into account the problem of domain shift in the query process. Finally, using a direct AL method (Badge) fails in the context of domain shift.

Ablation of Sage. We ablate the core components of Sage *i.e.* POP and the `k-means++` in Figures 9.5(a) and 9.5(b). Interestingly, Sage without POP fails to improve performances in the target domain. This demonstrates that POP brings information about uncertainty into the embedding. Sage without diversity performs poorly on VisDA($b = 128$), demonstrating that `k-means++` based sampling brings diversity. Diversity on Sage has a small effect on W→A.

Ablation of queries. We ablate in Figures 9.5(c) and 9.5(d) more AL strategies : (**Random**), where target samples are selected at random, (**Clusters**) that selects the closest samples to b clusters of representations obtained with k-means, (**Entropy**) based on the highest entropy $\max_{x \in \mathcal{U}_T} -h(x) \cdot \log h(x)$ (Wang and Shang 2014) and (**Confidence**), that used the smallest confidence $\min_{x \in \mathcal{U}_T} \max_c h(x)_c$ (Wang and Shang 2014) (**Confidence**). Sage is compared with a wide spectrum of AL queries based on representative (Random), diversity (Clusters) and uncertainty sampling (Entropy, Confidence). Sage outperforms them substantially on the two tasks, demonstrating it is well-suited for ADA.

Ablation of Ω. We report **TSF+$\Omega_{S \cup T}$** and **TSF+Ω_{MME}** which consists in adding MME as a regularization of TSF *i.e.* Ω used here is $\Omega_{S \cup T}$ and Ω_{MME}, respectively. Results are reported in Figures 9.5(e) and 9.5(f). We observe that using Ω_{S+T} and Ω_{MME} improve consistently wrt $\Omega_{S \cup T}$ on VisDA($b = 128$) while performing similarly on W→A. Furthermore, we observe that adding MME to TSF+Sage achieves the best performances on VisDA($b = 128$). Importantly, MME+Entropy is already strong for VisDA($b = 128$) explaining the substantial improvement when adding MME to TSF for this task.

ADA vs SSDA: ADA is a more realistic setting. We compare SSDA (a fixed number of labelled target samples per class are available, we refer to k shot the setting where k labelled target samples are available per class) with ADA (an Oracle provides ground-truth for queried target samples) when the number of target labelled samples are equal. Crucially, enforcing a fix number of labelled samples per class is unrealistic in practice. We report performances on DomainNet of MME (1 and 3 shot) (Saito et al. 2019) and Sage (here we used TSF + Sage + Ω_{MME}). AL is performed during 6 rounds with $b = 21$ and $b = 63$ for 1 and 3 shot respectively, leading to the same number of target labelled samples[3]. Results are presented in Table 9.2. In the 3-shot scenario Sage improves upon MME on all the tasks, except P→R. In the 1-shot scenario, Sage and MME perform similarly. This demonstrates that active annotation with Sage performs equally, or better, than MME, and benefits from more realistic assumptions.

9.5 Conclusion

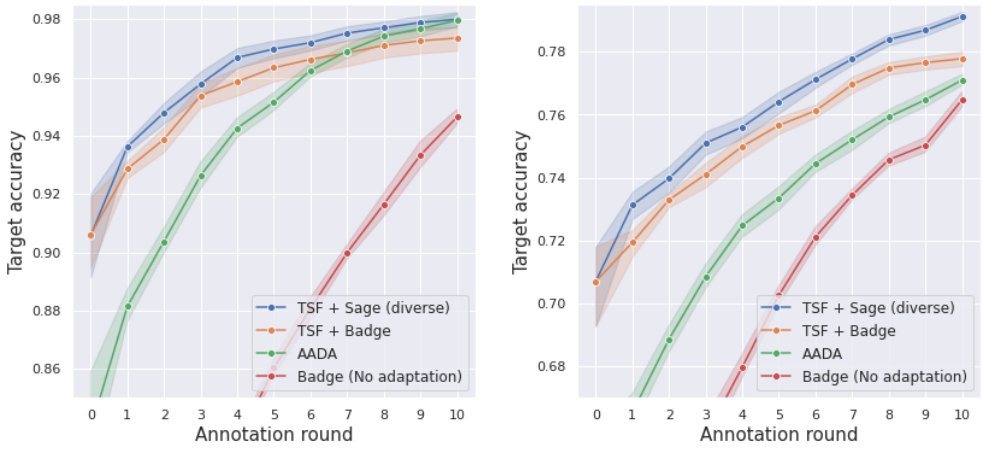
We have introduced Sage, an efficient method for ADA which identifies target samples that are likely to improve representations' transferability when annotated. It

[3] $|\mathcal{L}_T| = 21 \times 6 = 126$ (1 shot) and $|\mathcal{L}_T| = 63 \times 6 = 3 \times 126$ (3 shot)

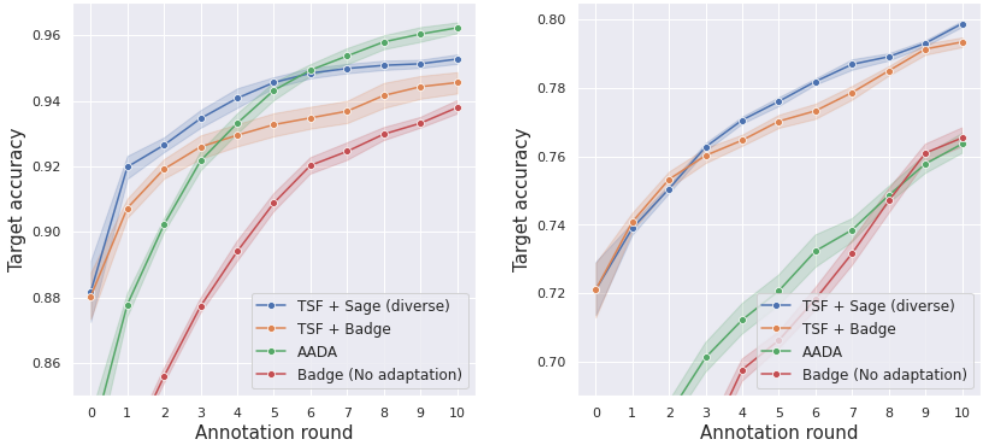
Tasks	1-shot			3-shot		
	MME	AADA	Sage	MME	AADA	Sage
R→C	67.5	64.4	**69.3**	70.1	68.8	**73.9**
R→P	**69.6**	65.5	69.4	70.8	67.0	**71.4**
P→C	69.0	63.2	**69.9**	71.4	67.3	**74.1**
C→S	**62.2**	57.4	61.5	64.7	60.1	**65.4**
S→P	**67.9**	62.6	**67.9**	69.6	64.9	**69.8**
R→S	61.2	57.0	**62.1**	63.6	59.9	**65.8**
P→R	**79.3**	74.9	79.0	80.9	76.9	**81.2**
Mean	68.1	63.6	**68.5**	70.2	66.3	**71.7**

Table 9.2: SSDA (MME) vs ADA (AADA and Sage) on DomainNet. MME's results deviate from (Saito et al. 2019) due to train/test split, ResNet50 as backbone and minor implementation changes.

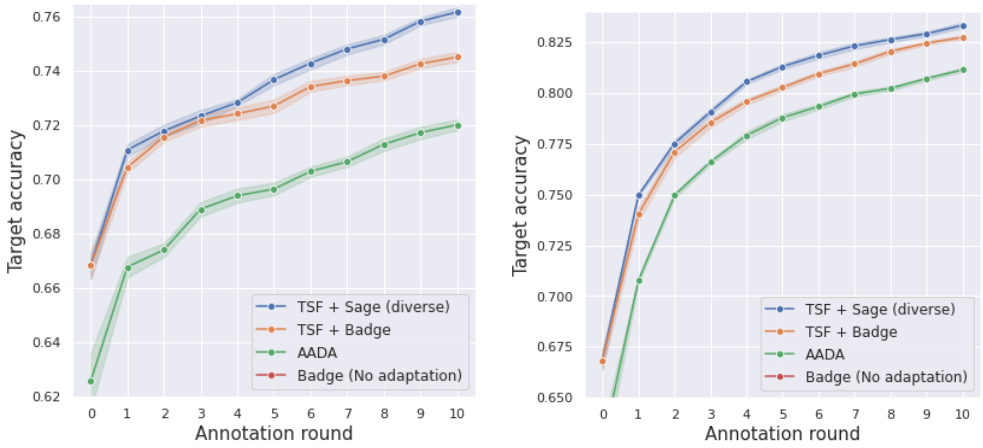
relies on two core components; a stochastic embedding of the gradient of the transferability loss and a `k-means++` initialization, which guarantees that each annotation round annotates a diverse set of target samples. Through various experiments, we have demonstrated the effectiveness of Sage and its capacity to take the best of uncertainty, representative, and diversity sampling. New SSDA strategies when using Sage is an interesting direction for future works.



(a) A→W (b) W→A

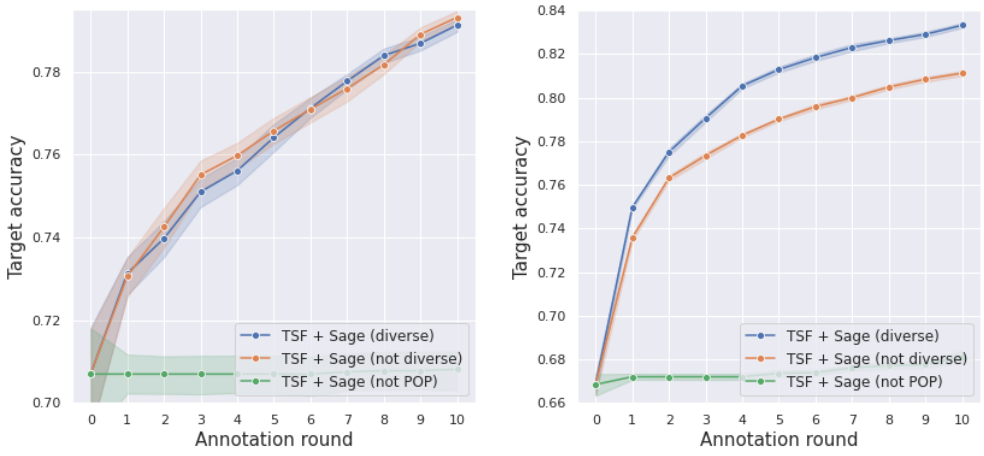


(c) A→D (d) D→A

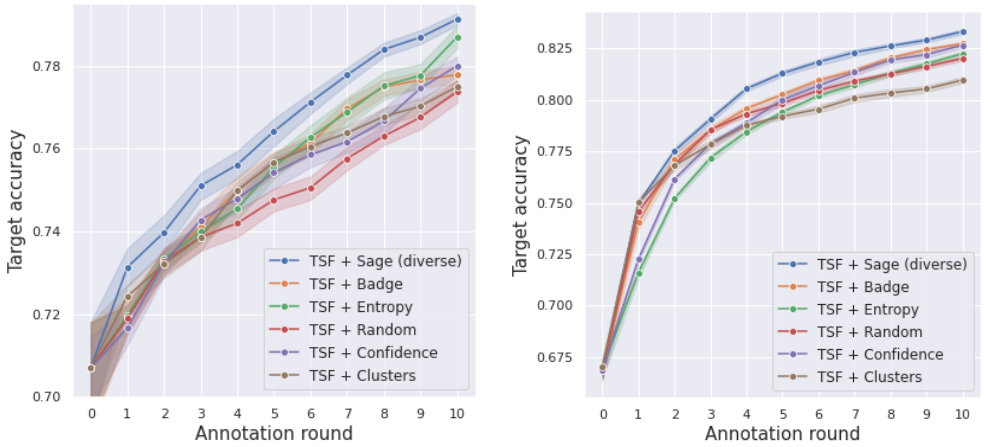


(e) VisDA($b = 16$) (f) VisDA($b = 128$)

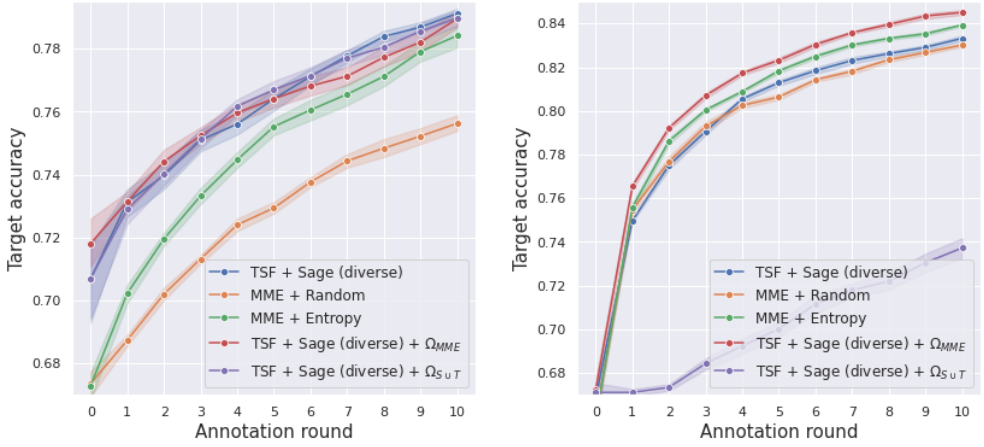
Figure 9.4: Annotation of target samples improves adaptation drastically for the considered tasks. TSF+Sage (in blue) improves upon the state-of-the-art of ADA (AADA, in green), except for task A→D. AL (Badge, in red) performs poorly in this context (Badge without adaptation does not appear on VisDA tasks since it performs poorly: 47.0% and 63.4% after 10 rounds of annotation for $b = 16$ and $b = 128$, respectively) showing the importance of addressing the problem of adaptation for AL under distribution shift. Naively combining Badge with TSF (TSF+Badge, in orange) performs worsen than Sage. Sage takes into account the problem of domain shift when querying samples.



(a) W→A (b) VisDA($b = 128$)



(c) W→A (d) VisDA($b = 128$)



(e) W→A (f) VisDA($b = 128$)

Figure 9.5: (a) and (b): Both the POP and `k-means++` are crucial components for the empirical success of Sage. (c) and (d): Sage outperforms AL query based on representative, diversity and uncertainty samplings. (e) and (f): Effect of adding MME to TSF+Sage.

Part IV

Adaptation in the Real-World: Towards Adaptive Models

10 Bridging Adaptation and Few-Shot Learning

Contents

Part II and Part III focus on improving invariant representations' transferability for the *Unsupervised Domain Adaptation* (UDA) problem, from the theoretical point of view and the applications, respectively. Part IV opens novel questions towards more realistic applications of adaptation.

The previous contributions followed the classical assumptions of the paradigm of UDA, namely the access to a labelled source domain and an unlabeled target domain both populated with abundant data (see Definition 3.2.2). Thus, the well-studied sampling complexity term governs this regime, that we have introduced when developing the principle of *Empirical Risk Minimization* (ERM) from Section 3.1.2 (Theorem 3.2) the theory of learning from different domains from Section 3.2 (Theorem 3.3).

The present Chapter aims to confront such common assumptions to realistic case settings. We address the challenging problem of adapting in the small data regime, *i.e.* not assuming anymore abundant data populates both the source and the target domains. To provide a formal description of this novel problem, we rely on the well-established field of *Few-Shot Learning* (FSL) (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Finn, Abbeel, and Levine 2017). Few-Shot Learning classifies at test-time *query instances* from novel classes, *i.e.* not seen at train-time, by only requiring *support set* composed of a few labelled samples. We deviate from the standard setup of FSL by considering the case where the support set, *i.e.* labelled samples, and the

query set, *i.e.* unlabelled samples, are sampled from different distributions, *i.e.* the source and the target distributions, respectively. We refer to this new problem as ***F**ew-Shot Learning under **S**upport/**Q**uery **S**hift* (**FSQS**).

Chapter 10 is organized as follows;

1. We provide a formal statement of FSQS, and we position this new problem among existing learning paradigms.
2. We introduce `FewShiftBed`, a testbed for FSQS[1]. The testbed includes 3 challenging benchmarks along with a protocol for fair and rigorous comparison across methods as well as an implementation of relevant baselines, and an interface to facilitate the implementation of new methods.
3. We conduct extensive experimentation of a representative set of few-shot algorithms. We empirically show that *Transductive* Batch-Normalization (Bronskill et al. 2020) mitigates an important part of the inopportune effect of FSQS.
4. We bridge *Unsupervised Domain Adaptation* (UDA) with FSL to address FSQS. We introduce *Transported Prototypes* (TP), an efficient transductive algorithm that couples *Optimal Transport* (OT) (Peyré, Cuturi, et al. 2019) with the celebrated *Prototypical Networks* (Snell, Swersky, and Zemel 2017). The use of OT follows a long-standing history in UDA for aligning distributions (Courty et al. 2016). Our experiments demonstrate that OT shows a remarkable ability to perform this alignment even with only a few samples to compare distributions and provide a simple but strong baseline.

Chapter 10 is based on the publication and an oral presentation in an international conference;

> Bridging Few-Shot Learning and Adaptation: New Challenges of Support-Query Shift,
> *Étienne Bennequin, Victor Bouvier, Myriam Tami, Antoine Toubhans and Céline Hudelot*,
> European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Bilbao (Basque Country), Online, 2021.

Étienne Bennequin and Victor Bouvier contributed equally.

[1] `https://github.com/ebennequin/meta-domain-shift`

10.1 The Support Query Shift Problem

10.1.1 Motivations

In the last few years, we have witnessed outstanding progress in supervised deep learning (He et al. 2016). As the abundance of labelled data during training is rarely encountered in practice, ground-breaking works in *Few-Shot Learning* (FSL) have emerged (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Finn, Abbeel, and Levine 2017), particularly for image classification. This paradigm relies on a straightforward setting. At test-time, given a set of *few* (typically 1 to 5) labelled examples (not seen during training) for each of those classes, the task is to classify query samples among them. We usually call the set of labelled samples the *support set*, and the set of query samples the *query set*. Well-adopted FSL benchmarks (Vinyals et al. 2016; Ren et al. 2018; Triantafillou et al. 2019) commonly sample the support and query sets from the same distribution. We stress that this assumption does not hold in most use cases. When deployed in the real-world, we expect an algorithm to infer on data that may shift, resulting in an acquisition system that deteriorates, lighting conditions that vary, or real world objects evolving (Amodei et al. 2016). As presented in Chapter 2 (see the Section 3.2), the situation of *Distribution Shift* (DS), *i.e.* when training and testing distributions differ, is ubiquitous and has dramatic effects on deep models.

The state of the art in FSL brings insufficient knowledge on few-shot learners' behaviours when facing distribution shift. Some pioneering works demonstrate that advanced FSL algorithms do not handle cross-domain generalization better than more naive approaches (Chen et al. 2019a). Despite its great practical interest, FSL under distribution shift between the support and query set is an under-investigated problem and attracts a very recent attention (Du et al. 2021). We refer to it as ***F**ew-Shot Learning under **S**upport/**Q**uery **S**hift* (**FSQS**) and provide an illustration in Figure 10.1. We will detail formally the problem of FSQS in Section 10.1.3. For brevity, we refer to Support-Query Shift as SQS. It reflects a more realistic situation where the algorithm is fed with a support set at the time of deployment and infers continuously on data subject to shift. We aim to design an algorithm that is robust to the distribution shift encountered during inference. This is the subject of the present chapter.

(a) Standard FSL (b) FSL under Support / Query Shift

Figure 10.1: Illustration of the FSQS problem with a 5-way 1-shot classification task sampled from the miniImageNet dataset (Vinyals et al. 2016). In (a), a standard FSL setting where support and query sets are sampled from the same distribution. In (b), the same task but with shot-noise and contrast perturbations from (Hendrycks and Dietterich 2019b) applied on support and query sets (respectively) that results in a support-query shift. In the latter case, a similarity measure based on the Euclidean metric (Snell, Swersky, and Zemel 2017) may become inadequate.

	SQ problems	Train-Time				Test-Time				
		Support		Query		Support		Query	New	New
		Size	Labels	Size	Labels	Size	Labels	Transductivity	classes	domains
No SQS	FSL	Few	✓	Few	✓	Few	✓	Point-wise	✓	
	TransFSL	Few	✓	Few	✓	Few	✓	Small	✓	
	CDFSL	Few	✓	Few	✓	Few	✓	Point-wise	✓	✓
SQS	UDA					Large	✓	Large		
	TTA	Large	✓					Small		✓
	ARM	Large	✓	Few	✓			Small		✓
	Ind FSQS	Few	✓	Few	✓	Few	✓	Point-wise	✓	✓
	Trans FSQS	Few	✓	Few	✓	Few	✓	Small	✓	✓

Table 10.1: An overview of SQ problems including FSL (Snell, Swersky, and Zemel 2017; Finn, Abbeel, and Levine 2017), TransFSL (Ren et al. 2018; Liu et al. 2019b), CDFSL (Chen et al. 2019a), UDA (Quinonero-Candela et al. 2009; Pan and Yang 2009), TTA (Sun et al. 2020; Schneider et al. 2020a; Wang et al. 2021b), ARM (Zhang et al. 2021b). We divide SQ problems into two categories, presence or not of **S**upport-**Q**uery shift; **No SQS** *vs* **SQS**. We consider three classes of transductivity: point-wise transductivity that is equivalent to inductive inference, small transductivity when inference is performed at batch level (typically in (Wang et al. 2021b; Zhang et al. 2021b)), and large transductivity when inference is performed at dataset level (typically in UDA). New classes (resp. new domains) describe if the model is evaluated at test-time on novel classes (resp. novel domains). Note that we frame UDA as a fully test-time algorithm. Notably, Cross-Domain FSL (CDFSL) (Chen et al. 2019a) assumes that the support set and query set are drawn from the same distribution, thus No SQS.

10.1.2 Positioning and Related Works

To highlight FSQS's novelty, our discussion revolves around the problem of inferring on a given *Query Set* provided with the knowledge of a *Support Set*. We refer to this class of problems as *Support-Query problems* (SQ problems). Intrinsically, FSL falls into the category of SQ problems. Interestingly, *Unsupervised Domain Adaptation* (Pan and Yang 2009) (UDA), defined as labelling a dataset sampled from a target domain based on labelled data sampled from a source domain, is also a SQ problem. Indeed, in this case, the source domain plays the role of support, while the target domain plays the query's role.

Transductive algorithms also have a special place in FSL (Dhillon et al. 2020; Liu et al. 2019b; Ren et al. 2018) and show that leveraging a query set as a whole brings a significant boost in performances. For more details about the role of transductivity in Machine Learning, we refer to the Section 3.1.4. Nevertheless, UDA and FSL exhibit fundamental differences. UDA addresses the problem of distribution shift using important source data and target data (typically thousands of instances) to align distributions. In contrast, FSL focuses on the difficulty of learning from few samples. To this purpose, we frame UDA as both SQ problem with *large* transductivity and Support / Query Shift, while Few-Shot Learning is a SQ problem, eventually with *small* transductivity for transductive FSL. Thus, FSQS combines both challenges: distribution shift and small transductivity. This new perspective allows us to establish fruitful connections with related learning paradigms, presented in Table 10.1, that we review in the following.

Adaptation. UDA requires a whole target dataset for inference, limiting its applications. Recent pioneering works, referred to as Test-Time Adaptation (TTA), adapt at test-time a model provided with a batch of samples from the target distribution. The proposed methodologies are test-time training by self-supervision (Sun et al.

2020), updating batch-normalization statistics (Schneider et al. 2020a) or parameters (Wang et al. 2021b), or meta-learning to condition predictions on the whole batch of test samples for an *Adaptative Risk Minimization* (ARM) (Zhang et al. 2021b). Inspired from the principle of invariant representations (Ben-David et al. 2007; Ganin and Lempitsky 2015), the seminal work (Courty et al. 2016) brings *Optimal Transport* (OT) (Peyré, Cuturi, et al. 2019) as an efficient framework for aligning data distributions. OT has been recently applied in a context of transductive FSL (Hu, Gripon, and Pateux 2020) and our proposal (TP) is to provide a simple and strong baseline following the principle of OT as it is applied in UDA. In this work, following (Bronskill et al. 2020), we also study the role of Batch-Normalization for SQS, that points out the role of transductivity. Our conviction was that the batch-normalization is the first lever for aligning distributions (Schneider et al. 2020a; Wang et al. 2021b).

Few-Shot Classification. We usually frame Few-Shot Classification methods (Chen et al. 2019a) as either metric-based methods (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017), or optimization-based methods that learn to fine-tune by adapting with few gradient steps (Finn, Abbeel, and Levine 2017). A promising line of study leverages *transductivity* (using the query set as unlabelled data while inductive methods predict individually on each query sample). Transductive Propagation Network (Liu et al. 2019b) meta-learns label propagation from the support to query set concurrently with the feature extractor. Transductive Fine-Tuning (Dhillon et al. 2020) minimizes the prediction entropy of all query instances during fine-tuning. Evaluating cross-domain generalization of FSL (FSCD), *i.e.* a distributional shift between meta-training and meta-testing, attracts the attention of a few recent works (Chen et al. 2019a). Zhao *et al.* propose a Domain-Adversarial Prototypical Network (Zhao et al. 2020) in order to both align source and target domains in the feature space while maintaining discriminativeness between classes. Sahoo *et al.* combine Prototypical Networks with adversarial domain adaptation at the task level (Sahoo et al. 2019). Notably, Cross-Domain Few-Shot Learning (Chen et al. 2019a) (CDFSL) addresses the distributional shift between meta-training and meta-testing assuming that the support set and query set are drawn from the same distribution, not making it a SQ problem with support-query shift. Concerning the novelty of FSQS, we acknowledge the very recent contribution of Du *et al.* (Du et al. 2021) which studies the role of learnable normalization for domain generalization, in particular when support and query sets are sampled from different domains. Note that our statement is more ambitious: we evaluate algorithms on both source and target domains that were unseen during training, while in their setting the source domain has already been seen during training.

Benchmarks in Machine Learning Releasing benchmark has always been an important factor for progress in the *Machine Learning* field, the most outstanding example being ImageNet (Deng et al. 2009) for the Computer Vision community. Recently, `DomainBed` (Gulrajani and Lopez-Paz 2021) aims to settle Domain Generalization research into a more rigorous process, where `FewShiftBed` takes inspiration from it. `Meta-Dataset` (Triantafillou et al. 2019) is an other example, this time specific to FSL.

10.1.3 Statement

Notations. We consider an input space $\mathcal{X}$, a representation space $\mathcal{Z} \subset \mathbb{R}^m$ ($d > 0$) and a set of classes C. A representation is a learnable function from $\mathcal{X}$ to $\mathcal{Z}$ and is noted $\varphi(\cdot;\theta)$ with $\theta \in \Theta$ for Θ a set of parameters. A dataset is a set $\Delta(\mathsf{C},\mathsf{D})$

Figure 10.2: During meta-learning (Train-Time), each episode contains a support and a query set sampled from different distributions (for instance, illustrated by noise and contrasts as in Figure 10.1(b)) from a set of *training domains* ($\mathsf{D}_{\text{train}}$), reflecting a situation that may potentially occurs at test-time. When deployed, the FSL algorithm using a trained backbone is fed with a support set sampled from new classes. As the algorithm is subject to infer continuously on data subject to shift (Test-Time), we evaluate the algorithm on data with an unknown shift (D_{test}). Importantly, both classes ($\mathsf{C}_{\text{train}} \cap \mathsf{C}_{\text{test}} = \emptyset$) and shifts ($\mathsf{D}_{\text{train}} \cap \mathsf{D}_{\text{test}} = \emptyset$) are not seen during training, making the FSQS a challenging problem of generalization.

defined by a set of classes C and a set of domains D, *i.e.* a domain $\mathcal{D} \in \mathsf{D}$. Earlier in this thesis, we identified a domain with its associated distribution. As a notation abuse in the chapter, a domain $\mathcal{D}$ is a set of IID realizations from a distribution noted $p_{\mathcal{D}}$. The distribution shift between domains $\mathcal{D}$ and $\mathcal{D}'$ is characterized by $p_{\mathcal{D}} \neq p_{\mathcal{D}'}$. Referring to the well known UDA terminology of source / target, we define a couple of source-target domains as a couple $(\mathcal{D}_S, \mathcal{D}_T)$ with $p_{\mathcal{D}_S} \neq p_{\mathcal{D}_T}$, thus presenting a distribution shift. Additionally, given $\mathcal{C} \subset \mathsf{C}$ and $\mathcal{D} \in \mathsf{D}$, the restriction of a domain $\mathcal{D}$ to images with a label that belongs to $\mathcal{C}$ is noted $\mathcal{D}^{\mathcal{C}}$.

Dataset splits. We build a split of $\Delta(\mathsf{C}, \mathsf{D})$, by splitting D (respectively C) into $\mathsf{D}_{\text{train}}$ and D_{test} (respectively $\mathsf{C}_{\text{train}}$ and C_{test}) such that $\mathsf{D}_{\text{train}} \cap \mathsf{D}_{\text{test}} = \emptyset$ and $\mathsf{D}_{\text{train}} \cup \mathsf{D}_{\text{test}} = \mathsf{D}$ (respectively $\mathsf{C}_{\text{train}} \cap \mathsf{C}_{\text{test}} = \emptyset$ and $\mathsf{C}_{\text{train}} \cup \mathsf{C}_{\text{test}} = \mathsf{C}$). This gives us a train/test split with the datasets $\Delta_{\text{train}} = \Delta(\mathsf{C}_{\text{train}}, \mathsf{D}_{\text{train}})$ and $\Delta_{\text{test}} = \Delta(\mathsf{C}_{\text{test}}, \mathsf{D}_{\text{test}})$. By extension, we build a validation set following the same protocol.

Few-Shot Learning under Support-Query Shift (FSQS). Given:

- $\mathsf{D}' \in \{\mathsf{D}_{\text{train}}, \mathsf{D}_{\text{test}}\}$ and $\mathsf{C}' \in \{\mathsf{C}_{\text{train}}, \mathsf{C}_{\text{test}}\}$,
- a couple of source-target domains $(\mathcal{D}_S, \mathcal{D}_T)$ from D'[2],
- a set of classes $\mathcal{C} \subset \mathsf{C}'$;
- a small labelled support set $\mathcal{S} = (x_i, y_i)_{i=1,\dots,|\mathcal{S}|}$ (named *source support set*) such that for all i, $y_i \in \mathcal{C}$ and $x_i \in \mathcal{D}_S$ *i.e.* $\mathcal{S} \subset \mathcal{D}_S^{\mathcal{C}}$;
- an unlabelled query set $\mathcal{Q} = (x_i)_{i=1,\dots,|\mathcal{Q}|}$ (named *target query set*) such that for all i, $y_i \in \mathcal{C}$ and $x_i \in \mathcal{D}_T$ *i.e.* $\mathcal{Q} \subset \mathcal{D}_T^{\mathcal{C}}$.

The task is to predict the labels of query set instances in $\mathcal{C}$. When $|\mathcal{C}| = n$ and the support set contains k labelled instances for each class, this is called an n-way k-shot FSQS classification task. Note that this paradigm provides an additional challenge

[2]Note that we do not split either $\mathcal{D}_S$ or $\mathcal{D}_T$ in a train / test sets. Indeed, our evaluation is performed for couple $\mathcal{D}_S, \mathcal{D}_T \in \mathsf{D}_{\text{test}}$, *i.e.* that are not in $\mathsf{D}_{\text{train}}$.

compared to classical few-shot classification tasks, since at test time, the model is expected to generalize to both new classes and new domains while support set and query set are sampled from different distributions. Additionally, it differs from the setup of Cross-Domain Few-Shot Learning (CDFSL (Chen et al. 2019a)) since the latter evaluates the model on a new domain but there is no shift between the support and the query set. This paradigm is illustrated in Figure 10.2.

Episodic training. We build an episode by sampling some classes $\mathcal{C} \subset \mathsf{C}_{\text{train}}$, and a source and target domain $\mathcal{D}_S, \mathcal{D}_T$ from $\mathsf{D}_{\text{train}}$. We build a support set $\mathcal{S} = (x_i, y_i)_{i=1...|\mathcal{S}|}$ of instances from source domain $\mathcal{D}_S^{\mathcal{C}}$, and a query set $\mathcal{Q} = (x_i, y_i)_{i=|\mathcal{S}|+1,...,|\mathcal{S}|+|\mathcal{Q}|}$ of instances from target domain $\mathcal{D}_T^{\mathcal{C}}$, such that $\forall i \in [1, |\mathcal{S}| + |\mathcal{Q}|]$, $y_i \in \mathcal{C}$. Using the labelled examples from $\mathcal{S}$ and unlabelled instances from $\mathcal{Q}$, the model is expected to predict the labels of $\mathcal{Q}$. The parameters of the model are then trained using a cross-entropy loss between the predicted labels and ground truth labels of the query set.

10.2 FewShiftBed: A Pytorch testbed for FSQS

10.2.1 Datasets

From three existing datasets, we extended them to design three new image classification datasets for the FSQS problem. These datasets have two specificities:

1. They are dividable into groups of images, assuming that each group corresponds to a distinct domain. A key challenge is that each group must contain enough images with a sufficient variety of class labels, so that it is possible to sample FSQS episodes.
2. They are delivered with a train/val/test split ($\Delta_{\text{train}}, \Delta_{\text{val}}, \Delta_{\text{test}}$), along both the class and the domain axis. This split is performed following the principles detailed in Section 10.1.3. Therefore, these datasets provide true few-shot tasks at test time, in the sense that the model will not have seen any instances of test classes and domains during training. Note that since we split along two axes, some data may be discarded (for instance images from a domain in $\mathsf{D}_{\text{train}}$ with a label in C_{test}). Therefore it is crucial to find a split that minimizes this loss of data.

Meta-CIFAR100-Corrupted (MC100-C). The dataset CIFAR-100 (Krizhevsky, Hinton, et al. 2009) is composed of 60k three-channel square images of size 32×32, evenly distributed in 100 classes. Classes are evenly distributed in 20 superclasses. We use the same method used to build CIFAR-10-C (Hendrycks and Dietterich 2019b), which makes use of 19 image perturbations, each one being applied with 5 different levels of intensity, to evaluate the robustness of a model to domain shift. We modify their protocol to adapt it to the FSQS problem: (i) we split the classes with respect to the superclass structure, and assign 13 superclasses (65 classes) to the training set, 2 superclasses (10 classes) to the validation set, and 5 superclasses (25 classes) to the testing set; (ii) we also split image perturbations (acting as domains), following the split of (Zhang et al. 2021b). We obtain 2,184k transformed images for training, 114k for validation and 330k for testing. The detailed split is available in the documentation of our code repository.

***mini*ImageNet-Corrupted (mIN-C).** *mini*ImageNet (Vinyals et al. 2016) is a popular benchmark for few-shot image classification. It contains 60k images from 100 classes from the ImageNet dataset. 64 classes are assigned to the training set, 16 to the validation set and 20 to the test set. Like MC100-C, we build mIN-C using the image perturbations proposed by (Hendrycks and Dietterich 2019b) to simulate different domains. We use the original split from (Vinyals et al. 2016) for classes, and use the same domain split as for MC100-C. Although the original *mini*ImageNet uses 84×84 images, we use 224×224 images. This allows us to re-use the perturbation parameters calibrated in (Hendrycks and Dietterich 2019b) for ImageNet. Finally, we discard the 5 most time-consuming perturbations. We obtain a total of 1.2M transformed images for training, 182k for validation and 228k for testing. The detailed split in the documentation of our code repository.

FEMNIST-FewShot (FEMNIST-FS). EMNIST (Cohen et al. 2017) is a dataset of images of handwritten digits and uppercase and lowercase characters. Federated-EMNIST (Caldas et al. 2018) is a version of EMNIST where images are sorted by writer (or user). FEMNIST-FS consists in a split of the FEMNIST dataset adapted to few-shot classification. We separate both users and classes between training, validation and test sets. We build each group as the set of images written by one user. The detailed split is available in the code. Note that in FEMNIST, many users provide several instances for each digits, but less than two instance for most letters. Therefore it is hard to find enough samples from a user to build a support set or a query set. As a result, our experiments are limited to classification tasks with only one sample per class in both the support and query sets.

10.2.2 Algorithms

We implement in `FewShiftBed` two representative methods of the vast literature of FSL, that are commonly considered as strong baselines: Prototypical Networks (**ProtoNet**) (Snell, Swersky, and Zemel 2017) and Matching Networks (**MatchingNet**) (Vinyals et al. 2016). Besides, for transductive FSL, we also implement with Transductive Propagation Network (**TransPropNet**) (Liu et al. 2019b) and Transductive Fine-Tuning (**FTNet**) (Dhillon et al. 2020). We also implement our novel algorithm *Transported Prototypes* (**TP**) which is detailed in Section 10.3. `FewShiftBed` is designed for favoring a straightforward implementation of a new algorithm for FSQS. To add a new algorithm, we only need to implement the `set_forward` method of the class `AbstractMetaLearner`. We provide an example with our implementation of the Prototypical Network (Snell, Swersky, and Zemel 2017) that only requires few line of codes:

```
class ProtoNet(AbstractMetaLearner):
    def set_forward(self, support_images, support_labels, query_images):
        z_support, z_query = self.extract_features(support_images, query_images)
        z_proto = self.get_prototypes(z_support, support_labels)
        return  - euclidean_dist(z_query, z_proto)
```

10.2.3 Protocol

To prevent the pitfall of misinterpreting a performance boost, we draw three recommendations to isolate the causes of improvement rigorously.

- **How important is episodic training?** Despite its wide adoption in meta-learning for FSL, in some situation episodic training does not perform better than more

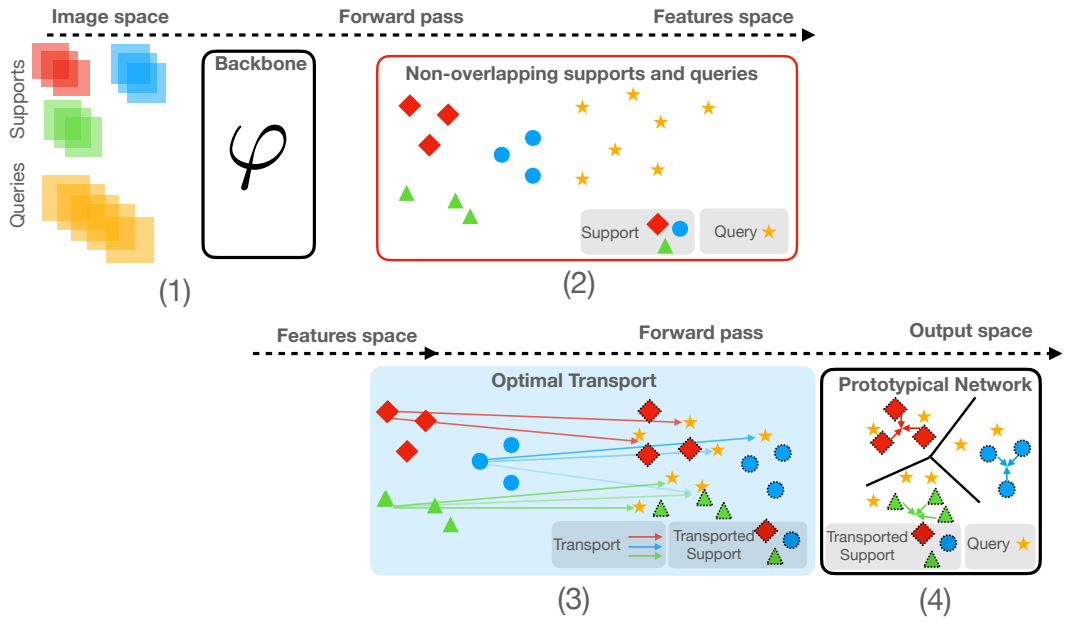


Figure 10.3: Overview of *Transported Prototypes*. (1) A support set and a query set are fed to a trained backbone that embeds images into a feature space. (2) Due to the shift between distributions, support and query instances are embedded in non-overlapping areas. (3) We compute the Optimal Transport from support instances to query instances to build the transported support set. Note that we represent the transport plan only for one instance per class to preserve clarity in the schema. (4) Provided with the transported support, we apply the Prototypical Network (Snell, Swersky, and Zemel 2017) *i.e.* L^2 similarity between transported support and query instances.

naive approaches (Chen et al. 2019a). Therefore we recommend to report both the result obtained using episodic training and standard ERM (see the documentation of our code repository).

- **How does the algorithm behave in the absence of Support-Query Shift?** In order to assess that an algorithm designed for distribution shift does not provide degraded performance in an ordinary concept, and to provide a top-performing baseline, we recommend reporting the model's performance when we do not observe, at test-time, a support-query shift. Note that it is equivalent to evaluate the performance in cross-domain generalization, as firstly described in (Chen et al. 2019a).

- **Is the algorithm transductive?** The assumption of transductivity has been responsible of several improvements in FSL (Ren et al. 2018; Bronskill et al. 2020) while it has been demonstrated in (Bronskill et al. 2020) that MAML (Finn, Abbeel, and Levine 2017) benefits strongly from the Transductive Batch-Normalization (TBN). Thus, we recommend specifying if the method is transductive and adapting the choice of the batch-normalization accordingly (Conventional Batch Normalization (Ioffe and Szegedy 2015a) and Transductive Batch Normalization for inductive and transductive methods, respectively) since transductive batch normalization brings a significant boost in performance (Bronskill et al. 2020).

10.3 Transported Prototypes: A baseline for FSQS

10.3.1 Overall idea

We present a novel method that brings UDA to FSQS. As aforementioned, FSQS presents new challenges since we no longer assume that we sample the support set and the query set from the same distribution. As a result, it is unlikely that the support set and query sets share the same representation space region (non-overlap). In particular, the L^2 distance, adopted in the celebrated Prototypical Network (Snell, Swersky, and Zemel 2017), may not be relevant for measuring similarity between query and support instances, as presented in Figure 10.1. To overcome this issue, we develop a two-phase approach that combines Optimal Transport (Transportation Phase) and the celebrated Prototypical Network (Prototype Phase). We give some background about Optimal Transport (OT) in Section 10.3.2 and the whole procedure is presented in Algorithm 6.

10.3.2 Background

Definition. We provide some basics about Optimal Transport (OT). A thorough presentation of OT is available at (Peyré, Cuturi, et al. 2019). Let p_S and p_T be two distributions on $\mathcal{X}$, we note $\Pi(p_S, p_T)$ the set of joint probability with marginal p_S and p_T *i.e.* $\forall \pi \in \Pi(p_S, p_T), \forall x \in \mathcal{X}, \pi(\cdot, x) = p_S, \pi(x, \cdot) = p_T$. The *Optimal Transport*, associated to cost c, between p_S and p_T is defined as:

$$W_c(p_S, p_T) := \min_{\pi \in \Pi(p_S, p_T)} \mathbb{E}_{(x_S, x_T) \sim \pi} \left[c(x_S, x_T) \right] \tag{10.1}$$

with $c(\cdot, \cdot)$ any metric. We note $\pi^\star(p_S, p_T)$ the joint distribution that achieves the minimum in equation 10.1. It is named the *transportation plan* from p_S to p_T. When there is no confusion, we simply note $\pi^\star$. For our applications, we will use as metric the euclidean distance in the representation space obtained from a representation $\varphi(\cdot; \theta)$ *i.e.* $c_\theta(x_S, x_T) := ||\varphi(x_S; \theta) - \varphi(x_T; \theta)||_2$.

Discrete OT. When p_S and p_T are only accessible through a finite set of samples, respectively $(x_{S,1}, ..., x_{S,n_S})$ and $(x_{T,1}, ..., x_{T,n_T})$ we introduce the empirical distributions $\hat{p}_S := \sum_{i=1}^{n_S} w_{S,i} \delta_{x_{S,i}}$, $\hat{p}_T := \sum_{j=1}^{n_T} w_{T,j} \delta_{x_{T,j}}$, where $w_{S,i}$ ($w_{T,j}$) is the mass probability put in sample $x_{S,i}$ ($x_{T,j}$) *i.e.* $\sum_{i=1}^{n_S} w_{S,i} = 1$ ($\sum_{j=1}^{n_T} w_{T,j} = 1$) and δ_x is the Dirac distribution in x. The discrete version of the OT is derived by introducing the set of couplings $\mathbf{\Pi}(p_S, p_T) := \left\{ \pi \in \mathbb{R}^{n_S \times n_T}, \pi \mathbf{1}_{n_S} = \mathbf{p}_S, \pi^\top \mathbf{1}_{n_T} = \mathbf{p}_T \right\}$ where $\mathbf{p}_S := (w_{S,1}, \cdots, w_{S,n_S})$, $\mathbf{p}_T := (w_{T,1}, \cdots, w_{1,n_T})$, and $\mathbf{1}_{n_S}$ (respectively $\mathbf{1}_{n_T}$) is the unit vector with dim n_S (respectively n_T). The discrete transportation plan $\pi_\theta^\star$ is then defined as:

$$\pi_\theta^\star := \underset{\pi \in \mathbf{\Pi}(p_S, p_T)}{\operatorname{argmin}} \langle \pi, \mathbf{C}_\theta \rangle_F \tag{10.2}$$

where $\mathbf{C}_\theta(i, j) := c_\theta(x_{S,i}, x_{T,j})$ and $\langle \cdot, \cdot \rangle_F$ is the Frobenius dot product. Note that $\pi_\theta^\star$ depends on both p_S and p_T, and θ since $\mathbf{C}_\theta$ depends on θ. In practice, we use Entropic regularization (Cuturi 2013) that makes OT easier to solve. Entropic regularization was proposed in (Cuturi 2013). It is defined as

$$\pi_\theta^\star := \underset{\pi \in \mathbf{\Pi}(p_S, p_T)}{\operatorname{argmin}} \langle \pi, \mathbf{C}_\theta \rangle_F + \varepsilon \cdot \Omega(\pi) \tag{10.3}$$

Algorithm 6 Transported Prototypes. Blue lines highlight the OT's contribution in the computational graph of an episode compared to the standard Prototypical Network (Snell, Swersky, and Zemel 2017).

Input: Support set $\mathcal{S} := (x_{s,i}, y_{s,i})_{1 \leq i \leq n_s}$, query set $\mathcal{Q} := (x_{q,j}, y_{q,j})_{1 \leq j \leq n_q}$, classes $\mathcal{C}$, backbone φ_θ.
Output: Loss $\mathcal{L}(\theta)$ for a randomly sampled episode.

1: $z_{s,i}, z_{q,j} \leftarrow \varphi(x_{s,i}; \theta), \varphi(x_{q,j}; \theta)$, for i, j ▷ Get representations.
2: $\mathbf{C}_\theta(i,j) \leftarrow ||z_{s,i} - z_{q,j}||^2$, for i, j ▷ Cost-matrix.
3: $\pi^\star_\theta \leftarrow$ Solve Equation 10.2 ▷ Transportation plan.
4: $\hat{\pi}^\star_\theta(i,j) \leftarrow \pi^\star_\theta(i,j) / \sum_j \pi^\star_\theta(i,j)$, for i, j ▷ Normalization.
5: $\hat{\mathbf{S}} = (\hat{z}_{s,i})_i \leftarrow$ Given by Equation 10.4 ▷ Get transported support set.
6: $\hat{\mathbf{c}}_k \leftarrow \frac{1}{|\hat{\mathbf{S}}_k|} \sum_{\hat{z}_S \in \hat{\mathbf{S}}_k} \hat{z}_S$, for $k \in \mathcal{C}$. ▷ Get transported prototypes.
7: $p_\theta(y|x_{q,j}) \leftarrow$ From Equation 10.5, for j
8: **Return:** $\mathcal{L}(\theta) := \frac{1}{n_q} \sum_{j=1}^{n_q} -\log p_\theta(y_{q,j}|x_{q,j})$.

where $\Omega(\pi) = \sum_{i,j=1}^{n_S, n_T} \pi(i,j) \log \pi(i,j)$ is the negative entropy. It promotes smoother transportation plan while allowing to derive a computationally efficient algorithm, based on Sinkhorn-Knopp's scaling matrix approach (Knight 2008). In our experiment, we set $\varepsilon = 0.05$, but it is possible to tune it, eventually meta-learning it.

10.3.3 Method

Transportation Phase. At each episode, we are provided with a source support set $\mathcal{S}$ and a target query set $\mathcal{Q}$. We note respectively $\mathbf{S}$ and $\mathbf{Q}$ their representations from a deep network $\varphi(\cdot; \theta)$ *i.e.* $z_S \in \mathbf{S}$ is defined as $z_s := \varphi(x_s; \theta)$ for $x_s \in \mathcal{S}$, respectively $z_q \in \mathbf{Q}$ is defined as $z_q := \varphi(x_q; \theta)$ for $x_q \in \mathcal{Q}$. As these two sets are sampled from different distributions, $\mathbf{S}$ and $\mathbf{Q}$ are likely to lie in different regions of the representation space. In order to adapt the source support set $\mathcal{S}$ to the target domain, which is only represented by the target query set $\mathcal{Q}$, we follow (Courty et al. 2016) to compute $\hat{\mathbf{S}}$ the *barycenter mapping* of $\mathcal{S}$, that we refer to as the *transported support set*, defined as follows:

$$\hat{\mathbf{S}} := \hat{\pi}^\star_\theta \mathbf{Q} \tag{10.4}$$

where $\pi^\star_\theta$ is the transportation plan from $\mathbf{S}$ to $\mathbf{Q}$ and $\hat{\pi}^\star_\theta := \pi^\star_\theta(i,j) / \sum_{j=1}^{n_t} \pi^\star_\theta(i,j)$. The *transported* support set $\hat{\mathbf{S}}$ is an estimation of labelled examples in the target domain using labelled examples in the source domain. The success relies on the fact that transportation conserves labels, *i.e.* a query instance close to $\hat{z}_s \in \hat{\mathbf{S}}$ should share the same label with x_S, where $\hat{z}_s$ is the barycenter mapping of $z_s \in \mathbf{S}$. See step (3) of Figure 10.3 for a visualization of the transportation phase.

Prototype Phase. For each class $k \in \mathcal{C}$, we compute the *transported prototypes* $\hat{\mathbf{c}}_k := \frac{1}{|\hat{\mathbf{S}}_k|} \sum_{\hat{z}_s \in \hat{\mathbf{S}}_k} \hat{z}_s$ (where $\hat{\mathbf{S}}_k$ is the transported support set with class k and $\mathcal{C}$ are classes of current episode). We classify each query x_q with representation $z_q = \varphi(x_q; \theta)$ using its euclidean distance to each transported prototypes;

$$p_\theta(y = k|x_q) := \frac{\exp\left(-||z_q - \hat{\mathbf{c}}_k||^2\right)}{\sum_{k' \in \mathcal{C}} \exp\left(-||z_q - \hat{\mathbf{c}}_{k'}||^2\right)} \tag{10.5}$$

	Meta-CIFAR100-C		miniImageNet-C		FEMNIST-FS
	1-shot	**5-shot**	**1-shot**	**5-shot**	**1-shot**
ProtoNet	30.02 ± 0.40	42.77 ± 0.47	36.37 ± 0.50	47.58 ± 0.57	84.31 ± 0.73
MatchingNet	30.71 ± 0.38	41.15 ± 0.45	35.26 ± 0.50	44.75 ± 0.55	84.25 ± 0.71
TransPropNet†	$\mathbf{34.15 \pm 0.39}$	47.39 ± 0.42	24.10 ± 0.27	27.24 ± 0.33	86.42 ± 0.76
FTNet†	28.91 ± 0.37	37.28 ± 0.40	39.02 ± 0.46	51.27 ± 0.45	86.13 ± 0.71
TP† (ours)	$\mathbf{34.00 \pm 0.46}$	$\mathbf{49.71 \pm 0.47}$	$\mathbf{40.49 \pm 0.54}$	$\mathbf{59.85 \pm 0.49}$	$\mathbf{93.63 \pm 0.63}$
TP w/o OT †	32.47 ± 0.41	48.00 ± 0.44	40.43 ± 0.49	53.71 ± 0.50	90.36 ± 0.58
TP w/o TBN †	33.74 ± 0.46	49.18 ± 0.49	37.32 ± 0.55	55.16 ± 0.54	92.31 ± 0.73
TP w. OT-TT †	32.81 ± 0.46	48.62 ± 0.48	$\mathbf{44.77 \pm 0.57}$	$\mathbf{60.46 \pm 0.49}$	$\mathbf{94.92 \pm 0.55}$
TP w/o ET †	$\mathbf{35.94 \pm 0.45}$	48.66 ± 0.46	42.46 ± 0.53	54.67 ± 0.48	94.22 ± 0.70
TP w/o SQS †	*85.67 ± 0.26*	*88.52 ± 0.17*	*64.27 ± 0.39*	*75.22 ± 0.30*	*99.72 ± 0.07*

Table 10.2: Top-1 accuracy of few-shot learning models in various datasets and numbers of shots with 8 instances per class in the query set (except for FEMNIST-FS: 1 instance per class in the query set), with 95% confidence intervals. The top half of the table is a comparison between existing few-shot learning methods and Transported Prototypes (TP). The bottom half is an ablation study of TP. OT denotes Optimal Transport, TBN is Transductive Batch-Normalization, OT-TT refers to the setting where Optimal Transport is applied at test time but not during episodic training, and ET means episodic training *i.e.* w/o ET refers to the setting where training is performed through standard Empirical Risk Minimization. TP w/o SQS reports model's performance in the absence of support-query shift. † flags if the method is transductive. For each setting, the best accuracy among existing methods is shown in bold, as well as the accuracy of an ablation if it improves TP. We note ProtoNet (Snell, Swersky, and Zemel 2017), MatchingNet (Vinyals et al. 2016), TransPropNet (Liu et al. 2019b) and FTNet (Dhillon et al. 2020).

Crucially, the standard Prototypical Networks (Snell, Swersky, and Zemel 2017) computes euclidean distance to each prototypes while we compute the euclidean to each *transported* prototypes, as presented in step (4) of Figure 10.3. Note that our formulation involves the query set in the computation of $(\hat{\mathbf{c}}_k)_{k \in \mathcal{C}}$.

Genericity of OT. `FewShiftBed` implements OT as a stand-alone module that can be easily plugged into any FSL algorithm. We report additional baselines in Appendix B where other FSL algorithms are equipped with OT. This technical choice reflects our insight that OT may be ubiquitous for addressing FSQS and makes its usage in the testbed straightforward.

10.4 Experiments

We compare the performance of baseline algorithms with *Transported Prototypes* on various datasets and settings. We also offer an ablation study in order to isolate the source to the success of *Transported Prototypes*. Extensive results are detailed in Appendix B. Instructions to reproduce these results can be found in the code's documentation.

Setting and details. We conduct experiments on all methods and datasets implemented in `FewShiftBed`. We use a standard 4-layer convolutional network for our experiments on Meta-CIFAR100-C and FEMNIST-FewShot, and a ResNet18 for our experiments on miniImageNet. Transductive methods are equipped with a Transductive Batch-Normalization. All episodic training runs contain 40k episodes, after which we retrieve model state with best validation accuracy. We run each individual

experiment on three different random seeds. All results presented in this paper are the average accuracies obtained with these random seeds.

Analysis. The top half of Table 10.2 reveals that Transported Prototypes (TP) outperform all baselines by a strong margin on all datasets and settings. Importantly, baselines perform poorly on FSQS, demonstrating they are not equipped to address this challenging problem, stressing our study's significance. It is also interesting to note that the performance of transductive approaches, which is significantly better in a standard FSL setting (Liu et al. 2019b; Dhillon et al. 2020), is here similar to inductive methods (notably, TransPropNet (Liu et al. 2019b) fails loudly without Transductive Batch-Normalization showing that propagating label with non-overlapping support/query can have a dramatic impact, see Appendix B). Thus, FSQS deserves a fresher look to be solved. Transported Prototypes mitigate a significant part of the performance drop caused by support-query shift while benefiting from the simplicity of combining a popular FSL method with a time-tested UDA method. This gives us strong hopes for future works in this direction.

Ablation study. Transported Prototypes (TP) combines three components: Optimal Transport (OT), Transductive Batch-Normalization (TBN) and episode training (ET). Which of these components are responsible for the observed gain? Following recommendations from Section 10.2.3, we ablate those components in the bottom half of Table 10.2. We observe that both OT and TBN individually improve the performance of ProtoNet for FSQS, and that the best results are obtained when the two of them are combined. Importantly, OT without TBN performs better than TBN without OT (except for 1-shot mIN-C), demonstrating the superiority of OT compared to TBN for aligning distributions in the few samples regime. Note that the use of TaskNorm (Bronskill et al. 2020) is beyond the scope of the paper[3]; we encourage future work to dig into that direction and we refer the reader to the very recent work (Du et al. 2021). We observe that there is no clear evidence that using OT at train-time is better than simply applying it at test-time on a ProtoNet trained without OT. Additionally, the value of Episodic Training (ET) compared to standard Empirical Risk Minimization (ERM) is not obvious. For instance, simply training with ERM and applying TP at test-time is better than adding ET on 1-shot MC100-C, 1-shot mIN-C and FEMNIST-FS, making it an another element to add to the study (Laenen and Bertinetto 2020) who put into question the value of ET. Understanding why and when we should use ET or only OT at test-time is interesting for future works. Additionally, we compare TP with MAP (Hu, Gripon, and Pateux 2020) which implements an OT-based approach for transductive FSL. Their approach includes a power transform to reduce the skew in the distribution, so for fair comparison we also implemented it into Transported Prototypes for these experiments[4]. We also used the OT module only at test-time and compared with two backbones, respectively trained with ET and ERM. Interestingly, our experiments in Table 10.3 show that MAP is able to handle SQS. Finally, in order to evaluate the performance drop related to Support-Query Shift compared to a setting with support and query instances sampled from the same distribution, we test Transported Prototypes on few-shot classification tasks without SQS (TP w/o SQS in Table 10.2), making a setup equivalent to CDFSL. Note that in both cases, the model is trained in an episodic fashion on tasks presenting a Support-Query Shift. These results show

[3] These normalizations are implemented in `FewShiftBed` for future works.

[4] Therefore results in Table 10.3 differ from results in Table 10.2.

	Meta-CIFAR100-C		miniImageNet-C		FEMNIST-FS
	1-shot	**5-shot**	**1-shot**	**5-shot**	**1-shot**
TP*	36.17 ± 0.47	50.45 ± 0.47	45.41 ± 0.54	57.82 ± 0.48	93.60 ± 0.68
MAP*	35.96 ± 0.44	49.55 ± 0.45	43.51 ± 0.47	56.10 ± 0.43	92.86 ± 0.67
TP†	32.13 ± 0.45	46.19 ± 0.47	45.77 ± 0.58	59.91 ± 0.48	94.92 ± 0.56
MAP†	32.38 ± 0.41	45.96 ± 0.43	43.81 ± 0.47	57.70 ± 0.43	87.15 ± 0.66

Table 10.3: Top-1 accuracy with 8 instances per class in the query set when applying Transported Prototypes and MAP on two different backbones: $\star$ is standard ERM (*i.e.* without Episodic Training) and † is ProtoNet (Snell, Swersky, and Zemel 2017). Transported Prototypes performs equally or better than MAP (Hu, Gripon, and Pateux 2020). Here TP includes power transform in the feature space.

that SQS presents a significantly harder challenge than CDFSL, while there is considerable room for improvements.

10.5 Conclusion

We release `FewShiftBed`, a testbed for the under-investigated and crucial problem of Few-Shot Learning when the support and query sets are sampled from related but different distributions, named FSQS. `FewShiftBed` includes three datasets, relevant baselines and a protocol for reproducible research. Inspired from recent progress of Optimal Transport (OT) to address Unsupervised Domain Adaptation, we propose a method that efficiently combines OT with the celebrated Prototypical Network (Snell, Swersky, and Zemel 2017). Following the protocol of `FewShiftBed`, we bring compelling experiments demonstrating the advantage of our proposal compared to transductive counterparts. We also isolate factors responsible for improvements. Our findings suggest that Batch-Normalization is ubiquitous, as described in related works (Bronskill et al. 2020; Du et al. 2021), while episodic training, even if promising on paper, is questionable. As a lead for future works, `FewShiftBed` could be improved by using different datasets to model different domains, instead of using artificial transformations. Since we are talking about domain adaptation, we also encourage the study of accuracy as a function of the size of the target domain, *i.e.* the size of the query set. Moving beyond the transductive algorithm, as well as understanding when meta-learning brings a clear advantage to address FSQS remains an open and exciting problem. `FewShiftBed` brings the first step towards its progress.

11 Conclusion and Perspectives

Contents

11.1 Summary of the contributions of the thesis

Although Machine learning (ML) systems are becoming increasingly prominent in the industry, they are not infallible. Indeed, these systems are sensitive to distribution shift, *i.e.* the data used to train the system is not representative of the real-world data, limiting deploying them in critical applications. The present doctoral thesis contributes to preventing such a lack of robustness of ML systems through the lens of adaptation, *i.e.* system's ability to adapt quickly to a novel situation. Towards learning adaptive models, we organize the contributions of this thesis into four parts.

Part I of this dissertation provides a historical retrospective of the adaptation problem. Chapter 2 starts from an anthropomorphic flavour connecting Jean Piaget's study of intelligence development in children with the founding ideas of Artificial Intelligence (AI) proposed by McCarthy and his peers. As machine learning becomes established in AI, we review some classical failures of those systems due to distribution shift. It motivates our interest into adaptive models, *i.e.* models that implement the principle of adaptation in situations where data is subject to shift. In Chapter 3, we conduct an original review of the related works to adaptation in machine learning, providing the necessary technical background of this thesis and the depiction of the fundamental problem of learning transferable domain invariant representations.

Part II focuses on the theoretical analysis of transferability of domain invariant representations. In Chapter 4, we introduce a new error term, called the *Hypothesis Class Reduction* error, building upon the seminal theory of (Ben-David et al. 2010a). This new term traduces the risk of deleting relevant information in the representation to achieve domain invariance. In particular, we demonstrate this new error term provides the needed theoretical ground of an influential regularization approach called *Batch Spectral Penalization* (Chen et al. 2019c). Chapter 5 unifies two important lines of study: Important Sampling (Quinonero-Candela et al. 2009) and Domain Invariant Representations (Ben-David et al. 2007; Ganin and Lempitsky 2015). This new analysis introduces a crucial term called the *transferability* term, which quantifies the lack of transferability of domain invariant representations. Importantly, this term depends on the label in the target domain, making it out of reach in the standard scenario of Unsupervised Domain Adaptation. To circumvent this limitation, we conduct an analysis of the role of inductive bias in Chapter 6. In particular, we show theoretically that the transferability term can be approximated as long as a strong inductive bias is available, demonstrating the efficiency of a well-adopted practice consisting in relying on target predicted labels during adaptation (Long et al. 2018).

Part III is dedicated to the empirical investigation of theoretical results established in Part II. In Chapter 7, we develop an algorithm that relies on two weak inductive biases, both on weights and representations, called *Robust Domain Adaptation* (RUDA). Our empirical study demonstrates RUDA improves substantially adaptation in the challenging scenario of label shift (Zhao et al. 2019). In Chapter 8, we study the role of strong inductive bias that states that prediction in the target domain shall remain invariant to perturbation of inputs that are uncorrelated to the label. This novel algorithm reaches state-of-the-art performances of methods based on domain invariant representations on several benchmarks. As such strong inductive bias may not be available in a broader range of applications, we study the role of Active Learning to improve adaptation in Chapter 9. In particular, we develop a criterion called

Sage that quantifies the lack of transferability of a target sample. Building a query upon this criterion, we design an annotation strategy that annotates a diverse set of samples for which adaptation has failed. Our experiment demonstrates that Sage improves the state-of-the-art in Active Domain Adaptation.

Part IV describes results prospective ideas towards learning adaptive models with invariant representations. Chapter 10 challenges unsupervised domain adaptation's common assumption where large samples populate both the source and the target domains. To this purpose, we introduce the new problem of adapting with few samples, bridging the gap between few-shot learning and adaptation. Chapter 11 exposes possible future research directions for which this thesis work may provide some foundations.

11.2 Short-term perspectives

11.2.1 Test-Time Adaptation

Statement

Test-Time Adaptation is the problem of adapting a model at test-time and shares strong similarities with the setup of *Unsupervised Domain Adaptation* (UDA) described in Chapter 3 (Definition 3.2.2). Test-Time Adaptation has two stages, thus differentiating it from UDA. The first stage, called *train-time*, involves access to the labelled source data. The second stage, called *test-time*, is marked by the availability of the target unlabeled data. Thus, TTA raises new challenges due to the requirement to adapt rapidly for making real-time predictions on target data at test-time.

Should we assume access to source labelled data at test-time. Indeed, source data storage could be a bottleneck for real-time prediction and when data is decentralized or protected by privacy issues. Such consideration motivates the field of *Source free adaptation* (Chidlovskii, Clinchant, and Csurka 2016; Liang, Hu, and Feng 2020), when a model is trained at train-time on source labelled data and adapted at test-time on target unlabelled data *in absence* of source data.

How to make efficient adaptation algorithm? Deep Unsupervised Domain Adaptation typically adapts all the parameters of the representation and the classifier as presented in Section 3.3. One can reasonably believe that adapting all the parameters is a sub-optimal strategy to adapt rapidly for making real-time predictions. Unsupervised Domain Adaptation (UDA) requires a whole target dataset for inference, limiting its applications. Recent pioneering works isolate the role of *batch-normalization* (Ioffe and Szegedy 2015a) by updating only batch-normalization statistics (Schneider et al. 2020a) or parameters by minimizing the entropy of predictions (Wang et al. 2021a), despite the recent criticism of (Burns and Steinhardt 2021).

How to adapt with few target samples? As mentioned in Chapter 10, UDA assumes that source and target data are populated with abundant data. Thus, we have proposed a meta-learning algorithm to address this unrealistic assumption. In a context of TTA, where target data is only available at test-time for making real-time predictions, it raises interesting questions.

The high combinatorial nature A deep neural network h is parametrized by a set of parameters $\theta \in \Theta$, *i.e.* $h = h(\cdot; \theta)$. Thus a natural way to adapt the model is to remove some parameters by masking operations;

$$h_{\text{modular}} = h(\cdot; \mathbf{m} \odot \theta), \text{ where } \mathbf{m} \in \{0,1\}^{|\Theta|} \tag{11.1}$$

where $|\Theta|$ is the number of parameters, $\mathbf{m}$ is a mask and $\odot$ is the coordinate-wise multiplication of vectors. Finding the optimal mask based is of great complexity due to the high combinatorial nature of the problem. Indeed, there are $2^{|\Theta|}$ choices possible for $\mathbf{m}$. The work (Zhang et al. 2021a) addresses this optimization problem introducing *Modular Risk Minimization* in a context of *Domain Generalization* (Gulrajani and Lopez-Paz 2021), hence the name of $h(\cdot; \mathbf{m} \odot \theta)$ as *modular* or *subnetwork* (Zhang et al. 2021a). To move from Domain Generalization to Test-Time Adaptation faces the challenges described above. How to find the optimal subnetwork quickly, eventually from a small number of target unlabelled samples? Seemingly, addressing such a question needs to implement heuristic (inductive bias) to reduce the exploration of the high combinatorial problem $2^{|\Theta|}$.

Adaptive models

Statement. Selecting a subnetwork is a combinatorial problem with high cardinality raises the issue of selecting rapidly such a subset from a small number of unlabelled target samples. As a short-term perspective to address these issues, we suggest to develop (simple) *adaptive models* defined as a class of deep neural networks that depend on two set of parameters θ and λ noted;

$$h(\cdot; \theta_\lambda), \quad \theta \in \Theta, \lambda \in \Lambda, \tag{11.2}$$

where θ are *train-time* parameters, *i.e.* are optimized at train-time in a classical setting of supervised learning, and λ are *test-time* parameters, *i.e.* are optimized at test-time in a setting of source free unsupervised domain adaptation. For instance, in the case of a modular network $h_{\text{modular}} = h(\cdot; \mathbf{m} \odot \theta)$, the *test-time* parameters are $\mathbf{m} \in \{0,1\}^{\Theta}$. Crucially, an adaptive model depends on data on which it will be applied, through the determination of the test-time parameters λ. The seminal idea that network's parameters can be function of the inputs itself dates back the introduction of *hyper-networks* (Ha, Dai, and Le 2016), where their *modularity*, defined as the ability to effectively learn a different function for each input, has been theoretically studied in (Galanti and Wolf 2020) and applied for continual learning (Oswald et al. 2019) or hyper-parameters optimization (Lorraine and Duvenaud 2018). Recently, adapting at test-time network's parameters has been sucesfully applied for model calibration (Zhou and Levine 2021) under distribution shift or for fast transfer learning (Maddox et al. 2021).

Spectral Filtering. As insight for future research is to develop such family of networks where the number of test-time parameters is limited, allowing fast adaptation with a small number of unlabelled target samples. Let us implement an illustrative simple example. We present a family of adaptive models with only one test-time parameter, here $\lambda \in \Lambda = \mathbb{R}^+$, through the lens of L^2 regularization. More precisely,

$$\theta^\star_\lambda := \arg\min_{\theta \in \Theta} L_{c,S}(\theta) + \lambda \cdot ||\theta||^2 \tag{11.3}$$

where $L_{c,S}(\theta) := \mathbb{E}_S[-\mathbf{Y} \cdot \log h(X;\theta)]$. L^2 penalization has been an ubiquitous tool for preventing to overfit the training data, thus a natural question raises: can it help to prevent distribution shift? Formally, can we determine $\lambda^\star$ based on few target unlabelled target samples[1] resulting in a model $h(\cdot, \theta^\star_{\lambda^\star})$ that performs better on the target data?

To gain insight about this problem, let us go back to the L^2 penalization of linear regressor, also known as *Ridge Regression*. Formally, for a regressor $h(\cdot,\theta)$ that fits Y based on X for some distribution $p(X,Y)$, *i.e.* $Y \approx X^\top\theta$, where θ is obtained by achieving a trade-off between a small risk and small values of parameters;

$$\theta_\lambda := \arg\min_\theta \mathbb{E}\left[\|Y - X^\top\theta\|^2\right] + \lambda \cdot \|\theta\|^2, \tag{11.4}$$

the optimal $\theta^\star_\lambda$ depends on λ is $\theta^\star_\lambda = \left(\mathbb{E}[X^\top X] + \lambda I\right)^{-1}\mathbb{E}[X^\top Y]$. Crucially;

$$\theta^\star_\lambda = F_\lambda \theta^\star_0, \text{ where } F_\lambda := \left(\mathbb{E}[X^\top X] + \lambda I\right)^{-1}\mathbb{E}[X^\top X], \tag{11.5}$$

thus adapted parameters $\theta^\star_\lambda$ are simply obtained from non-adapted parameters $\theta^\star_0$ by multiplying parameters by the matrix F_λ, that we refer to as the *Spectral Filtering* matrix. This suggests the simple procedure;

- **Train-time:** Acquire labelled source data and fit $\theta^\star_0$ from source labelled data, *i.e.* learn without L^2 penalization.
- **Test-time:** Acquire unlabelled target data and fit optimal $\lambda^\star$ by minimizing some loss L_u on unlabelled target data;

$$\lambda^\star = \arg\min_\lambda L_u(F_\lambda \theta^\star_0), \tag{11.6}$$

  i.e. adjust the L^2 penalization to better fit the unlabelled target data.

Crucially, in the presented setting, at test-time we only fit one parameter λ that calibrates the strength of the L^2 penalization.

To illustrate our proposal on a real-world case with deep neural networks, we extrapolate this analysis as follows; let $h(\cdot,\theta) = f_{\theta^L} \circ \cdots \circ f_{\theta^1}$ a deep neural network, where f_{θ^ℓ} is a simple non-linear function for $\ell \in \{1, \cdots, L\}$, the adapted model is $h(\cdot,\theta_\lambda) = f_{F_\lambda\theta^L} \circ \cdots \circ f_{F_\lambda\theta^1}$. Crucially, spectral filtering follows the analysis of *negative transfer* from (Chen et al. 2019b) showing that that both learned parameters (θ) and feature representations ($\varphi(X)$) are partially transferable. In particular, the lower layers of a deep neural network are transferable while eigen-vectors of representations associated to the smaller singular values are the less transferable. This suggests to shrink the eigen-vectors of representations, *i.e.* removing eigen-vectors associated with the smallest eigen-values, in order to make features more transferable (Chen et al. 2019b). L^2 penalization has the vertue of smoothly shrinking eigen-vectors with smaller eigen-values.

Preliminary experiments on a classical benchmark to evaluate the robustness of a model using corrupted data are presented in Table D.1, where we used as loss L_u entropy of predictions (Grandvalet and Bengio 2005). Spectral Filter improves adaptation on some corruptions, *e.g.* gaussian noise, while degrading on other corruptions,

[1] without retraining the model on source labelled data for fast adaptation

e.g. Defocus. It suggests that different approaches should be deployed depending on the shift, motivating our interest into data shift interpretation that we present in the following section.

11.2.2 Interpretability of distribution shift

Statement

Developing an efficient Test-Time Adaptation algorithm is seemingly related to the type of shifts encountered on real-world data. Our first empirical investigation from Table D.1 shows that shifts do not respond equally to adaptation strategy, *e.g.* Spectral Filter improves performance on Gaussian noise injected on images compared to Tent (Wang et al. 2021a). At the same time, we draw the opposite conclusion when defocus noise is applied to images. Such evidence suggests that there is no free lunch in test-time adaptation. Therefore, an adaptation strategy should be applied by carefully observing the shift observed on real-world data, motivating our interest in interpretability of distribution shift for driving the adaptation method.

A random matrix theory approach

For future research, we aim to investigate results from *Random Matrix Theory* (RMT) towards distribution shift interpretability. We rely on the founding theorem of RMT providing condition on eigen-values of a covariance matrix obtained from a finite set of samples.

Theorem 11.1 (Marchenko–Pastur's bulk)**.** *Let* $\mathbf{X} \in \mathbb{R}^{n \times m}$ *where entries are independant and identically distributed,* $\mathbb{E}[\mathbf{X}_{ij}] = 0$ *and* $\mathbb{V}[\mathbf{X}_{ij}] = \sigma^2 < +\infty$ *for* $i \in \{1, \cdots, n\}, j \in \{1, \cdots, m\}$*. Let the covariance matrix of* X*, noted* C*;*

$$\hat{\mathbf{C}} = \frac{1}{n}\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{m \times m} \tag{11.7}$$

Assume that $\frac{m}{n} \to \lambda \in \mathbb{R}^+$ *when* $m, n \to \infty$*. The spectrum of* $\mathbf{C}$*, noted* $\mathrm{Sp}(\mathbf{C})$ *verifies;*

$$\mathrm{Sp}(\mathbf{C}) \subset [\lambda_-, \lambda_+] \tag{11.8}$$

where $\lambda_\pm = \sigma^2 \left(1 \pm \sqrt{\lambda}\right)^2$.

Crucially, $[\lambda_-, \lambda_+]$ is named the Marchenko–Pastur's bulk (MP's bulk); if one observes an eigenvalue out of MP's bulk, thus the correlation observed does not result from estimation noise but from an actual correlation in the data. Note this theorem is asymptotic, *i.e.* it holds $\frac{m}{n} \to \lambda$ when $m, n \to \infty$, but it remains powerful even in the finite sample regime, *i.e.* when both m and n are finite. We typically apply such an approach for filtering covariance matrices (Bouchaud and Potters 2009; Laloux et al. 2000; Bun, Bouchaud, and Potters 2017). Besides, RMT has already been succesfully applied in a context of Transfer Learning (Seddik 2020).

Towards distribution shift interpretability, let note the covariance matrix in the domain $D \in \{S, T\}$ as follows;

$$\mathbf{C}_D := \mathbb{E}\left[(X - \mathbb{E}_D[X])^\top (X - \mathbb{E}_D[X])\right] \tag{11.9}$$

and $\widehat{\mathbf{C}_D}$ their empirical counterparts. Given the n IID, noted $\mathbf{X} := (X_1, \cdots, X_n) \in \mathbb{R}^{n \times m}$ realizations from $p_T(X)$, we set

$$\mathbf{X}' = \mathbf{C}_S^{-\frac{1}{2}} \mathbf{X} \text{ and } \hat{\mathbf{C}}' = \frac{1}{n} \mathbf{X}'^{\top} \mathbf{X}' \tag{11.10}$$

Crucially, if distribution does not shift, *i.e.* $p_S = p_T$, eigen-values of $\hat{\mathbf{C}}'$ are located in the bulk in $\left[\left(1 - \sqrt{\frac{m}{n}}\right)^2, \left(1 + \sqrt{\frac{m}{n}}\right)^2\right]$. If one observes an eigen-value out of the bulk, then one can interpret shift through the *direction* where distribution shift takes place. Indeed, shift from the source to the target distribution is directed by the vector $\mathbf{C}_S^{\frac{1}{2}} v$, where v is the eigen-vector associated to the eigen-value out of the bulk. Note our description eludes an insidious effect since in practice we do not have to the true source correlation matrix $\mathbf{C}_S$, only the empirical counterparts $\widehat{\mathbf{C}_S}$. Thus, our proposal deserves deeper mathematical investigations.

11.2.3 Quantifying Malignency of distribution shift

Statement

Quantifying malignency of distribution shift consists in estimating drop of model's performance on data subject to shift. Although this study line could significantly impact our understanding of the risks involved when applying a model on shifted data, few prior works address this question. The work (Rabanser, Günnemann, and Lipton 2019) suggests to annotate a selected subset of target samples, thus needing human intervention, while (Elsahar and Gallé 2019) shows the $\mathcal{H}-$divergence (Kifer, Ben-David, and Gehrke 2004) (see Section 3.2) is well-correlated with the actual drop of performances.

Learning shift malignency

As insight for future research, we develop the line of study of *learning* shift malignency, *i.e.* given a model h trained on source labelled data, a small number of unlabelled target data $\mathcal{D}_T$ as an IID sample from the target distribution p_T, we learn a model ζ such that;

$$\zeta(\mathcal{D}_T) \approx \text{Err}_T(h) \tag{11.11}$$

This challenging problem raises interesting questions. First, the task is to map a dataset to the error of the model commits on this dataset. Thus, to cast it as learning task, one should compose a set of N training examples expressed as $(\mathcal{D}_i, \text{Err}_{p_i}(h))_{1 \leq i \leq N}$ where for $i \in \{1, \cdots, N\}$, $\mathcal{D}_i$ is a IID sample from some distribution p_i. To some extent, one can describe this problem as an instance of meta-learning. Second, ζ is a function that should be applied on a whole dataset, *i.e.* not a single point. In particular each dataset instance may have a varying size, *i.e.* each dataset may have a varying number of samples. A natural way to deal with a dataset instance with varying size is to use the dataset $\mathcal{D}$ mean as a *representation* of $\mathcal{D}$. Formally, ζ is designed as follows;

$$\zeta_{w,b}(\mathcal{D}) = \sigma\left(w^{\top} z + b\right), \text{ where } z := \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} x \tag{11.12}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function and w and b are parameters to learn. As mentionned above, one can interpret z as the representation of the dataset $\mathcal{D}$,

thus one can easily generalize this principle to $z := \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \varphi(x)$ where φ is a deep neural network. Formally, ζ is designed as follows;

$$\zeta_{w,b,\varphi}(\mathcal{D}) = \sigma\left(w^\top z + b\right), \text{ where } z := \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \varphi(x) \tag{11.13}$$

Such intuition can also be extended to mean embedding of a distribution using kernels (Muandet et al. 2016).

11.3 Long-term perspectives

11.3.1 Inductive Adaptation

Most of the algorithms presented in this thesis rely on the distribution alignment of representations, *i.e.* the principle of invariance. We measure invariance by comparing two samples, the former from the source distribution and the latter from the target distribution. Such a principle falls into the *transductive learning* paradigm presented in Section 3.1.4. Indeed, we perform adaptation on a predefined set of samples; here, we predict the unlabelled target samples as a whole.

However, one can argue that humans can adapt with very few, or one, unlabelled target samples as presented in Figure 2.3 of Chapter 2. One could refer to the new challenge of adapting with one target sample as *inductive* adaptation. To describe formally this new paradigm, we rely on the description of hyper-networks (Ha, Dai, and Le 2016) and Test-Time Adaptation as presented in Section 11.2.1; given a model $h(\cdot;\theta)$ with parameters θ, and a sample $x \sim p_T(X)$, model's parameter should consider x itself to adapt, *i.e.* model's prediction is $h(x;\theta(x))$.

Learning model able to adapt with one target sample, *i.e.* *inductive* adaptation, is beyond our current understanding of adaptation. Indeed, most theoretical results rely on distribution alignment and limiting the implementation of adaptation to the transductive setting. We should mention the promising results from the pioneering work (Sun et al. 2020) that is, to our knowledge, the first implementation of the principle of *inductive adaptation*. (Sun et al. 2020) suggests to adapt a object recognition model by fine-tuning representations at test-time on a pretext task, here predicting the rotation of the image. The adaptation likely takes place because of the nature of the pretext task, which reflects an inductive bias that we wish to enforce in this kind of model; it must remain robust to image rotations. For more general tasks, one can already imagine that implementing a powerful inductive bias is necessary to perform inductive adaptation.

11.3.2 Interactive Adaptation

In Chapter 9 we have presented the problem of *Active Adaptation* following the work from (Su et al. 2020). Active Adaptation attracts recently more attention from the community as evidenced by the contributions (Prabhu et al. 2020; Fu et al. 2021). The paradigm of Active Adaptation relies on an Oracle, *i.e.* an expert, who knows the ground-truth in the target domain. Seemingly, as the annotation budget grows, one can expect to reach performances that we would obtain if labelled target data were available in a standard supervised learning setting. However, a limited budget characterizes most real-world applications. Besides, human assistance for tasks requiring real-time predictions may dramatically slow the process.

Investigating *Interactive Adaptation* is an exciting venue for future works, *i.e.* human assistance beyond the active setting where the human outputs are restricted to labelling data. For instance, based on the data shift observed, one can generate simple labelling rules that the expert should validate. To illustrate such an idea, let consider the problem of predicting revenue from features of the person, including diploma, occupation or gender, as is the case for the census dataset[2]. We assume that the training data is biased at the expense of women; women are mostly represented as having occupations that are less likely to have high incomes. Now, we apply such a model to real-world data that does not exhibit, or significantly less, such a bias. In particular, we observe that gender feature has shifted, for instance, the law that drives gender given diploma and occupation shifts. An interactive adaptation system returns to the expert;

- *"Gender value promotes smaller revenue*[3] *while diploma and occupation indicate a higher revenue."*
- *"Should I override gender to only focus on the diploma and occupation?"*

Suppose the fairness value drives the expert. In that case, it validates the labelling rule that only focuses on the diploma and revenue, resulting in improved performances on real-world data. To echo our theoretical analysis from Part II, building and validating labelling rule by an expert can be incorporated into learning domain invariant representations as presented in the case of Active Adaptation from Chapter 9. Building labelling function is a fruitful line of research, also referred to as *Data Programming* (Ratner et al. 2016). For instance a recent family of models called *Concept Bottleneck Models* (Koh et al. 2020) are designed for promoting interacting using high-level concepts.

11.3.3 Generic Adaptation

The thesis focused on learning representations through deep neural networks where all the illustrative experiments are conducted on image data. We shall challenge such a technical choice since the industrial needs for adaptive models go beyond our analysis. Indeed, this omits the prominence of tabular data where more classical learning models, such as random forest (Ho 1995) or gradient boosting (Friedman 2001), are widely adopted by the community as described in (Shwartz-Ziv and Armon 2021). As future work, it is undoubtedly interesting to understand how the results obtained for representation learning can benefit these algorithms that do not rely on this paradigm and the development of powerful inductive bias for tabular data.

[2] `archive.ics.uci.edu/ml/datasets/Census+Income`

[3] resulting to the bias in training data

A Learning Representations with Deep Neural Networks

A deep neural network h is a mapping from $\mathcal{X}$ to $\mathcal{Y}$;

$$h = f_{(w_L,b_L)} \circ \cdots \circ f_{(w_\ell,b_\ell)} \circ \cdots \circ f_{(w_1,b_1)} \tag{A.1}$$

parametrized by $(w_1, b_1, \cdots, w_L, b_L)$ where $f_{(w_\ell,b_\ell)}(x) = a\left(x^\top w_\ell + b_\ell\right)$ with a some non-linear function, typically ReLU, *i.e.* $a(x) = \max(x, 0)$. We say that h is deep network with L layers as it is the composition of L simple functions, here $f_{(w_\ell,b_\ell)}$ for $\ell \in \{1, \cdots, L\}$.

Crucially, each simple function transforms the inputs, leading to a representation of the inputs, *i.e.* the representation z_ℓ at layer $\ell < L$ is;

$$z_\ell := f_{(w_\ell,b_\ell)} \circ \cdots \circ f_{(w_1,b_1)}(x) \tag{A.2}$$

associated with a classifier $g_\ell := f_{(w_L,b_L)} \circ \cdots \circ f_{(w_{\ell+1},b_{\ell+1})}$.

For the sake of simplicity, we will consider a deep network h as the composition of a representation layer φ, from a set of representations Φ, and a classifier g, from a set of classifier $\mathcal{G}$. More precisely, we now consider a representation $\mathcal{Z}$ where Φ is a subset of functions from the input space $\mathcal{X}$ to the representation space $\mathcal{Z}$. The set of classifiers is a subset of functions from the representation space $\mathcal{Z}$ to the output space $\mathcal{Y}$;

$$h = g \circ \varphi. \tag{A.3}$$

From this perspective, the hypothesis class of deep neural networks is $\mathcal{H} = \{g \circ \varphi; g \in \mathcal{G}, \varphi \in \Phi\}$, noted $\mathcal{G}\Phi$. Note that both φ and g can also be deep neural networks.

B Proofs

B.1 Proofs of Chapter 7

Proposition B.1.1 (Inductive bias of weights 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:*

$$w(z') = \frac{p_T(z')}{p_S(z')} \quad \textit{and} \quad p_S(z|z') = p_T(z|z') \tag{B.1}$$

Furthermore, if additionally $\mathrm{Err}_S(g_S\varphi) = \mathrm{Err}_T(g_T\varphi) = 0$*, then $p_S(Y|Z) = p_T(Y|Z)$ and $P_S(Y|Z') = p_T(Y|Z')$.*

Proof. First, we show $w(z') = p_T(z')/p_S(z')$. Indeed,

$$\begin{aligned}
\mathrm{INV}(w, \varphi) &= \sup_{f \in \mathcal{F}} \mathbb{E}_S[w(Z')f(Z)] - \mathbb{E}_T[f(Z)] && \text{(B.2)}\\
&\geq \sup_{f \in \mathcal{F}} \mathbb{E}_S[w(Z')f \circ \psi(Z)] - \mathbb{E}_T[f \circ \psi(Z)] && (\mathcal{F} \circ \psi \subset \mathcal{F})\\
&\geq \sup_{f \in \mathcal{F}} \mathbb{E}_S[w(Z')f(Z')] - \mathbb{E}_T[f(Z')] = 0 && (Z' = \psi(Z))
\end{aligned}$$

Thus, $\sup_{f \in \mathcal{F}} \mathbb{E}_S[w(Z')f(Z')] - \mathbb{E}_T[f(Z')] = 0$ resulting to $w(Z') = p_T(Z')/p_S(Z')$.

Second, $\mathrm{INV}(w, \varphi) = 0$ also implies that $w(z')p_S(z) = p_T(z)$. Indeed, we note that $p_D(Z = z) = \int p_D(Z = z, Z' = z')\mathrm{d}z' = p_D(Z = z, Z' = \psi(z))$ since $z' = \psi(z)$ where ψ is a deterministic function.

$$\begin{aligned}
w(z') = \frac{p_T(z)}{p_S(z)} &= \frac{p_T(Z = z, Z' = \psi(z))}{p_S(Z = z, Z' = \psi(z))} && \text{(B.3)}\\
&= \frac{p_T(Z = z|Z' = \psi(z))}{p_S(Z = z|Z' = \psi(z))} \frac{p_T(Z' = \psi(z))}{p_S(Z' = \psi(z))} && \text{(B.4)}\\
&= \frac{p_T(z|z')}{p_S(z|z')} w(z') && \text{(B.5)}
\end{aligned}$$

which shows that $p_T(z|z')/p_S(z|z') = 1$.

Finally, if additionally $\text{Err}_S(g_S\varphi) = \text{Err}_T(g_T\varphi) = 0$, $\text{TSF}(w,\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')\mathbf{Y}\cdot\mathbf{f}(Z)] - \mathbb{E}_T[\mathbf{Y}\cdot\mathbf{f}(Z)]$, leading to

$$\text{TSF}(w,\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')\mathbf{Y}\cdot\mathbf{f}(Z)] - \mathbb{E}_T[\mathbf{Y}\cdot\mathbf{f}(Z)] \tag{B.6}$$

$$= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{Y}\cdot\mathbf{f}(Z)]\right] - \mathbb{E}_{Z'\sim p_T}\left[\mathbb{E}_{Z|Z'\sim p_T}[\mathbf{Y}\cdot\mathbf{f}(Z)]\right] \tag{B.7}$$

Noting that $w(z')p_S(z') = p_T(z')$;

$$\text{TSF}(w,\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{Y}\cdot\mathbf{f}(Z)]\right] - \mathbb{E}_{Z'\sim p_T} w(Z')\left[\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{Y}\cdot\mathbf{f}(Z)]\right] \tag{B.8}$$

$$= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{Y}\cdot\mathbf{f}(Z)] - \mathbb{E}_{Z|Z'\sim p_T}[\mathbf{Y}\cdot\mathbf{f}(Z)]\right)\right] \tag{B.9}$$

$$= \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{f}_S(Z)\cdot\mathbf{f}(Z) - \mathbf{f}_T(Z)\cdot\mathbf{f}(Z)]\right)\right] \tag{B.10}$$

Noting that $p_S(z|z') = p_T(z|z')$ and choosing $\mathbf{f} = \frac{1}{2}(\mathbf{f}_S - \mathbf{f}_T)$;

$$\text{TSF}(w,\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[\mathbf{f}_S(Z)\cdot\mathbf{f}(Z) - \mathbf{f}_T(Z)\cdot\mathbf{f}(Z)]\right)\right] \tag{B.11}$$

$$\geq 2\mathbb{E}_{Z'\sim p_S}\left[w(Z')\left(\mathbb{E}_{Z|Z'\sim p_S}[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2]\right)\right] \tag{B.12}$$

$$\geq 2\mathbb{E}_{Z'\sim p_T}\left[\left(\mathbb{E}_{Z|Z'\sim p_T}[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2]\right)\right] \tag{B.13}$$

$$\geq 2\mathbb{E}_{Z'\sim p_T}\left[\left(\mathbb{E}_{Z|Z'\sim p_T}[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2]\right)\right] \tag{B.14}$$

$$\geq 2\mathbb{E}_{Z\sim p_T}\left[||\mathbf{f}_S(Z) - \mathbf{f}_T(Z)||^2\right] \tag{B.15}$$

Which leads to $\mathbf{f}_S = \mathbf{f}_T$ and $p_T[Y|Z] = p_S[Y|Z]$. Now we finish by observing that:

$$\text{TSF}(w,\varphi) = \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')Y\cdot\mathbf{f}(Z)] - \mathbb{E}_T[Y\cdot\mathbf{f}(Z)] \tag{B.16}$$

$$\geq \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')Y\cdot\mathbf{f}\circ\psi(Z)] - \mathbb{E}_T[Y\cdot\mathbf{f}\circ\psi(Z)] \tag{B.17}$$

$$\geq \sup_{\mathbf{f}\in\mathcal{F}_C} \mathbb{E}_S[w(Z')Y\cdot\mathbf{f}(Z')] - \mathbb{E}_T[Y\cdot\mathbf{f}(Z')] \tag{B.18}$$

which leads to $p_S[Y|Z'] = p_T[Y|Z']$ following the same proof. □

C Supplemental Target Consistency (Chapter 8)

C.1 Detailed results

Table C.1: Accuracy (%) on VisDA-2017

ResNet-50		ResNet-101	
Method	Synthetic → Real	Method	Synthetic → Real
JAN (Long et al. 2017)	61.6	ResNet-101 (He et al. 2016)	52.4
GTA (Sankaranarayanan et al. 2018)	69.5	DANN (Ganin et al. 2016)	57.4
CDAN (Long et al. 2018)	70.0	CDAN (Long et al. 2018)	73.7
TAT (Liu et al. 2019a)	71.9	BSP (Chen et al. 2019c)	75.9
Ours	**77.5±0.7**	**Ours**	**79.0±0.1**

Table C.2: Accuracy (%) on the 5 hardest Office-Home task for Target Consistency ablation (ResNet-50)

	Ar→Cl	Cl→Ar	Pr→Ar	Pr→Cl	Rw→Cl	Avg
L_{DANN}	45.2±0.7	48.8±0.5	46.8±0.2	43.5±0.3	53.6±0.3	47.6
$L_{\text{DANN}} + L_{\text{VAT}}$	44.3±0.2	50.3±1.8	48.5±1.1	43.6±0.6	53.5±0.2	48.0
$L_{\text{DANN}} + L_{\text{AUG}}$	46.2±0.4	55.3±0.5	53.2±1.4	46.0±0.4	55.6±0.5	51.3
$L_{\text{DANN}} + L_{\text{VAT}} + L_{\text{AUG}}$	46.3±0.6	53.5±1.0	54.7±0.7	46.2±0.7	56.3±0.9	51.4
$L_{\text{DANN}} + L_{\text{VAT}} + L_{\text{AUG}}$ /w MT	46.6±0.3	53.3±0.7	52.8±0.3	46.9±0.8	55.6±0.5	51.0
L_{CDAN}	50.3±0.1	54.6±0.7	55.8±0.6	49.3±0.2	56.9±0.1	53.4
$L_{\text{CDAN}} + L_{\text{VAT}}$	50.1±0.5	58.5±0.6	59.1±0.6	49.8±0.2	57.9±0.1	55.1
$L_{\text{CDAN}} + L_{\text{AUG}}$	51.0±0.2	57.3±0.5	61.0±0.7	50.8±0.2	58.4±0.5	55.7
$L_{\text{CDAN}} + L_{\text{VAT}} + L_{\text{AUG}}$	51.5±0.2	60.9±0.3	61.4±0.9	51.7±0.2	59.1±0.5	56.9
$L_{\text{CDAN}} + L_{\text{VAT}} + L_{\text{AUG}}$ /w MT	51.3±0.9	59.0±0.4	60.0±0.5	51.8±0.2	57.9±0.3	56.0
L_{TSF}	52.6±0.8	60.1±0.3	60.6±0.9	52.1±0.7	58.3±0.4	56.7
$L_{\text{TSF}} + L_{\text{VAT}}$	52.4±0.6	60.1±0.5	61.2±0.9	53.1±0.2	58.9±0.8	57.1
$L_{\text{TSF}} + L_{\text{AUG}}$	53.1±0.5	62.3±0.6	62.6±0.8	53.1±1.0	59.5±0.3	58.1
$L_{\text{TSF}} + L_{\text{VAT}} + L_{\text{AUG}}$	53.0±0.1	**62.8±0.7**	62.8±0.2	53.8±0.8	**60.8±0.8**	58.6
$L_{\text{TSF}} + L_{\text{VAT}} + L_{\text{AUG}}$ /w MT	**53.1±1.5**	62.6±0.1	**63.8±0.7**	**54.4±0.6**	60.4±0.6	**58.9**

Table C.3: mIoU on **GTA5** → **Cityscapes**. *AdvEnt+MinEnt** is an ensemble of two models.

Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
ResNet-101 (He et al. 2016)	75.8	16.8	77.2	12.5	21.0	25.5	30.1	20.1	81.3	24.6	70.3	53.8	26.4	49.9	17.2	25.9	6.5	25.3	36.0	36.6
Adapt-SegMap (Tsai et al. 2018)	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
AdvEnt (Vu et al. 2019)	89.9	36.5	81.6	29.2	25.2	28.5	32.3	22.4	83.9	34.0	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8
Ours	**91.0**	**41.9**	**81.6**	**30.1**	22.6	26.0	28.8	13.6	82.6	**37.2**	**81.9**	56.1	**29.3**	**84.8**	**34.1**	**48.8**	0.0	26.8	**35.7**	44.9
AdvEnt+MinEnt* (Vu et al. 2019)	89.4	33.1	81.0	26.6	**26.8**	**27.2**	**33.5**	**24.7**	**83.9**	**36.7**	78.8	**58.7**	**30.5**	**84.8**	**38.5**	**44.5**	**1.7**	**31.6**	32.4	**45.5**

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet-50 (He et al. 2016)	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
DAN (Long et al. 2015)	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN (Ganin et al. 2016)	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
JAN (Long et al. 2017)	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN (Long et al. 2018)	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
TAT (Liu et al. 2019a)	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
BSP (Chen et al. 2019c)	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	72.2	59.3	81.9	66.3
TransNorm (Wang et al. 2019)	50.2	71.4	77.4	59.3	**72.7**	**73.1**	61.0	53.1	79.5	71.9	59.0	82.9	67.6
Ours	**53.1**	**73.0**	**77.0**	**62.6**	72.4	**73.1**	**63.8**	**54.4**	**79.8**	**74.6**	**60.4**	**83.3**	**69.0**

Table C.4: Accuracy (%) on Office-Home for unsupervised domain adaptation (ResNet-50)

C.2 Fourier Analysis

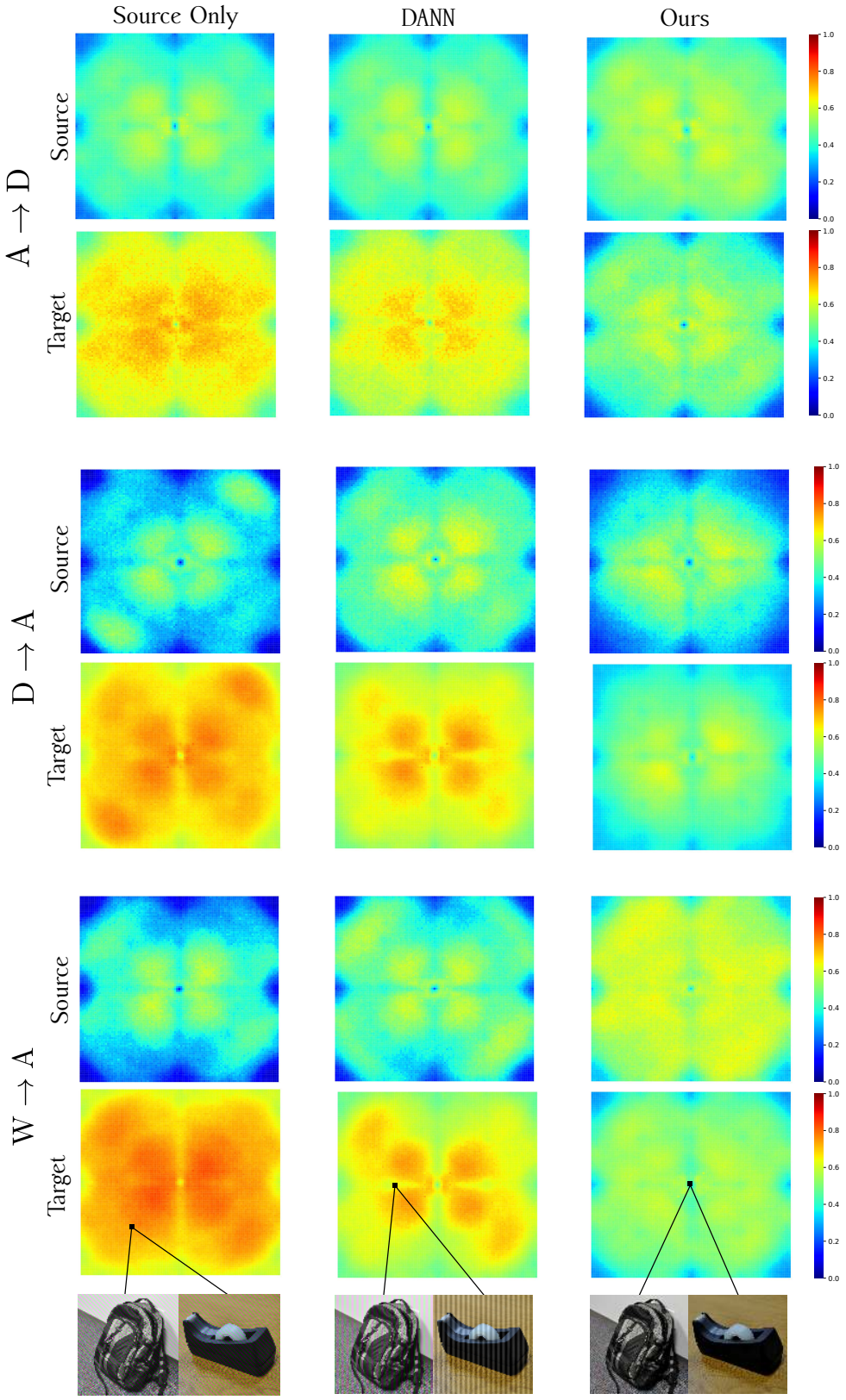


Figure C.1: Fourier Analysis of Model Robustness on Source and Target. An illustration of the Fourier sensitivity heatmaps on the source and target domains for a ResNet-50 trained with different objectives. Each pixel of the heatmap is the error of the model when all of its inputs are perturbed with a single Fourier basis vector.

D Preliminary experiments with Spectral Filtering

Experiments conducted with Thomas Cordier. Victor Bouvier and Thomas Cordier collaborated equally on this project.

Models. For the classification task, we use the publicly available pretrained WideResNet-28-10 (Zagoruyko and Komodakis 2016) of RobustBench (Croce et al. 2021). All methods are evaluated on the same model. During training and at test-time, we update the statistics of the batch normalizations (Ioffe and Szegedy 2015b) of the WideResNet-28-10 following (Wang et al. 2021a; Schneider et al. 2020b; Nado et al. 2020). SpectralFilter is set after the first convolutional layer (`conv1` in RobustBench's implementation (Croce et al. 2021)).

Dataset. The tested model is trained on the training set of CIFAR-10 (Krizhevsky 2009) which is composed of 50,000 pictures with 10 classes. SpectralFilter is evaluated on CIFAR-10-C (Hendrycks and Dietterich 2019b), the test set of 10,000 images of CIFAR-10 (Krizhevsky 2009) augmented by 15 common corruptions alongs 5 levels of severity. CIFAR-10-C is a standard image classification dataset for domain adaptation issues.

Optimization. We optimize the filter parameters λ by Adam Kingma and Ba 2015 for 100 steps on a offline fully test-time adaptation setting. We set the batch size at 200 samples and the learning rate at 10^{-3}.

Implementation. We follow the simple and generic implementation of Wang et al. 2021a based on PyTorch Paszke et al. 2019b and RobustBench Croce et al. 2021.

Baselines. We report baselines for TTA with Batch-Norm Adaptation (Schneider et al. 2020a), TENT (**wang2020fully**), a model which is not adapted (No) and we note Spectral Filter (SF). We also report SF^{++} where L^2 penalization parameter is a matrix, that we note Λ, which is equivalent to the Tikhonov regularization;

$$\theta^\star_\Lambda := \arg\min_{\theta\in\Theta} L_{c,S}(\theta) + ||\Lambda\theta||^2 \tag{D.1}$$

and the Spectral Filtering matrix is;

$$F_\Lambda := \left(\mathbb{E}[X^\top X] + \Lambda^\top\Lambda\right)^{-1}\mathbb{E}[X^\top X] \tag{D.2}$$

In particular, noting P such that $P^\top\mathbb{E}[X^\top X]P$ is diagonal, we enforce $P^\top\Lambda P$ to be a diagonal matrix.

Method	Mean	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
No	43.53	72.33	65.71	72.92	46.94	54.32	34.3	42.02	25.07	41.30	26.01	9.30	46.69	26.59	58.45	30.30
BN	20.44	28.08	26.12	36.27	12.82	35.28	14.17	**12.13**	17.28	17.39	15.26	8.39	12.63	**23.76**	19.66	27.30
TENT	**19.96**	28.05	26.11	36.31	**12.8**	35.28	**14.16**	12.14	**17.27**	**17.36**	**15.23**	**8.37**	**12.59**	23.77	**19.61**	27.31
SF	21.2	26.21	24.08	35.41	16.23	35.37	16.30	14.94	18.10	18.63	17.71	9.04	16.01	25.13	21.03	25.08
SF^{++}	20.35	**25.5**	**23.55**	**33.77**	14.82	**35.04**	15.24	13.76	17.73	17.43	16.09	8.62	14.58	24.44	20.00	**24.68**

Table D.1

E Résumé de la thèse en français

Bien que les systèmes d'apprentissage automatique (ML) occupent une place de plus en plus importante dans l'industrie, ils ne sont pas infaillibles. En effet, ces systèmes sont sensibles au changement de distribution, c'est-à-dire que les données utilisées pour entraîner le système ne sont pas représentatives des données du monde réel, ce qui limite leur déploiement dans des applications critiques. La présente thèse de doctorat contribue à prévenir un tel manque de robustesse des systèmes ML à travers le prisme de l'adaptation, c'est-à-dire la capacité du système à s'adapter rapidement à une situation nouvelle. Dans le but d'apprendre des modèles adaptatifs, nous organisons les contributions de cette thèse en quatre parties.

La partie I de cette thèse présente une rétrospective historique du problème de l'adaptation. Le chapitre 2 a une saveur anthropomorphique en reliant l'étude de Jean Piaget sur le développement de l'intelligence chez l'enfant aux idées fondatrices de l'Intelligence Artificielle (IA) proposées par McCarthy et ses pairs. Alors que l'apprentissage automatique s'impose dans l'IA, nous passons en revue certains échecs classiques de ces systèmes dus à un changement de distribution. Cela motive notre intérêt pour les modèles adaptatifs, c'est-à-dire les modèles qui mettent en œuvre le principe d'adaptation dans des situations où les données sont sujettes à des changements. Dans le chapitre 3, nous effectuons une revue originale des travaux relatifs à l'adaptation dans l'apprentissage automatique, en fournissant le contexte technique nécessaire à cette thèse et la description du problème fondamental de l'apprentissage de représentations invariantes et transférables entre domaines.

La partie II se concentre sur l'analyse théorique de la transférabilité des représentations invariantes du domaine. Dans le chapitre 4, nous introduisons un nouveau terme d'erreur, appelé erreur *Hypothesis Class Reduction*, en nous appuyant sur la théorie séminale de (Ben-David et al. 2010a). Ce nouveau terme traduit le risque de supprimer des informations pertinentes dans la représentation pour atteindre l'invariance du domaine. En particulier, nous démontrons que ce nouveau terme d'erreur fournit le fondement théorique nécessaire à une approche de régularisation influente appelée *Pénalisation Spectrale* (Chen et al. 2019c). Le chapitre 5 unifie deux lignes d'étude importantes : L'échantillonnage d'importance (Quinonero-Candela et al. 2009) et les représentations invariantes au domaine (Ben-David et al. 2007; Ganin and Lempitsky 2015). Cette nouvelle analyse introduit un terme crucial appelé le terme *transferability*, qui quantifie le manque de transférabilité des représentations invariantes entre le domaine source et le domaine cible. Il est important de noter que ce terme dépend de l'étiquette dans le domaine cible, ce qui le rend hors de portée dans le scénario standard de l'adaptation de domaine non-supervisé. Pour contourner cette limitation, nous effectuons une analyse du rôle du biais inductif dans le chapitre 6. En particulier, nous montrons théoriquement que le terme de transférabilité peut être approximé tant qu'un fort biais inductif est disponible, démontrant ainsi l'efficacité d'une pratique bien adoptée consistant à s'appuyer sur les étiquettes prédites de la cible pendant l'adaptation (Long et al. 2018).

La partie III est consacrée à l'investigation empirique des résultats théoriques établis dans la partie II. Dans le chapitre 7, nous développons un algorithme qui s'appuie sur deux biais inductifs faibles, à la fois sur les poids et les représentations, appelé *Robust Domain Adaptation* (RUDA). Notre étude empirique démontre que RUDA améliore considérablement l'adaptation dans le scénario difficile du changement de distributions des étiquettes (Zhao et al. 2019). Dans le chapitre 8, nous étudions le rôle du biais inductif fort qui stipule que la prédiction dans le domaine cible doit rester invariante à la perturbation des entrées qui ne sont pas corrélées à l'étiquette. Ce nouvel algorithme atteint les performances de pointe des méthodes basées sur des représentations invariantes du domaine sur plusieurs repères. Comme un biais inductif aussi fort peut ne pas être disponible dans un plus large éventail d'applications, nous étudions le rôle de l'apprentissage actif pour améliorer l'adaptation dans le chapitre 9. En particulier, nous développons un critère appelé *Sage* qui quantifie le manque de transférabilité d'un échantillon cible. En construisant une requête sur ce critère, nous concevons une stratégie d'annotation qui annote un ensemble diversifié d'échantillons pour lesquels l'adaptation a échoué. Nos expériences démontrent que Sage améliore l'état de l'art en matière d'adaptation de domaine actif.

La partie IV décrit les idées prospectives sur l'apprentissage de modèles adaptatifs avec des représentations invariantes. Le chapitre 10 remet en question l'hypothèse commune de l'adaptation de domaine non supervisée où un grand ensemble d'échantillons peuplent à la fois le domaine source et le domaine cible. À cette fin, nous introduisons le nouveau problème de l'adaptation avec peu d'échantillons, comblant ainsi le fossé entre l'apprentissage et l'adaptation à partir de quelques échantillons. Le chapitre 11 expose les possibilités d'adaptation avec peu d'échantillons.

Bibliography

Afridi, Muhammad Jamal, Arun Ross, and Erik M Shapiro (2018). "On automated source selection for transfer learning in convolutional neural networks". In: *Pattern recognition* 73, pp. 65–75.

Ahuja, Kartik et al. (2020). "Invariant risk minimization games". In: *International Conference on Machine Learning*. PMLR, pp. 145–155.

Aiguier, Marc et al. (2018). "Belief revision, minimal change and relaxation: A general framework based on satisfaction systems, and applications to description logics". In: *Artificial Intelligence* 256, pp. 160–180.

Alchourrón, Carlos E, Peter Gärdenfors, and David Makinson (1985). "On the logic of theory change: Partial meet contraction and revision functions". In: *Journal of symbolic logic*, pp. 510–530.

Amodei, Dario et al. (2016). "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565*.

Arjovsky, Martin (2020). "Out of Distribution Generalization in Machine Learning". PhD thesis. New York University.

– (2021). "Out of Distribution Generalization in Machine Learning". In: *arXiv preprint arXiv:2103.02667*.

Arjovsky, Martín and Léon Bottou (2017). "Towards Principled Methods for Training Generative Adversarial Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=Hk4_qw5xe.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (2017). "Wasserstein gan". In: *arXiv preprint arXiv:1701.07875*.

Arjovsky, Martin et al. (2019). "Invariant Risk Minimization". In: *arXiv preprint arXiv:1907.02893*.

Arthur, David and Sergei Vassilvitskii (2006). *k-means++: The advantages of careful seeding*. Tech. rep. Stanford.

Ash, Jordan T. et al. (2020a). "Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds". In: *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net. URL: https://openreview.net/forum?id=ryghZJBKPS.

– (2020b). "Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=ryghZJBKPS.

Bach, Francis (2021). "Learning Theory from First Principles Draft". In:

Baktashmotlagh, Mahsa et al. (2013). "Unsupervised domain adaptation by domain invariant projection". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 769–776.

Balcan, Maria-Florina, Alina Beygelzimer, and John Langford (2009). "Agnostic active learning". In: *Journal of Computer and System Sciences* 75.1, pp. 78–89.

Bascol, Kevin, Rémi Emonet, and Elisa Fromont (2019). "Improving domain adaptation by source selection". In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3043–3047.

Beery, Sara, Grant Van Horn, and Pietro Perona (2018). "Recognition in terra incognita". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 456–473.

Belkin, Mikhail et al. (2019). "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32, pp. 15849–15854.

Ben-David, Shai et al. (2007). "Analysis of representations for domain adaptation". In: *Advances in neural information processing systems*, pp. 137–144.

Ben-David, Shai et al. (2010a). "A theory of learning from different domains". In: *Machine learning* 79.1-2, pp. 151–175.

Ben-David, Shai et al. (2010b). "Impossibility theorems for domain adaptation". In: *International Conference on Artificial Intelligence and Statistics*, pp. 129–136.

Bengio, Yoshua et al. (2009). "Learning deep architectures for AI". In: *Foundations and trends® in Machine Learning* 2.1, pp. 1–127.

Bennequin, Etienne et al. (2021). "Bridging Few-Shot Learning and Adaptation: New Challenges of Support-Query Shift". In: *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part I*. Ed. by Nuria Oliver et al. Vol. 12975. Lecture Notes in Computer Science. Springer, pp. 554–569. DOI: 10.1007/978-3-030-86486-6_34. URL: https://doi.org/10.1007/978-3-030-86486-6_34.

Bhushan Damodaran, Bharath et al. (2018). "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 447–463.

Blanchard, Gilles, Gyemin Lee, and Clayton Scott (2010). "Semi-supervised novelty detection". In: *The Journal of Machine Learning Research* 11, pp. 2973–3009.

Bottou, Leon et al. (2018). "Geometrical insights for implicit generative modeling". In: *Braverman Readings in Machine Learning. Key Ideas from Inception to Current State*. Springer, pp. 229–268.

Bouchaud, Jean-Philippe and Marc Potters (2009). "Financial applications of random matrix theory: a short review". In: *arXiv preprint arXiv:0910.1205*.

Bousquet, Olivier, Stéphane Boucheron, and Gábor Lugosi (2003). "Introduction to statistical learning theory". In: *Summer school on machine learning*. Springer, pp. 169–207.

Bouvier, Victor et al. (2020a). *Domain-Invariant Representations: A Look on Compression and Weights*. URL: https://openreview.net/forum?id=B1xGxgSYvH.

Bouvier, Victor et al. (2020b). "Robust Domain Adaptation: Representations, Weights and Inductive Bias". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I*. Ed. by Frank Hutter et al. Vol. 12457. Lecture Notes in Computer Science. Springer, pp. 353–377. DOI: 10.1007/978-3-030-67658-2_21. URL: https://doi.org/10.1007/978-3-030-67658-2_21.

Bouvier, Victor et al. (2020c). "Stochastic Adversarial Gradient Embedding for Active Domain Adaptation". In: *arXiv preprint arXiv:2012.01843*.

Bouvier, Victor et al. (2021). "Robust Domain Adaptation: Representations, Weights and Inductive Bias (Extended Abstract)". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Ed. by Zhi-Hua Zhou. ijcai.org, pp. 4750–4754. DOI: 10.24963/ijcai.2021/644. URL: https://doi.org/10.24963/ijcai.2021/644.

Bronskill, John et al. (2020). "Tasknorm: Rethinking batch normalization for meta-learning". In: *ICML*. PMLR.

Brown, Tom B. et al. (2020). "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Bun, Joël, Jean-Philippe Bouchaud, and Marc Potters (2017). "Cleaning large correlation matrices: tools from random matrix theory". In: *Physics Reports* 666, pp. 1–109.

Burns, Collin and Jacob Steinhardt (2021). "Limitations of Post-Hoc Feature Alignment for Robustness". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2525–2533.

Byrd, Jonathon and Zachary Lipton (2019). "What is the effect of importance weighting in deep learning?" In: *International Conference on Machine Learning*. PMLR, pp. 872–881.

Caldas, Sebastian et al. (2018). "Leaf: A benchmark for federated settings". In: *arXiv preprint arXiv:1812.01097*.

Cao, Yue, Mingsheng Long, and Jianmin Wang (2018). "Unsupervised domain adaptation with distribution matching machines". In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Cao, Zhangjie et al. (2018). "Partial adversarial domain adaptation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 135–150.

Carmon, Yair et al. (2019). "Unlabeled data improves adversarial robustness". In: *Advances in Neural Information Processing Systems*, pp. 11190–11201.

Chabot, Florian et al. (2017). "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2040–2049.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar (2009). "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3, pp. 1–58.

Chang, Shiyu et al. (2020). "Invariant rationalization". In: *International Conference on Machine Learning*. PMLR, pp. 1448–1458.

Chapelle, O., B. Scholkopf, and A. Zien, Eds. (2009). "Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews]". In: *IEEE Transactions on Neural Networks* 20.3, pp. 542–542.

Chapelle, Olivier, Bernhard Scholkopf, and Alexander Zien (2009). "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3, pp. 542–542.

Chapelle, Olivier, Vladimir Vapnik, and Jason Westontt (2000). "Transductive Inference for Estimating". In: *Advances in Neural Information Processing Systems 12: Proceedings of the 1999 Conference*. Vol. 1. MIT Press, p. 421.

Chattopadhyay, Rita et al. (2013). "Joint transfer and batch-mode active learning". In: *International Conference on Machine Learning*, pp. 253–261.

Chen, Liang-Chieh et al. (2017). "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *IEEE transactions on pattern analysis and machine intelligence* 40.4, pp. 834–848.

Chen, Wei-Yu et al. (2019a). "A Closer Look at Few-shot Classification". In: *International Conference on Learning Representations*.

Chen, Xinyang et al. (2019b). "Catastrophic Forgetting Meets Negative Transfer: Batch Spectral Shrinkage for Safe Transfer Learning". In: *Advances in Neural Information Processing Systems* 32, pp. 1908–1918.

Chen, Xinyang et al. (2019c). "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation". In: *International Conference on Machine Learning*, pp. 1081–1090.

Chidlovskii, Boris, Stephane Clinchant, and Gabriela Csurka (2016). "Domain adaptation in the absence of source domain data". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 451–460.

Chuang, Ching-Yao, Antonio Torralba, and Stefanie Jegelka (2020). "Estimating Generalization under Distribution Shifts via Domain-Invariant Representations". In: *International Conference on Machine Learning*. PMLR, pp. 1984–1994.

Cohen, Gregory et al. (2017). "EMNIST: Extending MNIST to handwritten letters". In: *IJCNN*. IEEE.

Combes, Remi Tachet des et al. (2020a). "Domain adaptation with conditional distribution matching and generalized label shift". In: *Advances in Neural Information Processing Systems* 33.

Combes, Remi Tachet des et al. (2020b). "Domain Adaptation with Conditional Distribution Matching and Generalized Label Shift". In: *arXiv preprint arXiv:2003.04475*.

Corbett-Davies, Sam and Sharad Goel (2018). "The measure and mismeasure of fairness: A critical review of fair machine learning". In: *arXiv preprint arXiv:1808.00023*.

Corbière, Charles et al. (2019). "Addressing Failure Prediction by Learning Model Confidence". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 2902–2913. URL: http://papers.nips.cc/paper/8556-addressing-failure-prediction-by-learning-model-confidence.pdf.

Cordts, Marius et al. (2016). "The cityscapes dataset for semantic urban scene understanding". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223.

Cortes, Corinna, Yishay Mansour, and Mehryar Mohri (2010). "Learning bounds for importance weighting". In: *Advances in neural information processing systems*, pp. 442–450.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Cortes, Corinna et al. (2008). "Sample selection bias correction theory". In: *International conference on algorithmic learning theory*. Springer, pp. 38–53.

Courty, Nicolas et al. (2016). "Optimal transport for domain adaptation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.9, pp. 1853–1865.

Courty, Nicolas et al. (2017). "Joint distribution optimal transportation for domain adaptation". In: *Advances in Neural Information Processing Systems*, pp. 3730–3739.

Croce, Francesco et al. (2021). *RobustBench: a standardized adversarial robustness benchmark*. arXiv: 2010.09670 [cs.LG].

Cubuk, Ekin D et al. (2018). "Autoaugment: Learning augmentation policies from data". In: *arXiv preprint arXiv:1805.09501*.

– (2019a). "Autoaugment: Learning augmentation strategies from data". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 113–123.

Cubuk, Ekin D et al. (2019b). "RandAugment: Practical data augmentation with no separate search". In: *arXiv preprint arXiv:1909.13719*.

Cuturi, Marco (2013). "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in neural information processing systems* 26, pp. 2292–2300.

Deng, Jia et al. (2009). "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee, pp. 248–255.

Dhillon, Guneet Singh et al. (2020). "A Baseline for Few-Shot Image Classification". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rylXBkrYDS.

Du, Yingjun et al. (2021). "MetaNorm: Learning to Normalize Few-Shot Batches Across Domains". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=9z_dNsC4B5t.

D'Amour, Alexander et al. (2021). "Overlap in observational studies with high-dimensional covariates". In: *Journal of Econometrics* 221.2, pp. 644–654.

Edwards, Harrison and Amos J. Storkey (2016). "Censoring Representations with an Adversary". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1511.05897.

Elsahar, Hady and Matthias Gallé (2019). "To annotate or not? predicting performance drop under domain shift". 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. 2163–2173.

Esteva, Andre et al. (2017). "Dermatologist-level classification of skin cancer with deep neural networks". In: *nature* 542.7639, pp. 115–118.

Ferri, César, José Hernández-Orallo, and R. Modroiu (2009). "An experimental comparison of performance measures for classification". In: *Pattern Recognit. Lett.* 30.1, pp. 27–38. DOI: 10.1016/j.patrec.2008.08.010. URL: https://doi.org/10.1016/j.patrec.2008.08.010.

Feutry, Clément et al. (2018). "Learning anonymized representations with adversarial neural networks". In: *arXiv preprint arXiv:1802.09386*.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International Conference on Machine Learning*. PMLR, pp. 1126–1135.

Friedman, Jerome H (2001). "Greedy function approximation: a gradient boosting machine". In: *Annals of statistics*, pp. 1189–1232.

Fu, Bo et al. (2021). "Transferable Query Selection for Active Domain Adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7272–7281.

Galanti, Tomer and Lior Wolf (2020). "On the Modularity of Hypernetworks". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. URL: https://proceedings.neurips.cc/paper/2020/hash/75c58d36157505a600e0695ed0b3a22d-Abstract.html.

Gama, Joao (2010). *Knowledge discovery from data streams*. CRC Press.

Gammerman, Alex, Volodya Vovk, and Vladimir Vapnik (2013). "Learning by transduction". In: *arXiv preprint arXiv:1301.7375*.

Ganin, Yaroslav and Victor Lempitsky (2015). "Unsupervised Domain Adaptation by Backpropagation". In: *International Conference on Machine Learning*, pp. 1180–1189.

Ganin, Yaroslav et al. (2016). "Domain-adversarial training of neural networks". In: *The Journal of Machine Learning Research* 17.1, pp. 2096–2030.

Gao, Mingfei et al. (2020). "Consistency-based semi-supervised active learning: Towards minimizing labeling cost". In: *European Conference on Computer Vision*. Springer, pp. 510–526.

Geiger, Mario et al. (2020). "Scaling description of generalization with number of parameters in deep learning". In: *Journal of Statistical Mechanics: Theory and Experiment* 2020.2, p. 023401.

Geirhos, Robert et al. (2019). "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=Bygh9j09KX.

Germain, Pascal et al. (2013). "A PAC-Bayesian approach for domain adaptation with specialization to linear classifiers". In: *International conference on machine learning*. PMLR, pp. 738–746.

– (2016). "A new PAC-Bayesian perspective on domain adaptation". In: *International conference on machine learning*. PMLR, pp. 859–868.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in neural information processing systems* 27.

Grandvalet, Yves and Yoshua Bengio (2004). "Semi-supervised learning by entropy minimization". In: *Advances in neural information processing systems* 17, pp. 529–536.

– (2005). "Semi-supervised learning by entropy minimization". In: *Advances in neural information processing systems*, pp. 529–536.

Gretton, Arthur et al. (2009). "Covariate shift by kernel mean matching". In: *Dataset shift in machine learning* 3.4, p. 5.

Gretton, Arthur et al. (2012). "A kernel two-sample test". In: *Journal of Machine Learning Research* 13.Mar, pp. 723–773.

Gulrajani, Ishaan and David Lopez-Paz (2021). "In Search of Lost Domain Generalization". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=lQdXeXDoWtI.

Gulrajani, Ishaan et al. (2017). "Improved Training of Wasserstein GANs". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 5767–5777. URL: https://proceedings.neurips.cc/paper/2017/hash/892c3b1c6dccd52936e27cbd0ff683d6-Abstract.html.

Ha, David, Andrew Dai, and Quoc V Le (2016). "Hypernetworks". In: *arXiv preprint arXiv:1609.09106*.

Hanneke, Steve et al. (2014). "Theory of disagreement-based active learning". In: *Foundations and Trends® in Machine Learning* 7.2-3, pp. 131–309.

Hastie, Trevor et al. (2019). "Surprises in high-dimensional ridgeless least squares interpolation". In: *arXiv preprint arXiv:1903.08560*.

He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hendrycks, Dan and Thomas Dietterich (2019a). "Benchmarking neural network robustness to common corruptions and perturbations". In: *arXiv preprint arXiv:1903.12261*.

– (2019b). "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=HJz6tiCqYm.

Hendrycks, Dan et al. (2020). "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=S1gmrxHFvB.

Hernández-Orallo, José (2017). "Evaluation in artificial intelligence: from task-oriented to ability-oriented measurement". In: *Artif. Intell. Rev.* 48.3, pp. 397–447. DOI: 10.1007/s10462-016-9505-7. URL: https://doi.org/10.1007/s10462-016-9505-7.

Ho, Tin Kam (1995). "Random decision forests". In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE, pp. 278–282.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hofmann, Thomas, Bernhard Schölkopf, and Alexander J Smola (2008). "Kernel methods in machine learning". In: *The annals of statistics*, pp. 1171–1220.

Hsu, Wei-Ning and Hsuan-Tien Lin (2015). "Active learning by learning". In: *Twenty-Ninth AAAI conference on artificial intelligence*. Citeseer.

Hu, Yuqing, Vincent Gripon, and Stéphane Pateux (2020). "Leveraging the feature distribution in transfer-based few-shot learning". In: *arXiv preprint arXiv:2006.03806*.

Huang, Jiayuan et al. (2007). "Correcting sample selection bias by unlabeled data". In: *Advances in neural information processing systems*, pp. 601–608.

Ioffe, Sergey and Christian Szegedy (2015a). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR, pp. 448–456.

– (2015b). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML'15. Lille, France: JMLR.org, 448–456.

Isola, Phillip et al. (2017). "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Jing, Longlong and Yingli Tian (2020). "Self-supervised visual feature learning with deep neural networks: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Johansson, Fredrik, David Sontag, and Rajesh Ranganath (2019). "Support and Invertibility in Domain-Invariant Representations". In: *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 527–536.

Kannan, Harini, Alexey Kurakin, and Ian J. Goodfellow (2018). "Adversarial Logit Pairing". In: *CoRR* abs/1803.06373. arXiv: 1803.06373. URL: http://arxiv.org/abs/1803.06373.

Kifer, Daniel, Shai Ben-David, and Johannes Gehrke (2004). "Detecting change in data streams". In: *VLDB*. Vol. 4. Toronto, Canada, pp. 180–191.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980.

Knight, Philip A (2008). "The Sinkhorn–Knopp algorithm: convergence and applications". In: *SIAM Journal on Matrix Analysis and Applications* 30.1, pp. 261–275.

Koh, Pang Wei et al. (2020). "Concept bottleneck models". In: *International Conference on Machine Learning*. PMLR, pp. 5338–5348.

Krizhevsky, Alex (2009). "Learning multiple layers of features from tiny images". In: Technical report, University of Toronto.

Krizhevsky, Alex, Geoffrey Hinton, et al. (2009). "Learning multiple layers of features from tiny images". In:

Krueger, David et al. (2021). "Out-of-distribution generalization via risk extrapolation (rex)". In: *International Conference on Machine Learning*. PMLR, pp. 5815–5826.

Laenen, Steinar and Luca Bertinetto (2020). "On Episodes, Prototypical Networks, and Few-shot Learning". In: *arXiv preprint arXiv:2012.09831*.

Laloux, Laurent et al. (2000). "Random matrix theory and financial correlations". In: *International Journal of Theoretical and Applied Finance* 3.03, pp. 391–397.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *nature* 521.7553, pp. 436–444.

LeCun, Yann et al. (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551.

LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Li, Fengpei, Henry Lam, and Siddharth Prusty (2020). "Robust importance weighting for covariate shift". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 352–362.

Liang, Jian, Dapeng Hu, and Jiashi Feng (2020). "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation". In: *International Conference on Machine Learning*. PMLR, pp. 6028–6039.

Lim, Sungbin et al. (2019). "Fast autoaugment". In: *Advances in Neural Information Processing Systems*, pp. 6662–6672.

Lipton, Zachary, Yu-Xiang Wang, and Alexander Smola (2018). "Detecting and correcting for label shift with black box predictors". In: *International conference on machine learning*. PMLR, pp. 3122–3130.

Litjens, Geert et al. (2017). "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42, pp. 60–88.

Liu, Hong et al. (2019a). "Transferable Adversarial Training: A General Approach to Adapting Deep Classifiers". In: *International Conference on Machine Learning*, pp. 4013–4022.

Liu, Yanbin et al. (2019b). "Learning to Propagate Labels: Transductive Propagation Network for Few-Shot Learning". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=SyVuRiC5K7.

Long, Mingsheng et al. (2015). "Learning transferable features with deep adaptation networks". In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*. JMLR. org, pp. 97–105.

Long, Mingsheng et al. (2017). "Deep transfer learning with joint adaptation networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2208–2217.

Long, Mingsheng et al. (2018). "Conditional adversarial domain adaptation". In: *Advances in Neural Information Processing Systems*, pp. 1640–1650.

Lopez-Paz, David et al. (2017). "Discovering causal signals in images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6979–6987.

Lorraine, Jonathan and David Duvenaud (2018). "Stochastic hyperparameter optimization through hypernetworks". In: *arXiv preprint arXiv:1802.09419*.

Louizos, Christos et al. (2016). "The Variational Fair Autoencoder". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1511.00830.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov, pp. 2579–2605.

Maddox, Wesley et al. (2021). "Fast Adaptation with Linearized Neural Networks". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2737–2745.

Madras, David et al. (2018). "Learning adversarially fair and transferable representations". In: *International Conference on Machine Learning*. PMLR, pp. 3384–3393.

Madry, Aleksander et al. (2018). "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=rJzIBfZAb.

Mansour, Yishay, Mehryar Mohri, and Afshin Rostamizadeh (2009). "Domain adaptation: Learning bounds and algorithms". In: *22nd Conference on Learning Theory, COLT 2009*.

Marcus, Gary (2020). "The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence". In: *arXiv preprint arXiv:2002.06177*.

Markou, Markos and Sameer Singh (2003). "Novelty detection: a review—part 1: statistical approaches". In: *Signal processing* 83.12, pp. 2481–2497.

McCarthy, J, ML Minsky, and N Rochester (1955). "A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE". In:

McCarthy, John (1959). "Programs with Common Sense". In: *Proceedings of the Teddington Conference on the Mechanization of Thought Processes. Her Majesty's Stationery Office, London.*

Mei, Song and Andrea Montanari (2019). "The Generalization Error of Random Features Regression: Precise Asymptotics and the Double Descent Curve". In: *Communications on Pure and Applied Mathematics*.

Miller, John et al. (2020). "The effect of natural distribution shift on question answering models". In: *International Conference on Machine Learning*. PMLR, pp. 6905–6916.

Mirza, Mehdi and Simon Osindero (2014). "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784*.

Miyato, Takeru et al. (2018). "Virtual adversarial training: a regularization method for supervised and semi-supervised learning". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8, pp. 1979–1993.

Moyer, Daniel et al. (2018). "Invariant representations without adversarial training". In: *Advances in Neural Information Processing Systems*, pp. 9084–9093.

Muandet, Krikamol et al. (2016). "Kernel mean embedding of distributions: A review and beyond". In: *arXiv preprint arXiv:1605.09522*.

Müller, Alfred (1997). "Integral probability metrics and their generating classes of functions". In: *Advances in Applied Probability* 29.2, pp. 429–443.

Nado, Zachary et al. (2020). *Evaluating Prediction-Time Batch Normalization for Robustness under Covariate Shift*. arXiv: 2006.10963.

Novak, Roman et al. (2018). "Sensitivity and Generalization in Neural Networks: an Empirical Study". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=HJC2SzZCW.

Odena, Augustus, Christopher Olah, and Jonathon Shlens (2017). "Conditional image synthesis with auxiliary classifier gans". In: *International conference on machine learning*. PMLR, pp. 2642–2651.

Oswald, Johannes von et al. (2019). "Continual learning with hypernetworks". In: *International Conference on Learning Representations*.

Ouali, Yassine, Céline Hudelot, and Myriam Tami (2020). "An overview of deep semi-supervised learning". In: *arXiv preprint arXiv:2006.05278*.

Ouali, Yassine et al. (2020). "Target Consistency for Domain Adaptation: when Robustness meets Transferability". In: *arXiv preprint arXiv:2006.14263*.

Pan, Sinno Jialin and Qiang Yang (2009). "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10, pp. 1345–1359.

Panareda Busto, Pau and Juergen Gall (2017). "Open set domain adaptation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 754–763.

Parisi, German I et al. (2019). "Continual lifelong learning with neural networks: A review". In: *Neural Networks* 113, pp. 54–71.

Paszke, Adam et al. (2019a). "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems*, pp. 8024–8035.

Paszke, Adam et al. (2019b). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Pearl, Judea (2009). *Causality*. Cambridge university press.

Peng, Xingchao et al. (2017). "Visda: The visual domain adaptation challenge". In: *arXiv preprint arXiv:1710.06924*.

Peng, Xingchao et al. (2019). "Moment matching for multi-source domain adaptation". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415.

Peters, Jonas, Peter Bühlmann, and Nicolai Meinshausen (2016). "Causal inference by using invariant prediction: identification and confidence intervals". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78.5, pp. 947–1012.

Peyré, Gabriel, Marco Cuturi, et al. (2019). "Computational Optimal Transport: With Applications to Data Science". In: *Foundations and Trends® in Machine Learning* 11.5-6, pp. 355–607.

Piaget, Jean (1936). *Origins of intelligence in the child.* London: Routledge Kegan Paul.

Prabhu, Viraj et al. (2020). "Active Domain Adaptation via Clustering Uncertainty-weighted Embeddings". In: *arXiv preprint arXiv:2010.08666*.

Quinonero-Candela, Joaquin et al. (2009). *Dataset shift in machine learning*. The MIT Press.

Rabanser, Stephan, Stephan Günnemann, and Zachary Lipton (2019). "Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift". In: *Advances in Neural Information Processing Systems* 32, pp. 1396–1408.

Radford, Alec, Luke Metz, and Soumith Chintala (2016). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1511.06434.

Rai, Piyush et al. (2010). "Domain adaptation meets active learning". In: *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*. Association for Computational Linguistics, pp. 27–32.

Ratner, Alexander J et al. (2016). "Data programming: Creating large training sets, quickly". In: *Advances in neural information processing systems* 29, pp. 3567–3575.

Recht, Benjamin et al. (2019). "Do imagenet classifiers generalize to imagenet?" In: *International Conference on Machine Learning*. PMLR, pp. 5389–5400.

Redko, I (2015). "Nonnegative matrix factorization for unsupervised transfer learning". PhD thesis. PhD thesis, Paris North University.

Redko, Ievgen, Amaury Habrard, and Marc Sebban (2017). "Theoretical analysis of domain adaptation with optimal transport". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 737–753.

– (2019). "On the analysis of adaptability in multi-source domain adaptation". In: *Machine Learning* 108.8, pp. 1635–1652.

Redko, Ievgen et al. (2019). *Advances in domain adaptation theory*. Elsevier.
– (2020). "A survey on domain adaptation theory". In: *arXiv preprint arXiv:2004.11829*.
Ren, Mengye et al. (2018). "Meta-learning for semi-supervised few-shot classification". In: *arXiv preprint arXiv:1803.00676*.
Richter, Stephan R et al. (2016). "Playing for data: Ground truth from computer games". In: *European conference on computer vision*. Springer, pp. 102–118.
Rosenberg, Jarrett (1980). "Piaget and Artificial Intelligence." In: *AAAI*, pp. 266–268.
Roth, Dan and Kevin Small (2006). "Margin-based active learning for structured output spaces". In: *European Conference on Machine Learning*. Springer, pp. 413–424.
Russell, Stuart J. and Peter Norvig (2020). *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson. ISBN: 9780134610993. URL: http://aima.cs.berkeley.edu/.
Saenko, Kate et al. (2010). "Adapting visual category models to new domains". In: *European conference on computer vision*. Springer, pp. 213–226.
Saha, Avishek et al. (2011). "Active supervised domain adaptation". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 97–112.
Sahoo, Doyen et al. (2019). *Meta-Learning with Domain Adaptation for Few-Shot Learning under Domain Shift*.
Saito, Kuniaki et al. (2019). "Semi-supervised domain adaptation via minimax entropy". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8050–8058.
Salimans, Tim et al. (2016). "Improved techniques for training gans". In: *Advances in neural information processing systems* 29, pp. 2234–2242.
Sample, Ian (2020). "DeepMind AI cracks 50-year-old problem of protein folding". In: *The Guardian*. URL: /www.theguardian.com/technology/2020/nov/30/deepmind-ai-cracks-50-year-old-problem-of-biology-research.
Sanderson, Tyler and Clayton Scott (2014). "Class proportion estimation with application to multiclass anomaly rejection". In: *Artificial Intelligence and Statistics*. PMLR, pp. 850–858.
Sankaranarayanan, Swami et al. (2018). "Generate to adapt: Aligning domains using generative adversarial networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8503–8512.
Sato, Ikuro, Hiroki Nishimura, and Kensuke Yokoi (2015). "Apac: Augmented pattern classification with neural networks". In: *arXiv preprint arXiv:1505.03229*.
Schmidhuber, Jurgen, U Meier, and D Ciresan (2012). "Multi-column deep neural networks for image classification". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, pp. 3642–3649.
Schneider, Steffen et al. (2020a). "Improving robustness against common corruptions by covariate shift adaptation". In: *Advances in Neural Information Processing Systems* 33.
– (2020b). "Improving robustness against common corruptions by covariate shift adaptation". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 11539–11551. URL: https://proceedings.neurips.cc/paper/2020/file/85690f81aadc1749175c187784afc9ee-Paper.pdf.
Scott, Clayton, Gilles Blanchard, and Gregory Handy (2013). "Classification with asymmetric label noise: Consistency and maximal denoising". In: *Conference on learning theory*. PMLR, pp. 489–511.
Seddik, Mohamed El Amine (2020). "Random Matrix Theory for AI: From Theory to Practice". PhD thesis. Centrale Supélec.

Sener, Ozan and Silvio Savarese (2018). "Active Learning for Convolutional Neural Networks: A Core-Set Approach". In: *International Conference on Learning Representations*.

Settles, Burr (2009). *Active learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences.

Shankar, Vaishaal et al. (2020). "Evaluating machine accuracy on imagenet". In: *International Conference on Machine Learning*. PMLR, pp. 8634–8644.

Shen, Jian et al. (2018). "Wasserstein Distance Guided Representation Learning for Domain Adaptation". In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shimodaira, Hidetoshi (2000). "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Shu, Rui et al. (2018). "A DIRT-T Approach to Unsupervised Domain Adaptation". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=H1q-TM-AW.

Shwartz-Ziv, Ravid and Amitai Armon (2021). "Tabular Data: Deep Learning is Not All You Need". In: *arXiv preprint arXiv:2106.03253*.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587, p. 484.

Silver, David et al. (2017). "Mastering chess and shogi by self-play with a general reinforcement learning algorithm". In: *arXiv preprint arXiv:1712.01815*.

Simard, Patrice Y, David Steinkraus, John C Platt, et al. (2003). "Best practices for convolutional neural networks applied to visual document analysis." In: *Icdar*. Vol. 3. 2003.

Sinz, Fabian H et al. (2007). "An Analysis of Inference with the Universum." In: *NIPS*. Vol. 7, pp. 1–1.

Snell, Jake, Kevin Swersky, and Richard Zemel (2017). "Prototypical networks for few-shot learning". In: *Advances in Neural Information Processing Systems*, pp. 4077–4087.

Sohn, Kihyuk et al. (2020). "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence". In: *Advances in Neural Information Processing Systems* 33.

Su, Jong-Chyi et al. (2020). "Active adversarial domain adaptation". In: *The IEEE Winter Conference on Applications of Computer Vision*, pp. 739–748.

Sugiyama, Masashi, Matthias Krauledat, and Klaus-Robert MÃžller (2007). "Covariate shift adaptation by importance weighted cross validation". In: *Journal of Machine Learning Research* 8.May, pp. 985–1005.

Sugiyama, Masashi et al. (2008). "Direct importance estimation with model selection and its application to covariate shift adaptation". In: *Advances in neural information processing systems*, pp. 1433–1440.

Sun, Baochen, Jiashi Feng, and Kate Saenko (2016). "Return of frustratingly easy domain adaptation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1.

Sun, Baochen and Kate Saenko (2016). "Deep coral: Correlation alignment for deep domain adaptation". In: *European conference on computer vision*. Springer, pp. 443–450.

Sun, Yu et al. (2020). "Test-time training with self-supervision for generalization under distribution shifts". In: *International Conference on Machine Learning*. PMLR, pp. 9229–9248.

Tarvainen, Antti and Harri Valpola (2017). "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in neural information processing systems*, pp. 1195–1204.

Tishby, Naftali, Fernando C Pereira, and William Bialek (2000). "The information bottleneck method". In: *arXiv preprint physics/0004057*.

Torralba, Antonio, Alexei A Efros, et al. (2011). "Unbiased look at dataset bias." In: *CVPR*. Vol. 1. 2. Citeseer, p. 7.

Triantafillou, Eleni et al. (2019). "Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples". In: *International Conference on Learning Representations*.

Tsai, Yi-Hsuan et al. (2018). "Learning to adapt structured output space for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7472–7481.

Turing, Alan (1948). "Intelligent Machinery". In: *The Essential Turing*. Oxford University Press.

Vapnik, VetASTERIN and Alexander Sterin (1977). "On structural risk minimization or overall risk in a problem of pattern recognition". In: *Automation and Remote Control* 10.3, pp. 1495–1503.

Vapnik, Vladimir (2013). *The nature of statistical learning theory*. Springer science & business media.

Vapnik, VN and A Ya Chervonenkis (1971). "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". In: *Theory of Probability & Its Applications* 16.2, pp. 264–280.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

Venkateswara, Hemanth et al. (2017). "Deep hashing network for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027.

Villani, Cédric et al. (2018). *Donner un sens à l'intelligence artificielle: pour une stratégie nationale et européenne*. Conseil national du numérique.

Vinyals, Oriol et al. (2016). "Matching Networks for One Shot Learning". In: *Advances in Neural Information Processing Systems* 29, pp. 3630–3638.

Vu, Tuan-Hung et al. (2019). "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2517–2526.

Wan, Li et al. (2013). "Regularization of neural networks using dropconnect". In: *International conference on machine learning*, pp. 1058–1066.

Wang, Dan and Yi Shang (2014). "A new active labeling method for deep learning". In: *2014 International joint conference on neural networks (IJCNN)*. IEEE, pp. 112–119.

Wang, Dequan et al. (2021a). "Tent: Fully Test-Time Adaptation by Entropy Minimization". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=uXl3bZLkr3c.

Wang, Dequan et al. (2021b). "Tent: Fully Test-Time Adaptation by Entropy Minimization". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=uXl3bZLkr3c.

Wang, Ximei et al. (2019). "Transferable Normalization: Towards Improving Transferability of Deep Neural Networks". In: *Advances in Neural Information Processing Systems* 32, pp. 1953–1963.

Wang, Xuezhi, Tzu-Kuo Huang, and Jeff Schneider (2014). "Active transfer learning under model shift". In: *International Conference on Machine Learning*. PMLR, pp. 1305–1313.

Wu, Yifan et al. (2019). "Domain Adaptation with Asymmetrically-Relaxed Distribution Alignment". In: *International Conference on Machine Learning*, pp. 6872–6881.

Xie, Qizhe et al. (2017). "Controllable invariance through adversarial feature learning". In: *Advances in Neural Information Processing Systems*, pp. 585–596.

Xie, Qizhe et al. (2020). "Unsupervised Data Augmentation for Consistency Training". In: *Advances in Neural Information Processing Systems* 33.

Yadav, Chhavi and Leon Bottou (2019). "Cold Case: The Lost MNIST Digits". In: *Advances in Neural Information Processing Systems* 32, pp. 13443–13452.

Yin, Dong et al. (2019). "A fourier perspective on model robustness in computer vision". In: *Advances in Neural Information Processing Systems*, pp. 13255–13265.

You, Kaichao et al. (2019). "Universal domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2720–2729.

Zagoruyko, Sergey and Nikos Komodakis (2016). "Wide Residual Networks". In: *BMVC*.

Zemel, Rich et al. (2013). "Learning fair representations". In: *International Conference on Machine Learning*, pp. 325–333.

Zhang, Dinghuai et al. (2021a). "Can Subnetwork Structure Be the Key to Out-of-Distribution Generalization?" In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 12356–12367. URL: http://proceedings.mlr.press/v139/zhang21a.html.

Zhang, Jing et al. (2018). "Importance weighted adversarial nets for partial domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8156–8164.

Zhang, Kun et al. (2013). "Domain adaptation under target and conditional shift". In: *International Conference on Machine Learning*, pp. 819–827.

Zhang, Marvin et al. (2021b). "Adaptive Risk Minimization: A Meta-Learning Approach for Tackling Group Shift". In: *ICLR*.

Zhang, Yuchen et al. (2019). "Bridging theory and algorithm for domain adaptation". In: *International Conference on Machine Learning*. PMLR, pp. 7404–7413.

Zhao, An et al. (2020). "Domain-Adaptive Few-Shot Learning". In: *arXiv preprint arXiv:2003.08626*.

Zhao, Han et al. (2018). "Adversarial multiple source domain adaptation". In: *Advances in neural information processing systems* 31, pp. 8559–8570.

Zhao, Han et al. (2019). "On Learning Invariant Representations for Domain Adaptation". In: *International Conference on Machine Learning*, pp. 7523–7532.

Zheng, Stephan et al. (2016). "Improving the robustness of deep neural networks via stability training". In: *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 4480–4488.

Zhou, Aurick and Sergey Levine (2021). "Amortized Conditional Normalized Maximum Likelihood: Reliable Out of Distribution Uncertainty Estimation". In: *International Conference on Machine Learning*. PMLR, pp. 12803–12812.

Zhou, Dengyong et al. (2004). "Learning with local and global consistency". In: *Advances in neural information processing systems*, pp. 321–328.