Théorie et algorithmes pour l'adaptation de domaine en apprentissage profond : application à la vision par ordinateur

Rodrigue Siry

Normandie Université

UNIVERSITÉ CAEN NORMANDIE

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité INFORMATIQUE

Préparée au sein de l'Université de Caen Normandie

Théorie et algorithmes pour l'adaptation de domaine en apprentissage profond : application à la vision par ordinateur

Présentée et soutenue par
RODRIGUE SIRY

Thèse soutenue le 12/12/2022 devant le jury composé de		
M. AMAURY HABRARD	Professeur des universités, Université Saint-Etienne Jean Monnet	Rapporteur du jury
MME CÉLINE HUDELOT	Professeur des universités, 35 SUPELEC	Rapporteur du jury
M. FARID OUDYI	Expert, Safran	Membre du jury
M. LOIC SIMON	Maître de conférences, Université de Caen Normandie	Membre du jury Co-encadrant
M. SAMIA AINOUZ	Professeur des universités, INSA Rouen Normandie	Président du jury

Thèse dirigée par FREDERIC JURIE (Groupe de recherche en informatique, image, automatique et instrumentation)

MIIΣ

GREYC
Laboratoire de recherche en sciences du numérique

Résumé

Abstract

The following manuscript tackles the problem of domain adaptation, which is a sub-problem of machine learning where train and test data are biased in some way. More specifically, we focus on the case of deep models applied to image classification problems. We first propose several contributions to the existing domain adaptation theory and conduct a critical analysis showing its limits, as well as those of existing practical algorithms. We then show that the notion of inductive bias plays a crucial role in transfer learning problems. Finally, taking this notion into account, we explore various alternatives that leverage results from related literatures such as meta-learning, pre-training or learning of disentangled representations.

Résumé

Cette thèse porte sur l'adaptation de domaine : un sous-problème de l'apprentissage automatique où les données d'évaluation diffèrent qualitativement des données d'entraînement. Nous nous intéressons plus spécifiquement à son application aux problèmes de classification d'image avec des modèles profonds. Nous commencerons d'abord par proposer une série de contributions à la théorie de l'adaptation de domaine, nous mènerons ensuite une analyse critique montrant ses insuffisances, ainsi que celles des algorithmes pratiques existants. Nous montrerons ensuite que la notion de biais inductif joue un rôle central dans les problèmes de transfert. Enfin, en prenant en compte cette notion, nous explorerons diverses alternatives intégrant des éléments des littératures voisines du méta-apprentissage, du pré-entraînement et de l'apprentissage de représentations désentrelacées.

Abréviations utilisées

Abréviation	Signification
CNN	Convolutional Neural Network
SO	Source-Only
TO	Target-Only
UDA	Unsupervised Domain Adaptation
SSDA	Semi-Supervised Domain Adaptation
DG	Domain Generalization
SS-DG	Single-Source Domain Generalization
MS-DG	Multi-Source Domain Generalization
GAN	Generative Adversarial Network
BiGAN	Bidirectional GAN
VAE	Variational Autoencoder
AAE	Adversarial Autoencoder
GRL	Gradient Reversal Layer
PR	Precision-Recall
ROC	Receiver Operating Characteristic
KNN	K-Nearest Neighbors
SGD	Stochastic Gradient Descent
ELBO	Evidence Lower Bound

Table des matières

Chapitre 1

Introduction

La vision par ordinateur est un domaine en plein essor : déjà utilisée depuis longtemps pour reconnaître les chiffres manuscrits sur les chèques bancaires, on lui trouve maintenant d'innombrables applications telles que la reconnaissance faciale, la classification de tumeurs cancéreuses, la prévention de catastrophes naturelles ou encore la conduite autonome.

Toutefois, permettre à un ordinateur de reconnaître des objets n'est pas chose aisée : par exemple, distinguer des photos de chiens de photos de chats n'est pas un problème que l'on peut naturellement traduire en équation puis résoudre numériquement. Pour résoudre cette famille de problèmes, on préfère avoir recours au paradigme de l'apprentissage machine : au lieu d'établir explicitement un ensemble de règles définissant la nature d'une image de chien ou de chat, on laisse un modèle apprendre ces règles à partir d'un certain nombre d'exemples. Autrefois utilisées avec des modèles de complexité limitée, les méthodes d'apprentissage ont connu un développement spectaculaire avec l'arrivée des réseaux convolutifs profonds [O'Shea and Nash, 2015]. Ces modèles ont en l'effet l'avantage de pouvoir apprendre des fonctions de classification arbitrairement complexes en exploitant jusqu'à plusieurs millions d'exemples. Leur architecture est par ailleurs adaptée aux symétries inhérentes à la structure des images ce qui facilite la généralisation des concepts visuels appris. Au cours de la décennie passée, les modèles d'apprentissage profond ont continué à gagner en taille, en complexité et bien entendu en performance. AlexNet [Krizhevsky et al., 2012] fut le premier modèle d'apprentissage profond à gagner le concours de reconnaissance d'image *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) en 2012, il était alors entraîné sur un million d'images annotées. À titre de comparaison, les derniers modèles de vision en date sont pré-entraînés sur près de 5 milliards d'images.

Cependant, pour atteindre ces performances inégalées, les modèles d'apprentissage profond ont besoin d'exemples annotés en quantités importantes, ce qui est un obstacle à leur déploiement pour de nombreuses applications pratiques. Récolter une grande quantité d'exemples annotés est généralement fastidieux ; cela peut même être coûteux lorsque l'annotation nécessite une certaine expertise (par exemple en imagerie médicale) voire dans certains cas totalement impossible (par exemple des clichés d'animaux en voie de disparition).

Lorsque entraînés avec trop peu d'exemples, ou bien avec des exemples ne représentant pas exactement le cadre d'application ultérieur, les modèles d'apprentissage profond risquent de se spécialiser sur les données d'entraînement sans saisir les généralités de la tâche de classification et peuvent par conséquent avoir un comportement arbitraire sur les données de test. Un effort de recherche massif est donc logiquement dédié à l'assouplissement ces conditions expérimentales exigeantes : on pense par exemple au « few-shot learning », qui désigne l'ensemble des méthodes permettant d'entraîner des modèles profonds à partir d'un faible nombre d'exemples, ou encore à l'adaptation de domaine, qui est l'objet des travaux de cette thèse.

L'adaptation de domaine suppose que les conditions d'apprentissage ne sont pas idéales pour la tâche d'intérêt, le « domaine cible ». Cela signifie que l'on ne dispose pas de nombreuses images annotées représentatives de la tâche que l'on souhaite faire résoudre à notre modèle. On suppose, en revanche, que l'on dispose d'un ensemble d'exemples ayant un lien fort avec la tâche d'intérêt, mais pouvant avoir quelques différences qualitatives, on appelle cet ensemble « domaine source ». Imaginons par exemple que l'on souhaite détecter et classer différentes catégories de panneaux de signalisation en pleine nuit, mais que l'on ne dispose pas d'exemples annotés dans ces conditions. Si on dispose en revanche d'une base d'exemples annotés de ces mêmes panneaux, mais photographiés en plein jour (ce qui est probablement plus facile à trouver), alors une méthode d'adaptation de domaine permettra de tirer parti des connaissances apprises sur les images de jour et de les transférer au cas des images de nuit.

On définit maintenant rigoureusement les notions de *domaine*, de *transfert de domaine* et de *dérive de domaine*. On introduit par la même occasion le formalisme de base qui sera utilisé dans tout le reste du manuscrit. Ce formalisme est pour l'essentiel repris de Redko et al. [2020], du fait de sa généralité. D'autres notations, plus spécifiques, pourront ensuite s'y ajouter.

Définition 1. *(Domaine) Soit X l'espace des images et Y l'espace des labels. Un domaine $\mathcal{D}$ est une distribution définie sur $X \times Y$. On note par ailleurs $\mathcal{D}_X$ et $\mathcal{D}_Y$ les distributions marginales de*

$\mathcal{D}$ sur X (resp. Y). Enfin, on note $\mathcal{D}_{Y|X}$ et $\mathcal{D}_{X|Y}$ les distributions conditionnelles que l'on peut définir à partir de $\mathcal{D}$.

Définition 2. *(Transfert de domaine) Notons $\mathcal{S}$ et $\mathcal{T}$ deux domaines appelés domaine source et domaine cible. Notons $f_T : X \to Y$ la tâche de classification définie sur le domaine cible à partir de $\mathcal{T}_{Y|X}$. Le but de l'apprentissage par transfert est de tirer parti de l'information connue sur $\mathcal{S}$ pour améliorer l'apprentissage de f_T.*

Dans le cas général, $\mathcal{S} \neq \mathcal{T}$. On qualifie d'écart, de dérive ou de shift de domaine la différence existant entre $\mathcal{S}$ et $\mathcal{T}$. Il existe une taxonomie des différents transferts récurrente dans la littérature, établie par-rapport à la nature de cette dérive.

1. $\mathcal{S}_X \neq \mathcal{T}_X$ et $\mathcal{S}_{Y|X} = \mathcal{T}_{Y|X}$
   Dans ce cas, la tâche à accomplir est la même dans les deux domaines, mais la nature des données d'entrée change d'un domaine à l'autre. Ce cas correspond à l'exemple que nous avions donné en introduction : la détection de panneaux de signalisation de jour (domaine source) et la détection de ces mêmes panneaux de nuit (domaine cible). On appelle ce type de dérive de domaine **covariate shift** et on appelle le transfert associé **transfert transductif**.
2. $\mathcal{S}_X = \mathcal{T}_X$ et $\mathcal{S}_{Y|X} \neq \mathcal{T}_{Y|X}$
   Dans ce cas, les données d'entrée sont les mêmes dans les deux domaines, mais la tâche à accomplir change d'un domaine à l'autre. Ce cas correspondrait par exemple à un réajustement du niveau d'alarme dans un système de détection de séisme. On appelle ce type de dérive de domaine **concept shift** et on appelle le transfert associé **transfert inductif**.
3. $\mathcal{S}_X \neq \mathcal{T}_X$ et $\mathcal{S}_{Y|X} \neq \mathcal{T}_{Y|X}$
   Ce dernier cas combine les deux difficultés précédentes. On parle **d'apprentissage par transfert non-supervisé**. Même dans ce cas, il peut exister de l'information commune exploitable entre $\mathcal{S}$ et $\mathcal{T}$. Par exemple, pré-entraîner un modèle sur ImageNet (le domaine source) aura la plupart du temps un impact bénéfique sur une autre tâche de vision (le domaine cible), alors que les deux domaines ne partagent pas les mêmes classes et ont des images de nature potentiellement différente.

Remarque 1.1. *Pour établir cette taxonomie, on compare des distributions conditionnelles $\mathcal{S}_{Y|X}$ et $\mathcal{T}_{Y|X}$. Or, ce n'est pas autorisé lorsque* $\text{supp}(\mathcal{S}_X) \neq \text{supp}(\mathcal{T}_X)$*, puisque l'on ne peut pas conditionner par l'événement impossible. Cette taxonomie n'est donc plus valable si $\mathcal{S}_X$ et $\mathcal{T}_X$ ne partagent pas le même support. Ce point sera discuté en détail dans la section 4.3.*

Dans l'ensemble de ce travail de thèse, nous ne nous intéresserons qu'à la résolution du covariate shift : nous supposerons donc que la tâche à accomplir est la même dans les deux domaines, mais que la nature des données d'entrée peut varier d'un domaine à l'autre.

Toutes les méthodes d'adaptation de domaine traitant le cas du covariate shift supposent l'existence d'au moins un domaine source comportant beaucoup d'exemples annotés, et ont pour but de maximiser la performance dans le domaine cible. Cependant, elles peuvent faire des hypothèses différentes sur les données disponibles dans le domaine cible au cours de l'apprentissage. On peut les classer en trois grandes familles en fonction de leurs conditions expérimentales :

— L'adaptation de domaine non-supervisée (UDA) suppose qu'en plus du domaine source, on dispose d'un nombre raisonnable d'images du domaine cible, mais non-annotées.

— L'adaptation de domaine semi-supervisée (SSDA) suppose qu'en plus du domaine source, on dispose d'une majorité d'images dans le domaine cible non-annotées, ainsi qu'une petite fraction d'images cible annotées

— La généralisation de domaine (DG) suppose qu'on ne dispose d'aucune image dans le domaine cible en plus du domaine source

— Les variantes des méthodes précédentes ont été définies et étudiées dans le cas où on dispose de plusieurs domaines source, on parle alors d'adaptation de domaine multi-source (MS-UDA, MS-SSDA, MS-DG).

Essayons de donner une idée intuitive sur ce que l'on attendrait d'un bon algorithme d'adaptation de domaine : on peut imaginer que résoudre la tâche dans le domaine source fera émerger au sein du réseau deux types de descripteurs de l'image : les premiers sont les descripteurs *invariants* au domaine, dans notre exemple, on peut penser aux descripteurs sensibles uniquement à la forme ou au symbole au centre du panneau. Les seconds, en revanche, seront spécifiques au domaine source, on pense par exemple aux neurones sensibles à la gamme de couleurs propre aux images prises de jour. Ces derniers posent un problème puisque leur signification risque de changer dans le domaine cible, ce qui introduirait de la confusion dans la fonction de classification. Le but premier d'un algorithme d'adaptation de domaine est de pouvoir conditionner le modèle à n'utiliser que les descripteurs qui gardent la même signification dans les deux domaines, afin que la fonction de classification apprise ait le même comportement dans le domaine source et le domaine cible.

La plus grande portion de la littérature existante est consacrée à l'adaptation de domaine non-supervisée. Pour résoudre cette instance du problème, ce sont les méthodes dites « d'alignement de domaine » qui sont les plus nombreuses. Leur but est de produire un espace de représentation en sortie d'un réseau d'encodage qui soit informatif sur la classe, mais invariant au domaine. Elles procèdent en alignant statistiquement les distributions des descripteurs source et cible. Les premiers travaux ayant proposé de mettre en pratique ce principe pour l'apprentissage profond n'alignaient que des aspects statistiques grossiers des distributions que l'on pouvait représenter par une mesure de divergence calculée formellement. Par exemple, la méthode DeepCORAL [Sun and Saenko, 2016] entraîne le réseau d'encodage à aligner les moyennes et les variances-covariances des distributions de descripteurs source et cible. Long et al. [2015] ont ensuite proposé d'exploiter une mesure de divergence plus puissante, mais c'est l'algorithme DANN (pour domain-adversarial training of neural networks) [Ganin et al., 2016] qui marque l'avènement de cette famille de méthodes : DANN exploite l'apprentissage adversaire, originellement proposé pour la génération d'images, pour aligner avec une grande précision les distributions des descripteurs. De nombreuses variantes de cet algorithme suivirent.

Ces méthodes d'alignement de domaine furent directement inspirées par une série de résultats théoriques initiée par Ben-David et al. [2010b], que nous enrichirons au cours de cette thèse. Ces résultats prennent la forme de bornes supérieures sur l'erreur (aussi appelée risque) dans le domaine cible. On cherche donc logiquement à les minimiser en pratique avec un algorithme. Or, ces bornes dépendent, entre autres, d'une mesure de divergence entre les distributions de l'espace d'entrée du domaine source et du domaine cible, ce qui justifie au moins partiellement l'objectif des méthodes d'alignement. Cependant, plusieurs travaux analytiques ultérieurs, auxquels nous avons pu contribuer au cours de cette thèse, ont pu prouver que l'alignement de domaine ne minimise que partiellement ces bornes. Cela se traduit, en théorie comme en pratique, par une absence de garanties sur la performance du modèle dans le domaine cible. Nous allons répartir la présentation de ce travail de controverse dans trois chapitres du manuscrit : dans le chapitre 3, nous constaterons empiriquement l'inefficacité des méthodes d'alignement, nous l'expliquerons ensuite à la lumière de la théorie existante dans le chapitre 4. Enfin, dans le chapitre 5, nous discuterons de l'intérêt réel des méthodes d'alignement en comparaison d'autres facteurs favorisant la transférabilité.

En effet, puisque la théorie indique qu'aligner les domaines n'est pas un critère suffisant pour obtenir une bonne performance dans le domaine cible, il nous est apparu nécessaire de consacrer la suite de la thèse à déterminer les véritables facteurs qui permettent à un modèle de bien transférer d'un domaine vers l'autre. En effet, nous avons pu remarquer que dans les expériences présentées dans la littérature, de nombreux choix algorithmiques présentés comme des « détails expérimentaux » étaient en réalité susceptibles d'avoir un impact plus important sur la performance finale que l'algorithme d'adaptation de domaine étudié. Parmi ces choix algorithmiques, on peut par exemple citer l'utilisation du pré-entraînement, dont on connaît depuis longtemps les bénéfices pour la généralisation, ou encore le type d'architecture choisi (ResNet, Transformer, etc.). Par conséquent, nous présentons dans le chapitre 5 une série d'expériences comparatives et ablatives portant sur ces choix expérimentaux. Nous observons non seulement que la performance des méthodes d'adaptation de domaine n'est pas agnostique à ces choix, mais que l'apport de ces éléments de design a parfois

plus d'impact que la méthode d'adaptation elle-même. Cela remet en question la comparaison et la reproductibilité de la plupart des méthodes proposées dans la littérature. Nous émettons finalement l'hypothèse que ces choix participent à une instance du *biais inductif* qui favorise la transférabilité. La notion de biais inductif regroupe l'ensemble des choix algorithmiques qui conditionnent le comportement d'un modèle hors des données d'entraînement. Nous plaçons cette recherche de biais inductif au cœur de nos recherches ultérieures.

Au cours de ces recherches, nous explorons diverses méthodes incorporant des éléments des littératures de l'apprentissage de représentations et du méta-apprentissage, en plein essor et voisines de celle de l'adaptation de domaine, et évaluons leur intérêt lorsqu'elles sont appliquées dans le cadre d'un transfert de domaine.

— Dans le chapitre 6, nous explorons les méthodes visant à optimiser une instance de biais inductif paramétrique pour améliorer la transférabilité du modèle, nous exploitons notamment la famille d'algorithmes dits de « méta-apprentissage », qui ont pour objectif d'apprendre l'algorithme d'apprentissage lui-même.

— Dans le chapitre 7, nous évaluons l'intérêt des représentations dites « désentrelacées », qui cherchent à extraire les facteurs latents interprétables expliquant chaque domaine.

— Dans le chapitre 8, nous étudions l'intérêt du pré-entraînement pour l'adaptation de domaine. En effet, les bénéfices du pré-entraînement classique pour l'adaptation de domaine identifiés chapitre 5, ainsi que l'essor parallèle des méthodes de pré-entraînement de grande échelle dans la littérature dédiée, nous encouragent à analyser plus en détail quel serait meilleur moyen d'exploiter cette technique pour l'adaptation de domaine.

Pour résumer brièvement la teneur des contributions apportées lors de cette thèse : nos premiers travaux, essentiellement analytiques, nous encouragent à remettre fortement en question l'intérêt réel des avancées récentes en adaptation de domaine, qui ne trouvent justification que dans une série de résultats fragiles, établis uniquement sur une faible variété de transferts et uniquement dans des conditions expérimentales bien précises. Cette trop grande sensibilité aux hyperparamètres ainsi qu'au transfert considéré les rend difficiles à appliquer en pratique. Plutôt que de chercher la performance à tout prix sur une batterie de problèmes bien spécifiques, déjà éprouvés et sur-appris par des méthodes de la littérature, nous orientons donc plutôt notre travail vers des méthodes moins exigeantes en termes de données et plus robustes.

Dans l'ensemble de ce document, nous supposerons que le lecteur est à l'aise avec la plupart des concepts de l'apprentissage machine, de l'apprentissage profond et des probabilités, par exemple les notions de fonction de coût, de sur-apprentissage ou de descente du gradient.

Chapitre 2

État-de-l'art

Sommaire

Ce chapitre est dédié à la présentation des travaux ayant précédé les contributions de cette thèse et permettant de situer ces dernières. Dans ce bref état-de-l'art, nous nous maintiendrons à un niveau de détail grossier, puisque les notions, résultats et méthodes existants seront ensuite rappelés en détail dans les différents chapitres du manuscrit lorsque nous en aurons besoin. Nous verrons dans un premier temps les articles ayant contribué à la théorie de l'adaptation de domaine, puis ceux présentant les algorithmes dits « d'alignement de domaine » que cette théorie a pu inspirer. Nous nous intéresserons ensuite à une série d'analyses montrant que ces algorithmes ne contrôlent pas le risque dans le domaine cible, un travail critique que nous poursuivrons dans ce manuscrit. Enfin, nous présenterons plusieurs familles de méthodes plus modernes incluant d'autres mécanismes que l'alignement des descripteurs. Parmi l'ensemble des travaux présentés, on détaillera particulièrement certains résultats et algorithmes qu'il est indispensable de comprendre pour apprécier les contributions apportées au cours de cette thèse.

2.1 Résultats théoriques pour l'adaptation de domaine

Les premiers travaux traitant le problème de l'adaptation de domaine sont avant tout des contributions théoriques, proposées par exemple pour résoudre un problème de covariate shift par re-pondération [Zadrozny et al., 2003, Sugiyama et al., 2007, Wen et al., 2014, Bickel et al., 2007, Huang et al., 2006] ou pour adapter rapidement un classifieur [Li and Bilmes, 2007]. Mais les principaux résultats fournis par la théorie récente sont des bornes supérieures majorant le risque de classification dans le domaine cible afin d'estimer à quel point un problème de classification peut être adapté à partir d'un domaine source. La plupart de ces bornes font intervenir un terme mesurant la divergence entre les distributions d'entrée source et cible $\mathcal{S}_X$ et $\mathcal{T}_X$. De ce fait, les différences existant entre les différentes bornes découlent principalement de la divergence choisie et du framework statistique sur lesquelles elles se basent : (par exemple VC, Rademacher, PAC-Bayes).

Ben-David et al. [2010b] fut le premier à proposer de telles bornes. La première d'entre-elles est basée sur la divergence de variation totale, par nature indépendante de la classe d'hypothèses $\mathcal{H}$ choisie pour la fonction de classification.

Pour se rapprocher d'un cas réaliste et prendre en compte les régularités induites par l'hypothèse h choisie, de nombreux travaux ont proposé des bornes basées sur des divergences dépendantes de la classe d'hypothèses $\mathcal{H}$. La borne de Ben-David et al. [2010b] la plus connue utilise la divergence $H\Delta H$, qui quantifie à quel point deux hypothèses peuvent désynchroniser leur désaccord d'un domaine vers l'autre. Cette borne a ensuite été reprise par Zhang et al. [2019] en relaxant notamment la divergence à une seule hypothèse adversaire. De très nombreuses bornes dépendant de $\mathcal{H}$ ont ensuite été proposées pour majorer le risque dans le domaine cible : Mansour et al. [2009] propose une autre borne basée sur les classifieurs. Courty et al. [2017], Shen et al. [2017] proposent une borne basée sur la distance de Wasserstein, qui présente donc un certain degré de robustesse dans le cas où $\mathcal{S}$ et $\mathcal{T}$ auraient des supports disjoints. Un certain nombre de travaux ont pu proposer des bornes utilisant le framework PAC-bayésien [Germain et al., 2015, 2016]. Enfin, on notera l'existence d'une multitude de travaux annexes quantifiant la complexité d'estimation de ces bornes [Ben-David and Urner, 2012].

Le Chapitre 4 est entièrement consacré à la théorie de l'adaptation de domaine et donnera davantage de détails sur ces bornes. Le lecteur est par ailleurs invité à lire les surveys [Kouw, 2018, Redko et al., 2020] pour une liste exhaustive des bornes existantes en adaptation de domaine et une description détaillée de leurs propriétés.

2.2 Méthodes d'alignement de domaine

Ces résultats théoriques ont inspiré une multitude de travaux proposant des algorithmes dans le cadre non-supervisé (UDA) pour minimiser certaines de ces bornes théoriques (et donc de contrôler le risque dans le domaine cible). Une écrasante majorité d'entre eux proposent des méthodes dites

d'« alignement de domaine ». Ces méthodes se donnent pour objectif d'entraîner un modèle à produire des descripteurs invariants au domaine, c'est-à-dire au style de l'image, mais informatifs sur le contenu, en particulier la classe de l'image. Concrètement, on espère qu'une fonction de classification basée sur de tels descripteurs soit robuste aux variations de style, et donc généralise vers le domaine cible. D'un point de vue théorique, ces algorithmes minimisent le terme de divergence présent dans la plupart des bornes de la littérature.

En notant X, Z, Y l'espace des images, des descripteurs (resp.), des labels (resp.). Toutes les méthodes d'alignement découpent la chaîne de classification en deux parties : l'extracteur de descripteurs $\Psi : X \to Z$ et le classifieur $h : Z \to Y$.

— L'extracteur de descripteurs est la fonction qui produit le descripteur à partir d'une image donnée. En traitement d'images, il s'agira typiquement d'un réseau convolutif, prenant en entrée un tenseur de taille $C \times W \times H$ (nombre de canaux, largeur et hauteur de l'image) et donnant en sortie un vecteur de grande dimension (généralement 512 à 4096).

— Le classifieur est la fonction qui cherche à prédire l'étiquette d'une image à partir de son descripteur. Il s'agira typiquement d'un simple réseau complètement connecté, avec une couche cachée ou plus.

L'extracteur et le classifieur sont entraînés conjointement pour satisfaire les deux objectifs suivants :

— Minimiser l'erreur de classification sur le domaine source $R_{\mathcal{S}}$. C'est chose aisée puisque l'on dispose des annotations sur ce domaine, il suffit de minimiser une fonction de coût entre les prédictions et les vérités terrains, par exemple l'entropie croisée $\mathcal{L}_{CE}$.

— Minimiser une mesure de divergence entre la distribution des descripteurs issus des images sources et la distribution des descripteurs issus des images cibles. Notons $\mathcal{S}_X$ et $\mathcal{T}_X$ la distribution marginale des domaines source et cible (resp.) sur l'espace des images. Les distributions marginales des descripteurs source et cible peuvent être définies par les expressions abusives $\mathcal{S}_Z = \Psi(\mathcal{S}_X)$ et $\mathcal{T}_Z = \Psi(\mathcal{T}_X)$.

Le premier objectif assure que les descripteurs produits par Ψ sont informatifs sur la classe (pour le domaine source au moins). Le second assure que les descripteurs soient invariants au domaine. En effet, si la divergence $d(\mathcal{S}_Z, \mathcal{T}_Z)$ est totalement minimisée, $\mathcal{S}_Z = \mathcal{T}_Z$. L'objectif total de ces méthodes peut donc se résumer à l'expression suivante :

$$\mathcal{L} = \mathcal{L}_{CE}(\mathcal{S}) + \lambda d(\mathcal{S}_Z, \mathcal{T}_Z)$$

Avec λ un coefficient réglant le compromis entre les deux parties de l'objectif.

Les méthodes d'alignement se distinguent principalement par le choix de la divergence à minimiser dans l'espace des descripteurs. Historiquement, ces méthodes sont apparues à mesure que des divergences optimisables de plus en plus complexes apparurent dans la littérature, afin d'aligner S_Z et T_Z avec de plus en plus de finesse. Deep CORAL [Sun and Saenko, 2016] est une des premières méthodes proposant d'aligner les descripteurs d'un modèle profond. Très simple, elle n'impose que l'alignement des statistiques d'ordre 1 et 2 de S_Z et T_Z, à savoir les moyennes et les covariances.

$$\mathbb{E}[S_Z] = \mathbb{E}[T_Z] \text{ et } \mathrm{Var}(S_Z) = \mathrm{Var}(T_Z)$$

Plus sophistiquée que Deep CORAL, la méthode proposée par Tzeng et al. [2014] estime une divergence $d(S_Z, T_Z)$ basée sur la moyenne des projections des descripteurs de $\mathcal{Z}$ par des fonctions à noyaux reproduisant. Cette divergence est appelée Maximum Mean Discrepancy (MMD). Cette méthode a ensuite été étendue par Long et al. [2015] avec un ensemble de noyaux différents. La littérature naissante des modèles génératifs adversariaux [Goodfellow et al., 2014] a ensuite popularisé une nouvelle manière d'aligner les distributions, qui fut rapidement appliquée à l'adaptation de domaine avec l'algorithme DANN [Ganin et al., 2016], pour « Domain-Adversarial training of neural networks ». En complément de l'objectif habituel $\mathcal{L}_{CE}(\mathcal{S})$, l'algorithme utilise une fonction de coût adversaire obtenue par l'intermédiaire d'un discriminateur pour forcer l'alignement des descripteurs

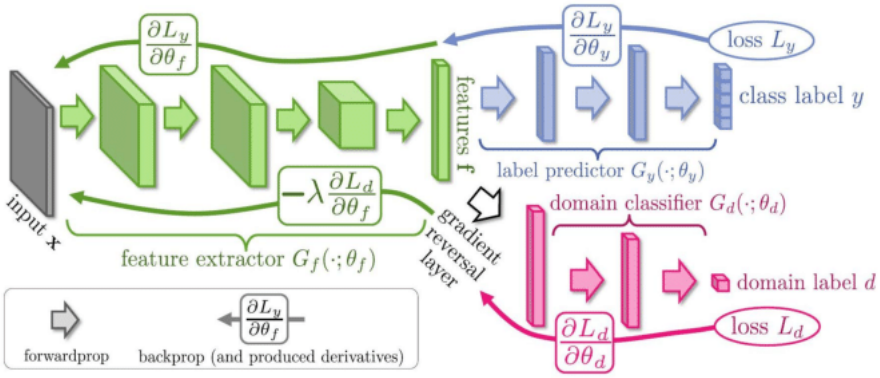

FIGURE 2.1 – Schéma de la méthode DANN

de chaque domaine : on entraîne ce discriminateur binaire $d : Z \to \{0, 1\}$ à prédire le domaine dont est issu un descripteur $z \in Z$ (en minimisant une erreur de classification binaire). Ensuite, l'extracteur de caractéristiques Ψ s'entraîne à son tour pour induire d en confusion (en maximisant l'erreur de classification binaire), ce qui a pour conséquence de rapprocher les domaines au sein de Z. On peut résumer cet entraînement par le jeu min-max suivant :

$$\max_{\Psi} \min_{d} \mathcal{L}_{CE}(d(\Psi(x)), dom(x)) \tag{2.1}$$

Le fonctionnement de l'algorithme est résumé dans la figure 2.1. Noter que DANN est directement inspiré de la borne de Ben-David.

De nombreuses variantes de DANN ont ensuite pu être proposées : ADDA [Tzeng et al., 2017] utilise deux encodeurs distincts pour le domaine source et le domaine cible et n'optimise pas les deux objectifs en même temps. PixelDA [Bousmalis et al., 2017] projette le domaine source vers le domaine cible directement dans l'espace des pixels : il s'agit donc de réaliser une traduction d'image pure et simple à l'instar d'un CycleGAN [Zhu et al., 2017] avant d'appliquer le classifieur. On peut considérer cela comme une forme d'alignement asymétrique (car on projette le domaine source vers le domaine cible au lieu de projeter les deux domaines dans un espace de représentation commun).

D'autres méthodes proposent d'améliorer DANN en gardant les mêmes choix algorithmiques, mais en se basant sur une borne plus récente que celle de Ben-David. Par exemple, Shen et al. [2017] exploite un discriminateur utilisant la distance de Wasserstein plutôt que celle de Jensen-Shannon (implicitement utilisée par DANN), cette distance a l'avantage d'être robuste et informative dans certains cas où les supports sont disjoints. Zhang et al. [2019] utilise la borne définie par l'équation (4.5) plutôt que celle de Ben-David, puisque le terme de divergence est plus facile à estimer et à minimiser.

Les méthodes d'alignement dont nous venons de faire l'inventaire sont directement inspirées des bornes dérivant de Ben-David et al. [2010b], en les appliquant notamment dans l'espace des descripteurs. Ces algorithmes sont habituellement présentés dans leurs publications respectives comme des minimisateurs de ces bornes [Ganin et al., 2016, Shen et al., 2017, Zhang et al., 2019], garantissant ainsi une bonne performance dans le domaine cible. Cependant, divers travaux analytiques [Shu et al., 2018, Zhao et al., 2019, Johansson et al., 2019, Bouvier et al., 2020] prouvent que les bornes ne sont en réalité que partiellement minimisées et que les algorithmes d'alignement ne sont donc pas suffisants pour garantir une bonne performance de classification dans le domaine cible. La poursuite de ce travail critique, complétée par une meilleure compréhension des faiblesses des méthodes d'alignement, est une des contributions principales de cette thèse [Siry et al., 2020a, 2021].

Après avoir constaté que les bornes théoriques ne peuvent pas entièrement expliquer comment obtenir une bonne performance dans le domaine cible, nous verrons dans ce manuscrit que

l'adaptation de domaine est en réalité fortement influencée par la notion de *biais inductif*. On appelle biais inductif l'ensemble des hypothèses (implicites ou explicites), qui, additionnées aux données d'entraînement, conditionnent le comportement d'un modèle sur les échantillons de test. L'importance de certaines instances de biais inductif en adaptation de domaine a déjà été remarquée par Bouvier et al. [2020]. Une partie du travail de cette thèse, résumée dans le chapitre 5, consiste à analyser et à cataloguer les différentes instances du biais inductif déjà couramment utilisées dans la littérature, rendant de facto la transférabilité plus facile, mais dont l'impact est peu, voire pas étudié dans les publications. Nous listons dans les sections suivantes l'ensemble des méthodes prenant en compte une autre forme de biais inductif que l'alignement des domaines.

Prenant acte que l'alignement de domaine n'était pas suffisant pour obtenir une bonne performance dans le domaine cible, de nouvelles améliorations furent trouvées par la communauté pour mieux conditionner le procédé d'alignement des distributions latentes : [Shu et al., 2018, Häusser et al., 2017, Xu et al., 2019] exploitent l'hypothèse que les distributions latentes forment des clusters pour relaxer l'alignement des distributions à l'alignement des centroides de ces clusters. D'autres travaux récents [Cao et al., 2018, Wu et al., 2019, You et al., 2019] suggèrent également d'aligner partiellement les domaines en exploitant certaines hypothèses pour gagner en performance. [Cicek and Soatto, 2019, Chen et al., 2019, Zhang et al., 2019, Saito et al., 2018, Kang et al., 2019, Lv et al., 2021] utilisent des pseudo-labels pour conditionner l'alignement des distributions par cette information de pseudo-classe. Kumar et al. [2018] réalise l'alignement sur un ensemble d'espace de descripteurs différents, et force un unique classifieur à fonctionner sur chacun d'entre eux. Bouvier et al. [2019] proposent un algorithme robuste à la fois au prior shift et au concept shit.

2.3 Méta-apprentissage

Le méta-apprentissage est un paradigme récent en apprentissage machine. Plutôt que d'entraîner un modèle à résoudre une unique tâche, on apprend un algorithme d'apprentissage spécialisé dans la résolution de n'importe quelle tâche d'un ensemble de tâches partageant des caractéristiques communes (par exemple, l'ensemble des tâches consistant à classifier N animaux). Autrement dit, on cherche à apprendre un biais inductif paramétrique qui soit utile pour une famille de tâches donnée. On peut résumer simplement ce principe comme le fait « d'apprendre à apprendre ». Le méta-apprentissage a d'abord été introduit dans le cadre du few-shot learning avec l'algorithme MAML [Finn et al., 2017]. Dans le cadre de MAML, une tâche est un scénario few-shot au sein duquel un modèle de classification de paramètre initial Φ subit K étapes de SGD sur les quelques shots d'entraînement. Dans ce contexte, le but du méta-apprentissage est d'optimiser le paramètre initial Φ pour que, en moyenne, sur l'ensemble des tâches, on obtienne un modèle de classification qui généralise bien après ces K étapes de descente de gradient. Pour ce faire, on évalue le modèle obtenu sur un ensemble de test avec une fonction de coût, puis on différencie ce coût par-rapport à Φ à travers les K mises à jour internes.

Le méta-apprentissage a ensuite été exploité pour l'adaptation de domaine, en particulier dans le cas de la généralisation de domaine multi-source (MS-DG), c'est-à-dire le cas où on a N domaines source annotés et un domaine cible non-annoté totalement inaccessible. Li et al. [2018] reprend la boucle d'apprentissage de MAML, mais en adaptant le scénario d'apprentissage de chaque tâche au contexte de la MS-DG. Dans ce cadre, une tâche de méta-entraînement consiste à fine-tuner K fois à partir d'échantillons provenant de $N-1$ domaines (choisis parmi les N domaines source) et d'évaluer le modèle obtenu sur les échantillons du domaine restant. Une fois le méta-entraînement terminé, on fine-tune l'initialisation obtenue sur les N domaines source simultanément et on espère que le modèle obtenu généralisera vers le véritable domaine cible.

Cette méthode a ensuite été améliorée par Balaji et al. [2018] : plutôt que de méta-apprendre une initialisation Φ, on optimise maintenant un poids de régularisation (L1 ou L2) pour chaque paramètre scalaire du classifieur. Ces poids sont méta-entraînés en suivant les mêmes scénarios que Li et al. [2018], mais en considérant cette fois un fine-tuning régularisé.

Enfin, Wei et al. [2021] utilise le méta-apprentissage pour apprendre une paramétrisation de

l'encodeur qui synchronise naturellement les dynamiques d'apprentissage de la supervision en source et de l'alignement de domaine pour accroître la stabilité et la performance d'un scénario de type DANN, cette dernière méthode couvrant les cas de l'UDA et de la MS-DG.

2.4 Apprentissage de représentations avec contraintes

Pour compléter les insuffisances des méthodes d'alignement en termes de biais inductif, des méthodes d'UDA récentes se sont inspirées de plusieurs pans de la littérature de l'apprentissage de représentations. Ces méthodes proposent généralement d'enrichir l'objectif d'alignement des domaines par des objectifs de modélisation auxiliaires visant à rendre les descripteurs plus robustes, transférables et parfois même interprétables. Ces objectifs sont non-supervisés et sont repris de la littérature de l'apprentissage de représentations désentrelacées, ou bien de la littérature de l'auto-supervision.

L'apprentissage de représentations désentrelacées consiste à entraîner un modèle à produire une représentation $\mathcal{D}_Z = \Psi(\mathcal{D}_X)$ satisfaisant plusieurs propriétés.

— Premièrement, elle doit contenir toute l'information de l'image, autrement dit, il doit être possible de reconstruire parfaitement l'image à partir de sa représentation, ce qui n'est pas le cas des représentations produites par l'alignement de domaines par exemple.

— Ensuite, les facteurs explicatifs importants de la distribution d'images d'entrée doivent pouvoir être identifiés et démêlés facilement au sein de cette représentation latente comme des sources d'information indépendantes. Cet objectif peut être imposé à plusieurs niveaux :

  — D'une part, on peut l'imposer de manière totalement non-supervisée à l'échelle des dimensions de Z. Autrement dit, on cherche à imposer que les dimensions de Z soient statistiquement indépendantes, soit $p(z) = \prod_i p(z_i)$. Cet objectif a à l'origine été proposé par Higgins et al. [2017] en imposant simplement $p(z) = \mathcal{N}(0, \mathbf{I}^N)$ (avec N le nombre de dimensions de Z). Les auteurs mettent en avant le fait que la découverte de composantes indépendantes dans l'information de $\mathcal{D}_X$ coïncide souvent avec la découverte de composantes interprétables. Ce paradigme a ensuite été largement étudié par d'autres méthodes [Makhzani et al., 2015, Donahue et al., 2016]. Notons enfin que des limites à ce paradigme ont été trouvées par au moins deux analyses critiques [Locatello et al., 2018, Mathieu et al., 2018], qui proposent des contre-exemples dans lesquels les représentations indépendantes ne sont pas interprétables.

  — D'autre part, si on souhaite désentrelacer des sources d'information bien identifiées et pour lesquels on dispose des labels, on peut imposer ce critère de façon supervisée. C'est le mécanisme principal utilisé par les méthodes de désentrelacement appliquées à l'adaptation de domaine. Ces dernières cherchent en effet à séparer l'information de classe de celle de domaine dans l'espace Z. Par exemple, si $N = 256$, on peut imposer que l'information de classe soit contenue uniquement dans les 128 premières dimensions et celle de domaine uniquement dans les 128 dernières [Peng et al., 2019, Bousmalis et al., 2016, Cai et al., 2019, Gonzalez-Garcia et al., 2018, Fu et al., 2017, Lee et al., 2021]. Cette propriété est imposée par l'intermédiaire de classifieurs adversaires d'une part, que l'on entraîne pour assurer qu'une partie de la représentation est invariante à une certaine source d'information (exactement comme ce qui est fait dans DANN avec l'information de domaine), et par l'intermédiaire de classifieurs normaux d'autre part pour s'assurer qu'une information est bien présente dans une partie de l'espace latent. L'ensemble des mécanismes est détaillé dans le chapitre 7. Dans le cas de l'application à l'adaptation de domaine, cet entraînement est en vérité semi-supervisé, puisque les labels de domaine sont disponibles dans les deux domaines, mais ceux de classe ne sont disponibles que dans le domaine source. On espère donc dans ce cas que la propriété de désentrelacement généralise vers le domaine cible.

Faisons remarquer que ces deux mécanismes de désentrelacement ne s'excluent pas et peuvent très bien être cumulés dans un même algorithme d'apprentissage [Peng et al., 2019, Cai et al., 2019].

L'apprentissage de représentations par auto-supervision est un autre paradigme de l'apprentissage de représentations. Il diffère radicalement de l'apprentissage de représentations désentrelacées dans le sens où il n'impose pas explicitement de propriété particulière aux représentations. En effet, l'apprentissage par auto-supervision consiste plutôt à obliger le modèle à résoudre une tâche auxiliaire qui, si elle est résolue par le modèle, fera (à priori) émerger de bonnes représentations comme effet de bord. Ces problèmes auxiliaires doivent pouvoir être synthétisés automatiquement.

Une des premières méthodes d'auto-supervision proposées [Gidaris et al., 2018] consiste à appliquer une rotation à l'image d'entrée de 0, 90, 180 ou 270 degrés et de demander à l'extracteur Ψ de prédire quelle rotation a été appliquée. Il s'agit donc d'un simple problème de classification à quatre sorties. Prédire cette rotation oblige le modèle à examiner un certain nombre d'indices visuels de haut niveau, ce qui fait émerger des descripteurs robustes. D'autres méthodes d'auto-supervision sont également basées sur une tâche de classification, par exemple la prédiction de contexte [Doersch et al., 2015] ou la résolution d'un puzzle [Noroozi and Favaro, 2016].

Nous pouvons identifier une seconde catégorie basée d'auto-supervision basée sur la reconstruction d'une partie manquante de l'information d'entrée. D'abord utilisé en traitement du langage (prédiction d'un mot manquant ou de la suite d'une phrase), ce principe a ensuite été étendu aux modèles d'image. Dans ce cas de figure, la tâche type consiste à masquer une zone plus ou moins significative de l'image (avec un rectangle noir par exemple). Le modèle s'entraîne ensuite à reconstruire (en espérance) la zone masquée étant donné la zone visible. Intuitivement, pour réaliser une reconstruction correcte, le modèle doit apprendre à analyser les éléments visibles de l'image à tous les niveaux d'abstraction (colorimétrie, mais aussi objets, disposition de la scène) afin de produire la meilleure estimation possible de la partie manquante. Des méthodes de ce type ont déjà été présentées avec des réseaux convolutifs [Pathak et al., 2016] et des vision transformers [He et al., 2021, Bao et al., 2021] et montrent que cet objectif fait émerger des représentations de bonne qualité.

Nous pouvons enfin identifier une troisième famille de méthodes d'auto-supervision, basée sur les objectifs dits de contraste (contrastive representation learning) [Chen et al., 2020b, Grill et al., 2020, Caron et al., 2021]. Ces méthodes ont récemment connu un grand succès du fait de leur simplicité et de leurs bonnes performances. La première proposée, et la plus emblématique, est SimCLR [Chen et al., 2020b] : elle consiste à entraîner un extracteur de caractéristiques Ψ à être, pour un jeu de données $\mathcal{D}$ et une famille de transformations aléatoires sur l'espace des images T, 1) équivariant à l'image d'entrée, c'est-à-dire que $\forall x_1, x_2 \sim \mathcal{D}_X$, $\Psi(x_1) \neq \Psi(x_2)$ et 2) invariant à n'importe quelle transformation de T, c'est-à-dire $\forall x \sim \mathcal{D}_X, \forall t_1, t_2 \sim T, \Psi(t_1(x)) = \Psi(t_2(x))$. T est choisie comme étant la famille résultant de la composition de fonctions d'augmentation aléatoires classiques en apprentissage profond (rognage aléatoire, flou, distorsion des couleurs, translation, etc). Pour obtenir une telle propriété d'invariance-équivariance sur les descripteurs, on entraîne le modèle à rapprocher les descripteurs de paires d'images « positives » $t_1(x), t_2(x)$, c'est-à-dire obtenues à partir d'une seule même image de $\mathcal{D}_X$, et en éloignant les descripteurs des paires « négatives » $t_1(x_1), t_2(x_2)$, c'est-à-dire obtenues à partir de deux images distinctes de $\mathcal{D}_X$. La méthode BYOL Grill et al. [2020] garde cette notion d'objectif de contraste, mais sans avoir besoin de paires négatives, ce qui permet d'entraîner avec des tailles de batch plus faibles. Enfin, Caron et al. [2021] applique un objectif de contraste aux modèles de vision transformer.

Dans l'ensemble des travaux mentionnés, l'objectif d'auto-supervision est utilisé comme pré-entraînement sur une base d'images importante comme ImageNet pour améliorer les propriétés de généralisation, que ce soit sur cette base ou sur des problèmes subséquents. Dans ce manuscrit, en particulier dans le chapitre 8, nous étudierons les bénéfices du pré-entraînement auto-supervisé sur ImageNet pour l'adaptation de divers transferts. Cependant, il existe des méthodes d'adaptation de domaine appliquant l'objectif d'auto-supervision directement sur les données source et cible : par exemple, Sun et al. [2019] complète l'alignement des domaines avec un objectif d'auto-supervision

par prédiction de rotation et localisation de patch.

2.5 Adaptation de domaine semi-supervisée

Nous avons jusque-là présenté des méthodes traitant le cas de l'adaptation non supervisée (UDA) et de la généralisation de domaine multi-source (MS-DG). Dans cette section, nous listons les méthodes se plaçant dans le cadre de l'adaptation de domaine semi-supervisée (SSDA). La plupart de ces méthodes essaient de diminuer la divergence existant dans le domaine cible entre les échantillons annotés et les échantillons non-annotés pour stabiliser l'entraînement. Pour ce faire, Yoon et al. [2022] utilise de l'auto-distillation. Saito et al. [2019], Kim and Kim [2020] diminuent l'entropie de classification dans le domaine cible. Ben-David et al. [2010a] analyse le problème de l'adaptation de domaine semi-supervisée d'un point de vue théorique, montrant qu'il existe une transition de phase entre le régime non supervisé (UDA) et celui ou tout est supervisé en cible.

Chapitre 3

Évaluation expérimentale des méthodes par alignement de domaine

Sommaire

Nous avons pu assister ces dernières années à une prolifération de méthodes d'adaptation de domaine ayant pour mécanisme de base, à l'instar de DANN [Ganin et al., 2016], l'alignement adversaire entre le domaine source et le domaine cible [Zhang et al., 2019, Tzeng et al., 2017, Shen et al., 2017]. Toutes ces méthodes proposent une variation particulière de ce mécanisme dans leur implémentation dont l'intérêt n'est pas toujours bien justifié. Ce chapitre est consacré à la première étude expérimentale menée au cours de cette thèse, dont le but est de constater et de caractériser les limites des méthodes d'alignement de domaine, populaires dans la littérature. L'ensemble du contenu présenté ici correspond aux résultats de notre première publication [Siry et al., 2020a]. Cette analyse empirique expose un certain nombre de faiblesses existant dans la pratique de l'adaptation de domaine.

3.1 Méthodes étudiées

Nous avons déjà eu l'occasion d'expliquer le principe général des méthodes d'alignement dans le chapitre 2. Parmi toutes les méthodes existant dans la littérature, nous menons notre étude sur trois papiers représentatifs de l'état-de-l'art : ADDA [Tzeng et al., 2017], DANN [Ganin et al., 2016] et MDD [Zhang et al., 2019], dont nous rappelons brièvement le fonctionnement.

— **ADDA** Est un algorithme en deux étapes, qui consiste dans un premier temps à entraîner conjointement un classifieur h et un extracteur de caractéristiques Ψ_s par supervision dans le domaine source, puis dans un deuxième temps entraîner un second extracteur de caractéristiques $\Psi_t(x)$ à reproduire les descripteurs du domaine source, mais à partir des échantillons cible (c'est-à-dire $\Psi_t(\mathcal{T}_X) == \Psi_s(\mathcal{S}_X)$). Les auteurs de la méthode défendent une plus grande facilité d'entraînement puisque les deux objectifs sont entraînés séparément. On peut cependant émettre des interrogations sur les performances de cet algorithme, dans la mesure où il définit la géométrie de l'espace de descripteurs sans prendre en compte le domaine cible.

— **DANN** entraîne un seul couple h, Ψ pour les deux domaines, et repose sur une descente de gradient alternée pour optimiser deux objectifs : une fonction de coût de classification standard calculée sur le domaine source et une fonction de coût adversaire obtenue par l'intermédiaire d'un discriminateur pour forcer l'alignement des descripteurs de chaque domaine. Les auteurs prétendent que cet algorithme découle naturellement de la borne supérieure présentée par Ben-David et al. [2010b]. Cependant, ce lien demeure ténu [1] et ne permet pas de définir précisément quelle est la meilleure manière d'aligner les domaines. À titre d'exemple, nous proposons d'étudier trois variantes de DANN en particulier :

  — La variante « source to target » (S2T), au sein de laquelle la distribution latente du domaine source est encouragée à se rapprocher des échantillons du domaine cible

  — La variante « target to source » (T2S), au sein de laquelle la distribution latente du domaine cible est encouragée à se rapprocher des échantillons du domaine source [2]

  — Enfin, la variante symétrique (SYM), où les deux distributions sont encouragées à se rapprocher l'une de l'autre

  L'implémentation des trois variantes est très simple, en choisissant de rétropropager le gradient de la fonction de coût relative à l'alignement à travers la passe d'encodage des échantillons source uniquement, cible uniquement, ou les deux. Nous comparons ces variantes, car elles ont théoriquement les mêmes points d'équilibre, mais présentent des dynamiques d'apprentissage différentes.

— **MDD** est un algorithme découlant d'une version améliorée de la borne de Ben-David et al. [2010b]. Son fonctionnement est très similaire à celui de DANN, à la différence près qu'il remplace le discriminateur de domaines par un classifieur adversaire h' dont l'objectif est

1. Nous expliquerons ce lien en détail dans le Chapitre 4
2. Noter qu'il s'agit du choix d'implémentation des auteurs de DANN

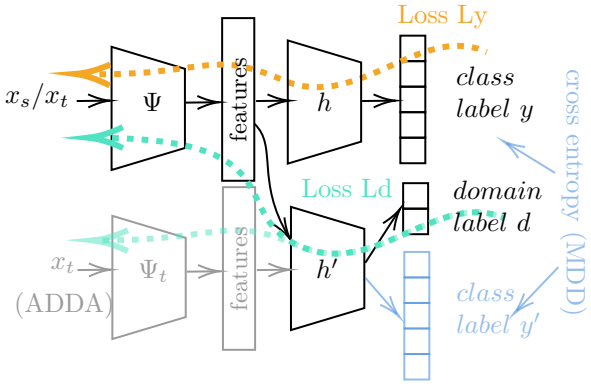

FIGURE 3.1 – Schéma bloc résumant toutes les méthodes considérées

d'imiter le comportement de h sur le domaine source et de s'en distinguer le plus possible dans le domaine cible. L'unique extracteur Ψ s'entraîne ensuite pour contrecarrer h'. En résumé, on a l'optimisation adversaire suivante :

$$\min_{\Psi,h} \max_{h'} = \sup_{h' \in \mathcal{H}} \mathop{\mathbf{Pr}}_{\mathcal{D}_2}(h' \neq h) - \mathop{\mathbf{Pr}}_{\mathcal{D}_1}(h' \neq h)$$

Pour savoir comment optimiser cette quantité, nous invitons le lecteur à lire le papier original [Zhang et al., 2019], ainsi la section 8.4 de ce manuscrit dans laquelle nous réexaminerons cet algorithme. Nous verrons dans le chapitre 4 qu'il n'existe pas de résultat théorique définitif justifiant que MDD soit meilleur que DANN.

La totalité des algorithmes comparés utilisent une architecture très similaire, la seule chose les distinguant étant la fonction de coût et les mécanismes d'apprentissage. On peut donc définir 5 modèles, que l'on résume dans la Figure 4.3 :

— SEP (separate) : h et Ψ_s sont entraînés par supervision sur source. On entraîne ensuite Ψ_t à s'aligner sur la distribution des descripteurs en sortie de Ψ_s. On utilise enfin $h \circ \Psi_t$ pour faire de l'inférence sur le domaine cible. Le discriminateur est un classifieur binaire. Cette méthode correspond exactement à la méthode ADDA que nous avons décrite plus haut.

— S2T (source to target) : Un seul encodeur est entraîné conjointement sur les domaines source et cible. La fonction de coût d'alignement est rétro-propagée sur le domaine source uniquement. Nous utilisons dans ce cas le même discriminateur que pour SEP.

— T2S (target to source) : Même principe que S2T, à la différence près que l'on rétropropage la fonction de coût d'alignement sur le domaine cible uniquement.

— SYM (symmetric) : Même principe que T2S, S2T, mais on rétropropage la fonction de coût d'alignement sur les deux domaines à la fois.

— MDD : Cette méthode reprend le même principe que SYM, mais on remplace le discriminateur par un classifieur h'. La fonction de coût adversaire est basée sur l'accord entre h et h'.

Nous ajoutons la baseline de référence faible « Source-Only », qui consiste à entraîner $h \circ \Psi$ sur le domaine source, puis directement tester sur le domaine cible sans aucune adaptation.

3.2 Jeux de données utilisés

3.2.1 Jeux de données usuels

On évalue l'ensemble des méthodes sur des transferts construits à partir des jeux de données à 10 classes suivants, largement utilisés dans la littérature : MNIST et USPS (chiffres manuscrits en

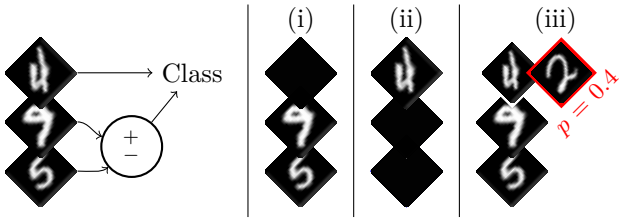

FIGURE 3.2 – Notre dataset MNIST-Algebra et ses trois variantes : i) HideDigit, ii) HideSub, iii) Polluted

noir et blanc), MNIST-M (une variante augmentée de MNIST avec des fonds et des textures tirés d'images réelles) et SVHN (des numéros de maison, très variés en termes de couleurs, de texture et de police). Le principal avantage de ces jeux de données est qu'ils permettent de construire des problèmes d'adaptation de difficulté très variable en dépit de leur relative simplicité. Ces jeux de données seront ré-utilisés tout au long de ce travail de thèse. Par conséquent, on centralise leur descriptif détaillé dans l'Annexe A.

3.2.2 Introduction de nouveaux jeux de données

Une des contributions apportées dans cette étude est l'introduction de nouveaux jeux de données synthétiques, spécialement conçus pour mettre en évidence certains phénomènes qui ne peuvent pas être observés dans les problèmes habituellement utilisés dans la littérature, tout en étant très simples à comprendre.

MNIST-Algebra : On construit ce jeu de données à 10 classes directement à partir de MNIST. Un échantillon de MNIST-Algebra est construit en empilant trois images en noir et blanc de MNIST sur la dimension des canaux, ce qui donne une image RGB. Le chiffre de la première image (R) donne directement la classe de l'échantillon. Les images des deux canaux restants (G, B) sont choisies telles que la différence modulo 10 de leurs deux chiffres MNIST corresponde au chiffre du premier canal (R), ce qui est résumé dans la Figure 3.2. En construisant le jeu de données de cette manière, on cherche à créer deux sources d'information redondantes, informatives sur la classe : la première, facile à exploiter, est donnée directement par le canal (R). La seconde, plus difficile à exploiter et donc susceptible d'être ignorée par le modèle, est donnée par les deux canaux restants (G, B).

Variantes de MNIST-Algebra : On construit par-ailleurs deux variantes de MNIST-Algebra en supprimant une des deux sources d'information, que l'on nomme *HideDigit* et *HideSub* respectivement. Dans *HideDigit*, le chiffre MNIST du canal (R) est remplacé par une image noire uniforme, tandis que dans *HideSub*, ce sont les canaux (G, B) qui sont occultés. On construit également une troisième variante du jeu de données, *Polluted*, au sein de laquelle l'image du canal (R) est remplacée par une image MNIST aléatoire avec une probabilité de 0.4. En corrompant aléatoirement la source d'information (R), on imagine que le modèle sera encouragé à l'ignorer et à se baser davantage sur (G, B).

3.3 Expériences

Détails d'implémentation : On choisit pour l'extracteur Ψ une architecture convolutive très simple, comportant 5 couches et une fonction d'activation LeakyReLU. Ψ produit un vecteur de description de 512 dimensions. Les discriminateurs et classifieurs sont des réseaux feed-forward, avec deux couches cachées comportant chacune 1024 unités. Pour satisfaire l'objectif d'alignement, Ganin et al. [2016] utilisent un module appelé *Gradient Reversal Layer*. Ce module se comporte comme la fonction identité lors d'une passe avant, et inverse le signe du gradient lors de la rétropropagation. Intercalé entre l'extracteur de caractéristiques et le classifieur adversaire, il permet d'entraîner

l'objectif min-max de manière synchrone, en une seule passe de rétropropagation. Nous avons constaté comme Tzeng et al. [2017] que cette méthode d'entraînement était particulièrement instable. À l'instar des papiers récents proposant divers choix algorithmiques pour stabiliser les GAN [Goodfellow et al., 2014, Arjovsky et al., 2017, Miyato et al., 2018], on choisit plutôt un entraînement alterné avec 4 itérations de discriminateur/classifieur pour chaque itération du générateur. Nous utilisons de plus la *Normalisation Spectrale* [Miyato et al., 2018] dans toutes les couches des modèles adversaires pour davantage de stabilité. La normalisation spectrale limite à 1 la valeur absolue des valeurs singulières de chaque couche linéaire, cela permet d'assurer que la constante de Lipschitz du discriminateur soit bornée, ce qui régularise la surface d'erreur produite par le discriminateur et par rapport à laquelle Ψ doit s'optimiser. Enfin, on utilise un objectif asymétrique pour le discriminateur et l'extracteur de caractéristiques, comme recommandé par Goodfellow et al. [2014], qui consiste à inverser la valeur des labels plutôt qu'à changer le signe de la fonction objectif.

$$\min_{\Psi} \mathcal{L}_{CE}(d(\Psi(x)), 1 - dom(x))$$

Au lieu de :

$$\max_{\Psi} \mathcal{L}_{CE}(d(\Psi(x)), dom(x))$$

On entraîne tous les modèles sur 5000 itérations, avec une batch size de 64 et on répète toutes les expériences 10 fois pour mesurer les écarts-types.

Bien que ça ne soit pas toujours fait en adaptation de domaine, on veille à ce que domaine cible soit bien découpé en sous-ensembles entraînement/test, en utilisant uniquement les exemples d'entraînement pour l'alignement.

Comparaison des différents mécanismes d'alignement Avant d'expérimenter sur notre propre jeu de données MNIST-Algebra, nous présentons quelques résultats sur des transferts communs dans la littérature afin d'évaluer la qualité de notre implémentation. Pour comparer les différentes fonctions de coût dédiées à l'alignement des domaines, nous les évaluons sur le transfert SVHN→ MNIST, très étudié dans la littérature, suffisamment simple pour les modèles non pré-entraînés sans être complètement trivial. Les mêmes performances relatives ont pu être observées entre les méthodes d'alignement sur les autres transferts. Les résultats sont résumés dans la Table 3.1. Notre implémentation de DANN atteint une performance similaire à celle du papier d'origine dans le cas T2S, voire légèrement meilleure lorsque l'alignement SYM est appliqué. Le mapping S2T obtient la performance la plus mauvaise, et la méthode SEP n'apporte aucun gain par rapport à la baseline Source-Only. Nous avons également pu collecter des preuves supplémentaires attestant du mauvais comportement de S2T et SEP. [3]. Finalement, MDD est meilleur que DANN sur ce transfert. Des expériences supplémentaires sur une variété de transferts ont pu montrer systématiquement une nette supériorité de SYM et MDD sur le reste. Pour cette raison et pour économiser du temps de calcul, on ne considérera que les variantes SYM et MDD dans les expériences qui suivront.

Expériences sur MNIST Algebra Ce nouveau jeu de données permet de construire des transferts particulièrement subtils et informatifs sur les capacités réelles des méthodes d'alignement. Les expériences sont résumées dans la Table 3.2. Nous observons différentes catégories de transferts : $MA \rightarrow MA_{HS}$ est le seul transfert réellement trivial, puisque c'est l'information de la soustraction

3. Pour ce faire, nous avons projeté les descripteurs de dimension 512 dans un espace de dimension 2 à l'aide de l'algorithme de visualisation t-SNE, qui a la particularité de conserver la structure des clusters des distributions projetées. Dans le cas S2T, nous avons observé des distributions alignées, mais mal structurées au lieu de clusters. Dans le cas SEP, nous avons observé un sévère mélange des labels dans la distribution des échantillons cible, malgré un bon alignement des distributions marginales

	S_{ideal}	SO	SEP	S2T	T2S	SYM	MDD	T_{ideal}
$S \to M$	92.0	65.7 ±1.4	63.2 ±3.6	67.3 ±1.3	71.2 ±3.4	75.4 ±2.3	84.6 ±1.6	99.35

TABLE 3.1 – Comparaison des méthodes d'alignement proposées (en performance dans le domaine cible) sur $S \to M$. S_{ideal} est la généralisation sur le domaine source, SO (Source-Only) la performance sur le domaine cible sans alignement, SEP est l'entraînement séparé de deux encodeurs (méthode ADDA), S2T est le mapping de source vers cible, T2S le mapping de cible vers source, SYM l'alignement symétrique, MDD la méthode éponyme et T_{ideal} la performance d'un modèle entraîné sur le domaine cible (en supposant les labels accessibles). (S=SVHN, M=MNIST)

transfer	S_{ideal}	SO	SYM	MDD	T_{ideal}
$MA \to MA_{HS}$	98.4	98.12	99.28	98.9	98.75
$MA_{HS} \to MA$	98.95	79.53	98.96	99.06	99.33
$MA \to MA_{HD}$	99.33	12.59	10.34	10.99	87.52
$MA_{HD} \to MA$	87.52	67.62	86.43	88.56	99.33
$MA_P \to MA_{HD}$	76.65	59.95	60.57	61.89	87.52
$MA_{HD} \to MA_P$	87.52	56.09	86.33	87.07	76.65
$MA \to MA_P$	99.33	63.87	63.12	63.21	76.65
$MA_P \to MA$	76.65	88.06	86.75	86.78	99.33

TABLE 3.2 – Performance dans le domaine cible des deux variantes sélectionnées SYM et MDD. MA=MNIST-Algebra, MA_{HS}=MNIST-Algebra-HideSub, MA_{HS}=MNIST-Algebra-HideDigit, MA_P=MNIST-Algebra-Polluted. Nous reportons par-ailleurs les performances idéales dans le domaine source et cible, qui correspondent à l'écart de généralisation intra-domaine, ce qui sert à quantifier la difficulté d'un jeu de données. Le code couleur indique l'écart de performance relativement à T_{ideal}.

qui est supprimée dans le domaine cible. Dans ce cas, la performance obtenue avec adaptation est similaire à celle obtenue en Source-Only, ce qui prouve que la supervision sur le domaine source pousse le modèle à ignorer l'information provenant des canaux (G, B), ce qui ne pose pas de problème lorsque le support de la distribution d'entrée change.

$MA \to MA_{HD}$ est le transfert le plus difficile, puisque c'est l'information facile qui est occultée dans le domaine cible, provoquant un échec total de l'adaptation. Le modèle n'exploite pas naturellement l'information liée à la soustraction par supervision, on observe donc sans surprise une performance Source-Only proche du hasard. Plus intéressant encore, SYM et MDD n'apportent aucun gain de performance, ce qui prouve qu'une contrainte d'alignement ne permet pas au modèle de découvrir par lui-même la source d'information transférable (dans ce cas, la soustraction). Fait intéressant, le fait de polluer légèrement la source d'information facile encourage naturellement le modèle à exploiter la source d'information difficile, ce qui améliore drastiquement sa performance de transfert vers MA_{HD}, comme on peut l'observer dans le transfert $MA_P \to MA_{HD}$.

De manière générale, on peut caractériser expérimentalement les transferts faciles, qui correspondent aux cas où l'information de classe utile dans le domaine cible est naturellement extraite grâce à la supervision sur le domaine source, mais où le changement de support introduit tout de même du bruit, diminuant ainsi légèrement la performance Source-Only. $MA_{HS} \to MA$, $MA_{HD} \to MA$, $MA_{HD} \to MA_P$ et dans une moindre mesure $MA_P \to MA_{HD}$ appartiennent à cette catégorie. $MA \to MA_P$ est un cas particulier qui ne peut être résolu sans une connaissance à-priori des labels de MA_P.

Expériences sur des transferts classiques Pour conclure, on mène un troisième round d'expériences visant à confirmer les phénomènes observés sur MNIST-Algebra et valider nos conjectures sur une gamme de transferts plus étendue. Les résultats sont donnés sur la Table 3.3.

Dans l'ensemble, on observe à nouveau un écart de performance négligeable entre SYM et MDD. Plus important, on reconnaît facilement les transferts difficiles : SYM et MDD échouent systématiquement en n'apportant aucun gain par rapport à SO quand on transfère d'un jeu de données très simple (ex : M ou U) vers un jeu de données beaucoup plus compliqué (ex : S).

transfer	S_{ideal}	SO	SYM	MDD	T_{ideal}
$S \to M$	91.88	69.79	74.98	83.97	99.38
$M \to S$	99.38	16.6	15.14	15.72	91.88
$U \to M$	98.28	89.12	96.16	95.91	99.06
$M \to U$	99.06	93.05	96.07	96.6	98.28
$U \to S$	97.34	18.46	15.21	12.4	91.41
$S \to U$	91.41	76.72	67.46	72.28	97.34
$MM \to M$	96.41	98.5	97.72	98.33	98.75
$M \to MM$	98.75	40.32	72.54	73.32	96.41
$MM \to S$	96.09	48.73	54.4	52.58	91.41
$S \to MM$	91.41	48.22	60.59	66.78	96.09
$MM \to U$	94.84	90.25	90.79	93.03	98.28
$U \to MM$	98.28	40.76	68.13	67.29	94.84

TABLE 3.3 – Performance dans le domaine cible des deux variantes sélectionnées SYM et MDD. S=SVHN, M=MNIST, U=USPS, MM=MNIST-M.

3.4 Conclusion

L'étude que nous avons mené a permis de comparer expérimentalement cinq stratégies d'adaptation différentes basées sur l'alignement de domaine avec les conclusions suivantes : premièrement, nous avons montré que les différentes variantes du mécanisme d'alignement convergent vers des équilibres différents, donnant lieu à des performances plus ou moins bonnes dans le domaine cible. De plus, nous avons pu observer que si le transfert est suffisamment complexe (par exemple lorsque le domaine source est beaucoup plus pauvre que le domaine cible en termes de diversité visuelle), l'adaptation de domaine n'apporte aucun gain par rapport à Source-Only. Nous pouvons généralement identifier ces transferts lorsque la performance de Source-Only est mauvaise. Enfin, à l'aide d'un nouveau problème de transfert appelé MNIST-Algebra, on a pu montrer quelques preuves étayant le fait que le mécanisme d'alignement des domaines ne découvre pas de descripteurs à proprement parler, mais ne fait qu'ajuster des descripteurs déjà découverts par supervision sur source. Par conséquent, le succès de ces méthodes d'adaptation de domaine semble intrinsèquement lié à la capacité de l'entraînement Source-Only à trouver par-lui même les descripteurs robustes, c'est-à-dire dont l'interprétation sémantique ne dépend pas du domaine de l'image à partir de laquelle il est produit.

Dans le chapitre suivant, on reprendra cette analyse à la lumière de la théorie existante pour l'adaptation de domaine.

Chapitre 4

Théorie : Bornes de généralisation pour l'adaptation de domaine

Sommaire

L'objectif premier de la théorie de l'adaptation de domaine est de trouver des bornes supérieures majorant le risque de classification dans le domaine cible afin d'estimer à quel point un problème de classification peut être adapté à partir d'un domaine source. La plupart d'entre-elles font intervenir un terme mesurant la divergence entre les distributions source et cible. De ce fait, les différences existant entre les différentes bornes découlent principalement de la divergence choisie et du framework statistique sur lesquelles elles se basent : (VC, Rademacher, PAC-Bayes). Ben-David et al. [2010b] fut le premier à proposer une borne de ce type. Elle utilise la divergence $H\Delta H$, qui quantifie à quel point deux hypothèses peuvent désynchroniser leur désaccord d'un domaine vers l'autre. Cette borne a ensuite été raffinée par Zhang et al. [2019] en relaxant la divergence à une seule hypothèse adversaire. Mansour et al. [2009] propose une autre borne basée sur les classifieurs, valable hors de l'hypothèse du covariate shift. Shen et al. [2017] propose une borne basée sur la distance de Wasserstein, qui présente un certain degré de robustesse dans le cas où $\mathcal{S}$ et $\mathcal{T}$ auraient des supports disjoints. Un certain nombre de travaux ont pu proposer des bornes utilisant le framework PAC-bayésien : [Li and Bilmes, 2007, Germain et al., 2015, 2016].

Dans ce chapitre, nous présentons dans un premier temps une série de nouvelles bornes basées sur une relation entre courbes PR et courbes de Lorenz (établie dans Siry et al. [2020b]), ainsi que sur les Phi-divergences. Nous compléterons dans un second temps une série d'observations débutée par Zhao et al. [2019], Johansson et al. [2019], Siry et al. [2020a] montrant les carences du lien entre théorie et algorithmes pratiques.

4.1 Présentation de quelques bornes existantes

Avant de présenter nos contributions, nous présentons dans cette section une série de résultats théoriques importants de la littérature qui nous permettront d'une part d'avoir une idée représentative de la théorie actuelle en adaptation de domaine et d'autre part de situer les apports de cette thèse par-rapport aux travaux existants. Pour des raisons de clarté, nous nous limiterons donc au strict nécessaire. Cependant, pour une présentation exhaustive de l'ensemble des bornes et autres résultats théoriques sur l'adaptation de domaine, nous invitons le lecteur à se référer à Redko et al. [2020].

4.1.1 Bornes de Ben-David

Le résultat auquel on se réfère le plus en adaptation de domaine est sans doute celui de Ben-David et al. (2010). Soit $\mathcal{D}$ un domaine ayant une loi conditionnelle $\mathcal{D}_{Y|X}$ déterministe pour toute valeur de x. On peut donc définir pour un tel domaine $f_{\mathcal{D}} : X \to \{0, 1\}$ la fonction de labellisation associée. Soit $\mathcal{H}$ une classe d'hypothèses, c'est-à-dire une famille de fonctions paramétriques sur X à valeurs dans $\{0, 1\}$.

Définition 3. *(Risque théorique et empirique) On définit le risque 0-1 théorique d'une hypothèse $h \in \mathcal{H}$ sur $\mathcal{D}$ comme la probabilité de désaccord entre h et la loi de labeling vérité-terrain f sur le domaine $\mathcal{D}$.*

$$R_{\mathcal{D}}(h) = E_{x\sim\mathcal{D}_X}[|h(x) - f_{\mathcal{D}}(x)|]$$

On définit également la probabilité de désaccord entre deux hypothèses $h, h' \in \mathcal{H}$ sur le domaine $\mathcal{D}$:

$$R_{\mathcal{D}}(h, h') = E_{x\sim\mathcal{D}_X}[|h(x) - h'(x)|]$$

On notera par ailleurs $\hat{R}_{\mathcal{D}}(h)$ et $\hat{R}_{\mathcal{D}}(h, h')$ la version empirique de ces quantités, calculées à partir d'un nombre fini d'échantillons iid provenant de $\mathcal{D}$. Par la suite, nous serons amenés à définir des risques basés sur d'autres fonctions de coût, pour lesquels nous utiliserons les mêmes conventions de notation.

Les bornes utilisées pour l'adaptation de domaine sont établies en exploitant une mesure de divergence entre les distributions des domaines source et cible. La première borne proposée par Ben-David utilise la distance L^1, aussi appelée distance en variation totale :

Définition 4. *(Distance L^1) Soient $\mathcal{D}_1$ et $\mathcal{D}_2$ deux distributions. La distance L^1 entre ces deux distributions est définie par :*

$$d_1(\mathcal{D}_1, \mathcal{D}_2) = 2 \sup_{B \in \mathcal{B}} |\underset{\mathcal{D}_1}{\mathbf{Pr}}(B) - \underset{\mathcal{D}_2}{\mathbf{Pr}}(B)|$$

Avec $\mathcal{B}$ l'ensemble des parties mesurables par $\mathcal{S}$ et $\mathcal{T}$.

Il est possible de donner une majoration de $R_{\mathcal{T}}(h)$ en exploitant cette divergence dans le cadre du covariate shift :

$$\forall h \in \mathcal{H}, R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_1(\mathcal{S}_X, \mathcal{T}_X) \tag{4.1}$$

Cependant, cette borne possède plusieurs inconvénients : 1) le terme de divergence L^1 ne peut pas être estimé convenablement avec un nombre fini d'échantillons, ce qui rend toute minimisation impossible de facto et 2) cette divergence ne dépend pas de la classe d'hypothèse $\mathcal{H}$. Or, cette dernière propriété est souhaitable, puisque moins $\mathcal{H}$ est expressive, plus les domaines devraient être confondus facilement. Ben-David et al. proposent donc de démontrer un nouveau résultat à partir de la $\mathcal{H}$-divergence :

Définition 5. *($\mathcal{H}$-Divergence) Soient $\mathcal{D}_1$ et $\mathcal{D}_2$ deux distributions et $\mathcal{H}$ un espace d'hypothèses. La $\mathcal{H}$-Divergence entre ces deux distributions est définie par :*

$$d_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = 2 \sup_{h \in \mathcal{H}} |\underset{\mathcal{D}_1}{\mathbf{Pr}}(I(h)) - \underset{\mathcal{D}_2}{\mathbf{Pr}}(I(h))|$$

Avec $I(h)$ l'ensemble dont h est la fonction caractéristique. Autrement dit, le suprémum est atteint avec la meilleure hypothèse de $\mathcal{H}$ capable de faire la distinction entre les deux domaines.

Cette divergence dépend de $\mathcal{H}$ et peut être approchée avec un nombre fini d'échantillons, en vertu des théorèmes de généralisation apportés par la théorie de Vapnik-Chervonenkis 8.3 (VC) :

$$d_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) \leq \hat{d}_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) + 4\sqrt{\frac{dlog(2m)+log(\frac{2}{\delta})}{m}}$$

Avec une probabilité au moins $1 - \delta$. m correspond au nombre d'échantillons et d est la dimension VC de h.

Remarque 4.1. *En apprentissage, la dimension VC quantifie la complexité d'une classe d'hypothèses $\mathcal{H}$. Elle correspond au cardinal de l'ensemble le plus grand que cette famille de fonctions peut pulvériser, i.e. pour tout étiquetage des éléments de cet ensemble, il existe $h \in \mathcal{H}$ ne faisant aucune erreur.*

On définit ensuite $\mathcal{H}\Delta\mathcal{H}$, l'espace des hypothèses qui sont la différence symmétrique de deux hypothèses $h, h' \in \mathcal{H}$

$$g \in \mathcal{H}\Delta\mathcal{H} \iff \exists h, h' \in \mathcal{H}, g = h \oplus h'$$

On définit également la $\mathcal{H}\Delta\mathcal{H}$-Divergence, qui est simplement la $\mathcal{H}$-Divergence associée à la famille d'hypothèses $\mathcal{H}\Delta\mathcal{H}$.

On peut donner une borne supérieure sur $R_{\mathcal{T}}(h)$ en exploitant cette divergence :

$$\forall h \in \mathcal{H}, R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) + \gamma \tag{4.2}$$

Avec $\gamma = \min_{h^\star} R_{\mathcal{S}}(h^\star) + R_{\mathcal{T}}(h^\star)$

Qui est la borne de Ben-David et al. [2010b] la plus citée dans la littérature de l'adaptation de domaine. Cette borne est obtenue en appliquant deux fois l'inégalité triangulaire, qui est valide sur le

risque 0-1, i.e.$R_{\mathcal{D}}(f,g) \leq R_{\mathcal{D}}(f,h) + R_{\mathcal{D}}(h,g)$. La démonstration complète est fournie dans l'annexe B.0.1. Cette borne a directement inspiré certaines méthodes d'alignement, telles que DANN, sans qu'il y ait pour autant un lien complet entre la théorie et la pratique. En effet, la distance $\mathcal{H}\Delta\mathcal{H}$ reste difficile à estimer et à minimiser pour deux raisons : 1) il faut pouvoir trouver un suprémum par-rapport à une instance de $\mathcal{H}\Delta\mathcal{H}$, c'est-à-dire deux instances de $\mathcal{H}$ et 2) la loss $0-1$ n'est pas dérivable, donc pas minimisable par descente du gradient.

4.1.2 Borne de Zhang et al. [2019]

La première difficulté est facile à résoudre : la borne de Ben-David peut-être facilement raffinée en considérant une variante plus relaxée de la $\mathcal{H}$-Divergence :

$$d_{h,\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = \sup_{h' \in \mathcal{H}} \mathop{\mathbf{Pr}}_{\mathcal{D}_2}(h' \neq h) - \mathop{\mathbf{Pr}}_{\mathcal{D}_1}(h' \neq h)$$

Cette divergence n'est autre que la divergence $\mathcal{H}\Delta\mathcal{H}$, allégée d'une valeur absolue et d'un des deux supremum sur $\mathcal{H}$. On obtient alors une divergence dépendante d'une hypothèse $h \in \mathcal{H}$. On obtient alors très facilement la borne suivante :

$$\forall h \in \mathcal{H}, R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) + \gamma \tag{4.3}$$

La démonstration étant très similaire à celle opérée B.0.1, en enlevant une valeur absolue et un suprémum sur $\mathcal{H}$. Cette idée de borne dépendante d'une hypothèse h a été émise par Zhang et al. [2019] sous une forme très similaire, mais en utilisant la marge comme fonction de coût, ce qui résout la deuxième difficulté. Cette borne a l'avantage d'être plus facilement traduisible en algorithme, puisque le terme de divergence peut directement être estimé pour h fixé, en maximisant la quantité par-rapport à une hypothèse adversaire h'. Dans le cas où on utilise la marge, h est à valeurs dans $\mathbb{R}^{|Y|}$ et définit on le risque de marge $R^l_{\mathcal{D}}(h,h')$:

$$R^l_{\mathcal{D}}(h,h') = \mathop{\mathbb{E}}_{x \sim \mathcal{D}_X}[l(h(x), h'(x))].$$

Avec l la marge :

$$\ell(\hat{y}; y) = \left[1 - \frac{1}{\rho}[\delta(\hat{y}; y)]_+\right]_+ = \begin{cases} 1 & \text{si } \operatorname{argmax}(\hat{y}) \neq \operatorname{argmax}(y) \\ \max(0, 1 - \frac{\delta(\hat{y};y)}{\rho}) & \text{sinon} \end{cases} \tag{4.4}$$

où $[x]_+ := \max(0,x)$ et $\delta(\hat{y}; y) = \frac{1}{2}\min_{k \neq k_y} \hat{y}_{k_y} - \hat{y}_k$ est une marge multi-classe entre le score $\hat{y}_{k_y}$ attribué à la plus grande composante de y ($k_y = \operatorname{argmax}_k y_k$) et le second meilleur score $\max_{k \neq k_y} \hat{y}_k$.

La distance devient :

$$d^l_{h,\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = \sup_{h' \in \mathcal{H}} R^l_{\mathcal{T}}(h,h') - R^l_{\mathcal{S}}(h,h')$$

On peut enfin définir la borne de Zhang et al. [2019] :

$$\forall h \in \mathcal{H}, R^l_{\mathcal{T}}(h) \leq R^l_{\mathcal{S}}(h) + d^l_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) + \gamma \tag{4.5}$$

γ s'exprimant évidemment avec les risques de marge.

4.1.3 Borne de Wasserstein

On présente maintenant la borne de Shen et al. [2017], basée sur la distance de Wasserstein. Pour cette borne, nous supposerons que h est une fonction K-lipschitzienne à valeurs dans l'intervalle $[0,1]$. On note pour ce type d'hypothèses le coût $\mathcal{L}_1$: $R^{\mathcal{L}_1}_{\mathcal{D}} = E_{x \sim \mathcal{D}_X}[|h(x) - f_{\mathcal{D}}(x)|]$.

On note par-ailleurs la distance de Wasserstein, que l'on peut exprimer dans sa formulation duale [Villani, 2016] :

$$d_{W_1}(\mathcal{S}_X, \mathcal{T}_X) = \sup_{h \in 1\text{-Lip}} [\mathop{\mathbb{E}}_{\mathcal{S}_X}(h(x)) - \mathop{\mathbb{E}}_{\mathcal{T}_X}(h(x))]$$

Nous avons alors la borne suivante :

$$R_{\mathcal{T}}^{\mathcal{L}_1}(h) \leq R_{\mathcal{S}}^{\mathcal{L}_1}(h) + 2K d_{W_1}(\mathcal{S}_X, \mathcal{T}_X) + \gamma \tag{4.6}$$

Avec $\gamma = \min_{h^\star} R_{\mathcal{S}}^{\mathcal{L}_1}(h^\star) + R_{\mathcal{T}}^{\mathcal{L}_1}(h^\star)$

Puisque basée sur la distance de Wasserstein, cette borne a l'avantage d'être plus informative lorsque les supports sont disjoints.

4.2 De nouvelles bornes pour l'adaptation de domaine

Nous présentons dans cette section notre contribution principale à la théorie de l'adaptation de domaine. Il s'agit d'un ensemble de bornes que nous avons déduit à partir de liens existant entre plusieurs notions courantes en théorie des probabilités et en apprentissage : à savoir les courbes de précision-rappel, les courbes de Lorenz et les Φ-divergences. Après avoir présenté chacune de ces notions en détail, nous présenterons quelques liens existant entre elles. Enfin, nous présenterons les bornes que nous avons pu déduire de cet ensemble de théories et de résultats. Les bornes, ainsi qu'une partie des liens établis, sont des contributions rapportées de notre papier Siry et al. [2020b]. Noter que dans cet article, nous avons également pu établir des liens avec une quatrième notion : celle des frontières de divergence. Ces résultats n'ont pas été exploités au profit de l'adaptation de domaine, par conséquent, nous ne les présenterons pas dans ce manuscrit pour des raisons de clarté.

4.2.1 Contexte et notions sur les courbes de compromis

Dans cette section, nous présentons quelques courbes caractéristiques proposées dans la littérature servant à décrire la similarité existant entre deux distributions P et Q. Nous nous contenterons de résumer les principaux résultats et définitions. Certaines notions seront toutefois sujettes à des adaptations mineures dans le but de simplifier l'exposition des liens existants entre les courbes considérées. Nous mentionnerons explicitement chacune de ces modifications lorsqu'elles seront présentées.

Commençons par rappeler quelques notations, définitions et résultats classiques en théorie de la mesure. À partir de maintenant, $(\Omega, \mathcal{A})$ représente un espace mesurable, et nous noterons par-ailleurs $\mathcal{M}(\Omega)$ l'ensemble des mesures signées sigma-finies, $\mathcal{M}^+(\Omega)$ l'ensemble des mesures positives sigma-finies et $\mathcal{M}_p(\Omega)$ l'ensemble des distributions de probabilités sur cet espace mesurable. La demi-droite réelle achevée est notée $\overline{\mathbb{R}^+} = \mathbb{R}^+ \cup \{\infty\}$.

Définition 6. *Soient μ, ν deux mesures signées. On note* $\mathrm{supp}(\mu)$ *le support*[1] *de μ, $|\mu|$ la mesure en variation totale de μ, $\frac{d\mu}{d\nu}$ la dérivée de Radon-Nikodym de μ par-rapport à ν et $\mu \wedge \nu = \min(\mu, \nu) := \frac{1}{2}(\mu + \nu - |\mu - \nu|)$ (c'est-à-dire la mesure de la plus grande masse commune entre μ et ν [Piccoli et al., 2019]). De plus, comme habituellement, $\mu \ll \nu$ signifie que μ est absolument continue par-rapport à ν.*

Courbes de précision-rappel

Les courbes PR ont été initialement proposées par Sajjadi et al. [2018] pour comparer des distributions discrètes, avant d'être étendues au cas général par Simon et al. [2019]. Nous reprenons la définition de ce dernier, à une correction près[2].

1. Bien qu'une définition précise du support nécessite une topologie, nous ignorons ce problème, car le support ne jouera pas un rôle central dans les applications techniques.

2. Il y a un problème avec la définition d'origine, où $(1, 0)$ et $(0, 1)$ sont toujours dans $\mathrm{PRD}(P, Q)$ (défini plus loin), alors qu'ils ne devraient pas lorsqu'une part de la masse de P est absente de Q et vice versa. Notre correctif consiste à considérer uniquement les distributions μ qui sont absolument continues par-rapport à P et Q.

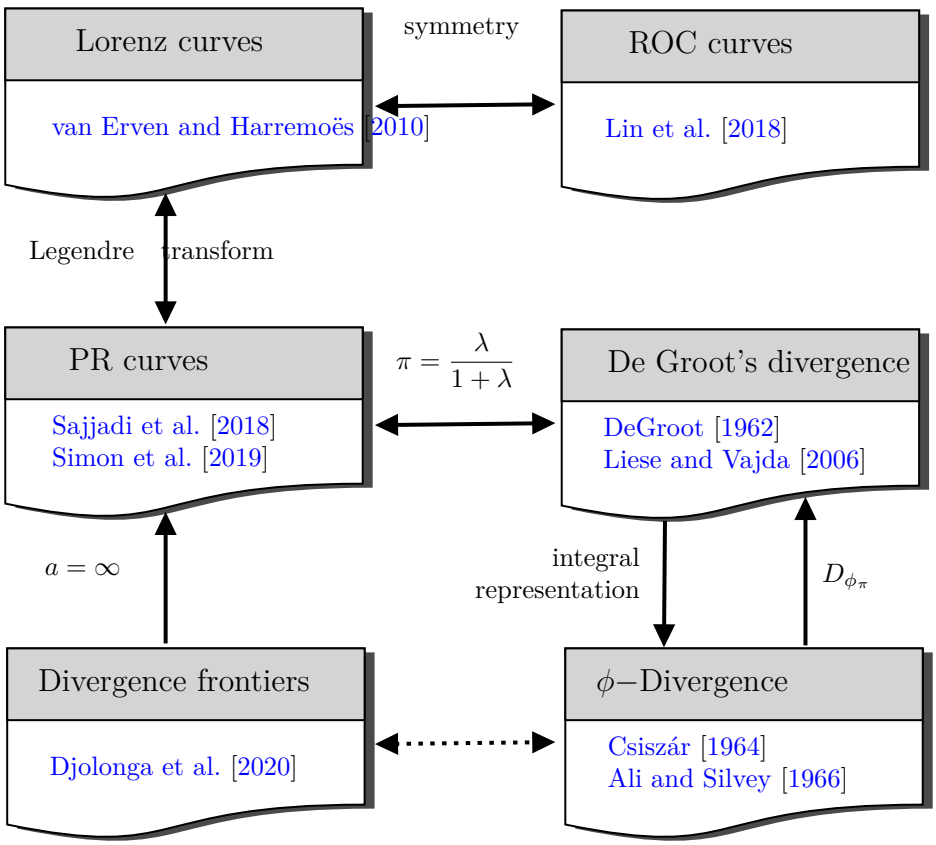

FIGURE 4.1 – Vue d'ensemble des différents travaux et des liens les unissant.

Définition 7. *Soient P, Q deux distributions de $\mathcal{M}_p(\Omega)$. Nous appelons ensemble de précision-rappel $PRD(P,Q)$ l'ensemble des paires de précision-rappel $(\alpha, \beta) \in \mathbb{R}^+ \times \mathbb{R}^+$ telles que*

$$\exists\, \mu \in AC(P,Q), P \geq \beta\mu, Q \geq \alpha\mu\ , \tag{4.7}$$

où $AC(P,Q) := \{\mu \in \mathcal{M}_p(\Omega)/\mu \ll P \text{ et } \mu \ll Q\}$.

La valeur de *précision* α est liée à la proportion de la distribution Q qui correspond aux données de P, et inversement la valeur de *rappel* β est la proportion de la distribution P qui peut être reconstruite à partir de Q. Du fait de l'absence d'ordre naturel sur $[0,1] \times [0,1]$, Simon et al. [2019] propose de s'intéresser au front de Pareto de $\text{PRD}(P,Q)$ définit comme suit.

Définition 8. *La courbe de précision-rappel $\partial PRD(P,Q)$ est l'ensemble des $(\alpha, \beta) \in PRD(P,Q)$ tels que*

$$\forall (\alpha', \beta') \in PRD(P,Q), \alpha \geq \alpha' \text{ ou } \beta \geq \beta'.$$

Notons que cette courbe est clairement le front de Pareto de l'ensemble :

$$\{(\kappa^*(Q|\mu), \kappa^*(P|\mu))/\mu \in AC(P,Q)\}$$

où, en suivant Scott et al. [2013], on définit $\kappa^*(P|\mu) := \max\{\alpha \in [0,1]/\exists \nu \in \mathcal{M}_p(\Omega), P = \alpha\mu + (1-\alpha)\nu\} = \inf_{\substack{A \in \mathcal{A} \\ \mu(A)>0}} \frac{P(A)}{\mu(A)}$ (et de même pour Q).

En réalité, cette frontière est une courbe pour laquelle Sajjadi et al. [2018] ont exposé une paramétrisation, qui a été plus tard généralisée par Simon et al. [2019]. Nous rappelons leur résultat maintenant :

Théorème 4.1. *Soient P, Q deux distributions de $\mathcal{M}_p(\Omega)$ et (α, β) non-négative. Alors, en notant*

$$\forall \lambda \in \overline{\mathbb{R}^+}, \begin{cases} \alpha_\lambda := ((\lambda P) \wedge Q)(\Omega) \\ \beta_\lambda := \left(P \wedge \frac{1}{\lambda} Q\right)(\Omega) \end{cases} \tag{4.8}$$

1. *$(\alpha, \beta) \in PRD(P, Q)$ ssi $\alpha \leq \alpha_\lambda$ et $\beta \leq \beta_\lambda$ où $\lambda := \frac{\alpha}{\beta} \in \overline{\mathbb{R}^+}$.*
2. *En conséquence de quoi la courbe PR peut être paramétrisée par :*

$$\partial PRD(P, Q) = \{(\alpha_\lambda, \beta_\lambda) / \lambda \in \overline{\mathbb{R}^+}\}. \tag{4.9}$$

Dans le théorème précédent, et conformément aux conventions d'usage dans la théorie de la mesure $0 \times \infty = 0$ de sorte que $\alpha_\infty = Q(\text{supp}(P))$ et $\beta_0 = P(\text{supp}(Q))$.

Remarque 4.2. *La paramétrisation précédente montre que la courbe de précision-rappel et intrinsèquement liée à l'information statistique de De Groot [DeGroot, 1962] qui est définie par $\Delta B_\pi(P, Q) := B_\pi(P, P) - B_\pi(P, Q)$ à l'aide de la divergence suivante (que nous appellerons par commodité divergence de De Groot) :*

$$B_\pi(P, Q) := [\pi P \wedge (1 - \pi) Q](\Omega) \tag{4.10}$$

Où $\pi \in [0, 1]$ est une probabilité à-priori arbitraire. En résumé, le lien avec les courbes de précision-rappel est simplement un changement de paramétrisation : $\pi = \frac{\lambda}{1+\lambda}$.

Un autre résultat utile de Simon et al. [2019] liant la frontière au test du rapport de vraisemblance est résumé dans ce qui suit :

Théorème 4.2. *Soient P, Q deux distributions de $\mathcal{M}_p(\Omega)$. Alors*

$$\forall \lambda \in \overline{\mathbb{R}^+}, \begin{cases} \alpha_\lambda = \lambda (1 - P(A_\lambda)) + Q(A_\lambda) \\ \beta_\lambda = 1 - P(A_\lambda) + \frac{Q(A_\lambda)}{\lambda} \end{cases}, \tag{4.11}$$

où les ensembles de rapports de vraisemblance sont définis comme

$$A_\lambda := \left\{ \frac{dQ}{d(P+Q)} \leq \lambda \frac{dP}{d(P+Q)} \right\}. \tag{4.12}$$

Courbes de Lorenz et ROC

Les courbes de Lorenz ont à l'origine été introduites par Lorenz [1905] pour délimiter les inégalités de revenu. Elles mettent essentiellement en évidence à quel point une seule distribution à une dimension diffère de la distribution uniforme. Cette notion a été ensuite généralisée pour caractériser la proximité de deux distributions arbitraires par Harremoës [2004], van Erven and Harremoës [2010].

Définition 9. *Soient P, Q deux distributions de $\mathcal{M}_p(\Omega)$. On définit le diagramme de Lorenz entre P et Q comme*

$$LD(P, Q) = \left\{ \left(\int f dP, \int f dQ \right) / 0 \leq f \leq 1 \right\}, \tag{4.13}$$

où la fonction f doit être mesurable.

Alors, la courbe de Lorenz entre P et Q est définie comme l'enveloppe inférieure du diagramme de Lorenz :

$$F_{LD}^{P,Q}(t) := \inf_{\substack{0 \leq f \leq 1 \\ \int f dP \geq t}} \int f dQ. \tag{4.14}$$

En l'absence d'ambiguité sur les distributions concernées, nous la noterons simplement $F(t)$ plutôt que $F_{LD}^{P,Q}(t)$. On peut facilement montrer que cette courbe est une fonction monotone et convexe.

Si on considère uniquement la famille des fonctions indicatrices dans (4.13), alors on retrouve un sous-ensemble du diagramme de Lorenz, depuis lequel le diagramme de Lorenz peut-être extrait en considérant simplement l'enveloppe convexe la plus proche. Cela relève une équivalence entre les courbes/diagrammes de Lorenz et les notions de région d'effondrement de mode / courbes ROC proposées par Lin et al. [2018]. En effet, Lin et al. [2017] montrent (en Remarque 6) que leur région d'effondrement de mode (MCR) peut être obtenue comme l'enveloppe convexe de l'ensemble de points $(P(A), Q(A))$ où A est n'importe quel ensemble mesurable tel que $Q(A) \geq P(A)$. Par conséquent, le MCR est la moitié supérieure du diagramme de Lorenz lorsqu'on le coupe suivant la diagonale principale (i.e. le segment joignant $(0, 0)$ et $(1, 1)$). Les auteurs définissent ensuite la courbe ROC comme l'enveloppe supérieure du MCR, qui est la transformée symétrique de l'enveloppe inférieure (i.e. la courbe de Lorenz) sur la même diagonale. Afin de respecter l'antériorité, nous ne nous référerons ensuite qu'à la courbe de Lorenz.

De la même façon, restreindre (4.14) aux fonctions indicatrices nécessite une convexification (plus précisément une Γ-régularisation) pour retrouver la courbe de Lorenz. En réalité, grâce au lemme de Neyman-Pearson, on peut même se restreindre aux fonctions indicatrices des ensembles des rapports de vraisemblance A_λ, ce qui à la lumière du Théorème 4.2 souligne un lien subtil avec les courbes de précision-rappel que nous détaillerons plus tard.

4.2.2 Relations entre les courbes de compromis

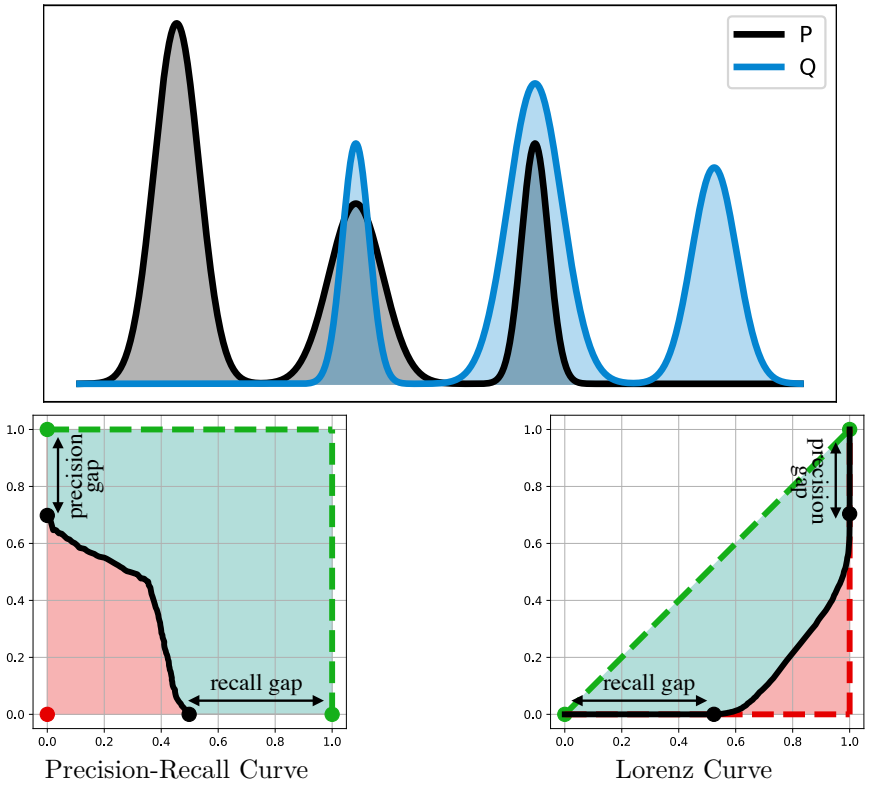

FIGURE 4.2 – haut : représentations graphiques de deux mixtures de gaussiennes P et Q. bas : les courbes de similarité correspondantes. Dans chaque cas, la courbe en pointillés verts illustre le cas extrême où $P = Q$ et le point (la courbe) rouge le cas où $P \perp Q$.

Avant de rentrer dans les détails sur les relations existant entre les courbes présentées dans la partie précédente, nous allons énumérer quelques observations générales marquant leurs différences.

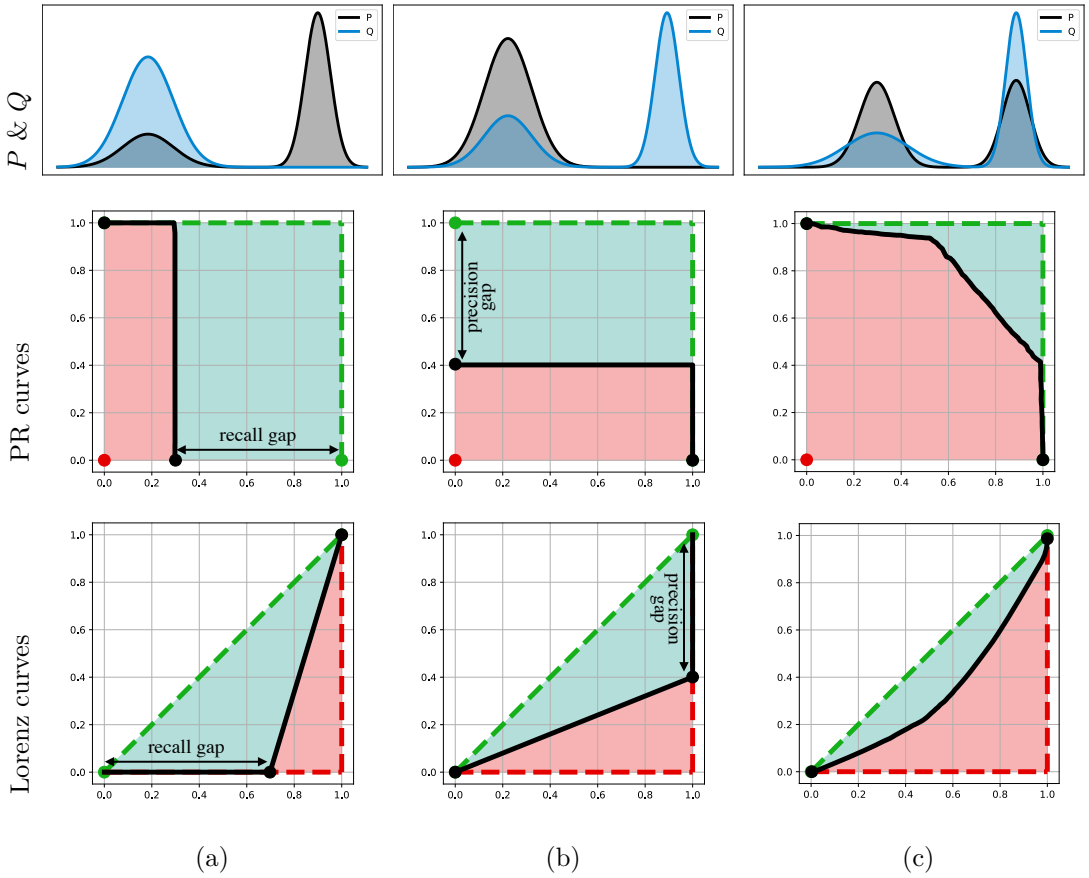

FIGURE 4.3 – Courbes PR et de Lorenz dans différents scénarios : (a) perte d'un mode pure – (b) invention de mode pure – (c) re-pondération des modes. Ligne du haut : les deux distributions P (en noir) et Q (en bleu). Ligne du milieu : courbes PR. Ligne du bas : courbes de Lorenz.

On peut d'abord noter que chaque courbe est sujette à plusieurs propriétés de « régularité » comme la monotonicité, la convexité et le fait d'être bornée. Par exemple, contrairement à la courbe de Lorenz, la courbe PR n'a aucune propriété de convexité. Similairement, la courbe PR et la courbe de Lorenz sont contenues dans le domaine $[0, 1] \times [0, 1]$, alors que les frontières de divergence ne sont en général pas bornées. Enfin, les frontières de divergence et les courbes PR sont décroissantes alors que la courbe de Lorenz est croissante. Malgré ces différences, chacune de ces courbes sert le même objectif et des liens forts existent entre elles. Dans le paragraphe suivant, nous commencerons par étudier quelques cas simples pour décrire le comportement des courbes PR et de Lorenz (les frontières de divergence étant volontairement exclues de ce préambule). Ensuite, dans la sous-section 4.2.2, nous établirons le lien exact entre ces courbes.

Quelques intuitions sur les courbes PR et de Lorenz

Dans ce préambule, nous nous concentrons sur la fonction principale des courbes étudiées : c'est-à-dire leur manière de caractériser la similarité entre P et Q. Pour rendre notre exposé plus concret, on considère un cas illustratif dans la figure 4.2, où P et Q sont deux mixtures de gaussiennes. Pour un choix de courbe donné, on peut considérer deux configurations extrêmes. D'une part, une correspondance parfaite entre P et Q *i.e.* $P = Q$, représentée en pointillés verts. D'autre part, un désaccord total entre P et Q, soit $P \perp Q$ correspondant à une intersection de leurs

supports nulle (ou plus formellement à deux distributions mutuellement singulières) représenté en rouge. Une instance quelconque de la courbe considérée apparaîtra comme un cas intermédiaire. Plus la courbe se rapproche de la caractéristique verte (et donc s'éloigne de la caractéristique rouge), plus P et Q seront similaires.

Pour illustrer le bénéfice d'une courbe de compromis par-rapport à une métrique scalaire, nous illustrons quelques exemples dans la figure 4.3. Les trois exemples sont obtenus en ajustant le positionnement, la largeur et la pondération des modèles de mixtures gaussiennes. Ils correspondent à des scénarios de modes idéaux de déviations entre deux distributions P et Q, c'est-à-dire (a) un effondrement de mode pur, (b) une invention de mode pure et (c) une re-pondération des modes pure. Dans (a) une composante gaussienne de P est absente de Q, ce qui se traduit dans les deux courbes. Dans la courbe PR, il s'observe à travers une chute de rappel qui est illustrée par l'écart horizontal séparant la courbe de la courbe verte. Dans la courbe de Lorenz, il s'observe à travers l'écart horizontal entre le point où la courbe devient positive et l'origine $(0, 0)$. Dans (b) une composante gaussienne supplémentaire est présente dans Q et une fois de plus, ce phénomène est lisible dans les courbes, mais cette fois par des écarts verticaux. Dans (c) P et Q présentent toutes les deux deux composantes situées au même endroit, mais avec des facteurs de mixture différents, et une des deux composantes est plus étalée dans Q. Dans ce scénario, une précision et un rappel maximum peuvent être atteints, mais pas simultanément. Ce compromis est illustré d'une manière différente par chacune des courbes. Dans les deux cas, on n'observe plus d'écarts horizontaux et verticaux correspondant à l'effondrement (resp. à l'invention) de mode. Au lieu de ça, la courbe interpole continûment du rappel total à la précision totale. Le fait qu'une composante gaussienne soit identique dans P et Q se lit aisément dans la courbe PR : en effet, un rappel total peut être atteint pour une précision non-nulle (environ 0.4 dans ce plot). La figure 4.2 montre une combinaison de ces trois scénarios extrêmes et dans ce cas, chacune des deux courbes de compromis illustrent les trois phénomènes.

Précision-Rappel vs Courbes de Lorenz

La question de la relation entre les courbes PR et les courbes de Lorenz/ROC rappelle celle de la comparaison entre les courbes PR et ROC pour la classification binaire Davis and Goadrich [2006]. Notons cependant que malgré leur nom, les courbes PR pour les distributions ne sont pas les mêmes que les courbes PR des classifieurs par rapport de vraisemblance, comme on pourrait le penser. En fait, elles sont composées d'un mélange de taux d'erreurs dudit classifieur. (voir Thm 4.2).

Essentiellement, les courbes PR et de Lorenz sont deux moyens d'exposer les paires $(P(A_\lambda), Q(A_\lambda))$. Pourtant, ces questions n'ont rien de trivial. Étant donné la courbe PR de P et Q, peut-on calculer leur courbe de Lorenz ? Réciproquement, peut-on calculer la courbe PR à partir de celle de Lorenz ? Si on avait une représentation plus complète, par exemple $(\lambda, P(A_\lambda), Q(A_\lambda))$, alors on pourrait facilement calculer à la fois la courbe PR et la courbe de Lorenz, mais dans chaque représentation, au moins une partie de l'information n'est pas connue explicitement :

1. Dans la courbe de Lorenz, λ n'est pas directement accessible, mais nous allons voir qu'il est en réalité caché dans la sous-différentielle de la courbe de Lorenz.
2. Dans la courbe PR, λ peut facilement être calculé comme le rapport $\frac{\alpha_\lambda}{\beta_\lambda}$ mais les valeurs de $P(A_\lambda)$ and $Q(A_\lambda)$ sont mêlées à α_λ, donc on a besoin de les désentrelacer avant de retrouver la courbe de Lorenz. Notons que, pour λ fixé, le système d'équations donné par α_λ et β_λ dans l'Eq 4.11 est toujours sous-déterminé (rang 1) et, de ce fait, n'est pas suffisant pour retrouver les valeurs de $P(A_\lambda)$ et $Q(A_\lambda)$.

Nous allons nous baser sur le Lemme suivant.

Lemme 4.1. *Soient P, Q deux distributions de $\mathcal{M}_p(\Omega)$. Alors $\forall \lambda \in \overline{\mathbb{R}^+}$,*

$$\alpha_\lambda = \min_{0 \leq f \leq 1} \lambda(1 - \int f dP) + \int f dQ \tag{4.15}$$

où les fonctions f sont mesurables.

Démonstration. Voir l'Appendice B.0.2. □

À partir du Lemme 4.1, on peut établir le lien suivant entre la courbe PR et la courbe de Lorenz.

Théorème 4.3. *Soient P et Q deux distributions. Soit $\lambda \in \mathbb{R}^+$. Considérons la courbe de Lorenz F définie dans l'Eq. (4.14), alors,*

$$F^*(\lambda) = \lambda - \alpha_\lambda \tag{4.16}$$

où $F^*(\lambda) = \sup_{t\in[0,1]} \lambda t - F(t)$ est la transformée de Legendre de F.

Démonstration. Soit $\lambda \geq 0$. Montrons que $F^*(\lambda) = \lambda - \alpha_\lambda$. Effectivement , $\forall t \in [0,1]$

$$\begin{aligned}
\lambda t - F(t) =& \lambda t - \inf_{\substack{0\leq f\leq 1 \\ \int f dP \geq t}} \int f dQ = \sup_{\substack{0\leq f\leq 1 \\ \int f dP \geq t}} \lambda t - \int f dQ \\
\leq& \sup_{\substack{0\leq f\leq 1 \\ \int f dP \geq t}} \lambda \int f dP - \int f dQ \\
\leq& \sup_{0\leq f\leq 1} \lambda \int f dP - \int f dQ \\
=& \lambda - \alpha_\lambda \qquad \text{(grâce au Lemme 4.1)}
\end{aligned}$$

Ce qui montre que $\lambda - \alpha_\lambda \geq \sup_{t\in[0,1]} \lambda t - F(t)$. De plus, en fixant $t_\lambda := P(A_\lambda)$

$$\begin{aligned}
\lambda - \alpha_\lambda =& \lambda - (\lambda(1 - P(A_\lambda)) + Q(A_\lambda)) \\
=& \lambda P(A_\lambda) - Q(A_\lambda) = \lambda t_\lambda - F(t_\lambda)
\end{aligned}$$

où nous avons exploité le fait que si $t_\lambda = P(A_\lambda)$ alors $F(t_\lambda) = Q(A_\lambda)$ (résultat induit par le Lemme de Neyman-Pearson). Donc, $\lambda - \alpha_\lambda = \sup_{t\in[0,1]} \lambda t - F(t) = F^*(\lambda)$. □

Remarque 4.3. *Le Théorème 4.3 apporte de nombreuses perspectives intéressantes concernant le lien entre les courbes PR et les courbes de Lorenz.*

1. *D'abord, puisque la transformée de Legendre est une involution bijective, les courbes PR et de Lorenz sont théoriquement équivalentes.*
2. *De plus, en fixant $t_\lambda := P(A_\lambda)$ et en se basant sur l'identité de Fenchel, on obtient $\lambda \in \partial F(t_\lambda)$, ce qui fournit théoriquement un moyen d'extraire l'information manquante à partir du moment où on peut calculer la sous-différentielle de la courbe de Lorenz.*
3. *Plus concrètement, le théorème nous donne un moyen pratique de calculer α_λ à partir de la courbe de Lorenz. En effet, étant donné λ, α_λ peut être obtenu en résolvant le problème convexe 1D suivant :*

   $$\alpha_\lambda = \lambda - F^*(\lambda) = \min_{t\in[0,1]} F(t) + \lambda(1 - t)$$

   Ce qui peut être fait de façon très efficace avec la méthode de la bisection si la sous-différentielle de F est disponible. On peut dans le cas contraire utiliser des méthodes ne requérant pas la différentielle, comme la méthode du nombre d'or. Alors β_λ est obtenu par $\frac{\alpha_\lambda}{\lambda}$.
4. *Dans l'autre sens, étant donné $t \in [0,1]$, on peut résoudre pour $F(t)$ en considérant le problème concave 1D suivant :*

   $$\begin{aligned}
   F(t) = F^{**}(t) =& \sup_{\lambda\in\mathbb{R}^+} \lambda t - F^*(\lambda) \\
   =& \sup_{\lambda\in\mathbb{R}^+} \alpha_\lambda + \lambda(t - 1).
   \end{aligned}$$

4.2.3 Première borne basée sur les courbes PR/Lorenz

Nous avons défini dans les sections précédentes les différentes courbes de compromis, puis montré les liens qui les unissaient, certains de ces liens relevant de notre contribution. Nous allons maintenant voir comment exploiter ces objets et relations pour enrichir la théorie de l'adaptation de domaine en proposant notamment une première borne, que l'on peut exprimer au choix soit avec les courbes de Lorenz, soit avec les courbes PR.

Pour ce faire, on commence par réexaminer une borne standard de l'adaptation de domaine. On nomme P et Q les deux distributions source et cible définies sur l'espace joint échantillons-labels $\Omega = X \times Y$. Etant donné une hypothèse h, on rappelle la première borne de Ben-David, présentée dans l'équation 4.1 :

$$R_Q(h) := \int \mathbf{1}_{h(x)\neq y} dQ(x,y) \leq R_P(h) + d_1(P_X, Q_X) \tag{4.17}$$

Dans l'hypothèse du covariate shift, la borne peut s'exprimer en fonction des distributions marginales sur X. On rappelle les deux désavantages pratiques de cette borne : 1) elle ne peut pas être estimée à partir d'un nombre fini d'échantillons et 2) elle ne dépend pas de la classe d'hypothèses $\mathcal{H}$ à laquelle appartient h (qui pourrait par exemple être une classe de faible dimension VC ou de faible complexité de Rademacher). Par conséquent, la borne en question est trop pessimiste, puisqu'elle tient pour acquis le fait que l'ensemble des erreurs $\{h(x) \neq y\}$ peut être n'importe quel ensemble mesurable. Ce n'est généralement pas le cas quand on considère les restrictions qui s'appliquent à la classe de h. Les auteurs de Ben-David et al. [2010b] ont ensuite produit une borne plus adaptée, basée sur la divergence $\mathcal{H}\Delta\mathcal{H}$ qui corrige les deux défauts de la précédente, qui fut suivie par de nombreux autres travaux. Nous invitons les lecteurs à lire le survey [Redko et al., 2020] pour une revue exhaustive et à jour de ces contributions.

Une borne plus raffinée

Malgré ses défauts, on propose de réexaminer la borne de l'Eq. 4.17 et d'en proposer une version optimisée qui dispose d'une interprétation intuitive. On commence par démontrer une borne basée sur la courbe de Lorenz entre P et Q, avant de l'exprimer sous une forme plus proche de celle de l'Eq 4.17 en utilisant la dualité entre les courbes PR et les courbes de Lorenz que nous avons établie.

Proposition 4.1. *La courbe de Lorenz fournit une borne supérieure pour l'adaptation de domaine.*

$$R_Q(h) \leq 1 - F(1 - R_P(h)) \tag{4.18}$$

Démonstration.

$$\begin{aligned}
F(1-R_P(h)) &= \inf_{\substack{g \text{ measurable} \\ 0\leq g\leq 1 \\ \int g dP \geq 1 - \int \mathbf{1}_{h\neq f} dP}} \int g dQ \\
&= \inf_{\substack{g \text{ measurable} \\ 0\leq 1-g\leq 1 \\ \int (1-g) dP \leq \int \mathbf{1}_{h\neq f} dP}} \int 1-(1-g) dQ \\
&= 1 - \sup_{\substack{g' \text{ measurable} \\ 0\leq g'\leq 1 \\ \int g' dP \leq \int \mathbf{1}_{h\neq f} dP}} \int g' dQ
\end{aligned}$$

On considérant le cas particulier : $g' = \mathbf{1}_{h\neq f}$ on a

$$F(1 - R_P(h)) \leq 1 - \int \mathbf{1}_{h\neq f} dQ = 1 - R_Q(h)$$

□

Cette même borne peut être exprimée avec la paramétrisation PR en vertu du Théorème 4.3 de dualité.

Proposition 4.2.

$$R_Q(h) \leq \lambda^* R_P(h) + (1 - \alpha_{\lambda^*}) \tag{4.19}$$

with $\lambda^* = \underset{\lambda \in \mathbf{R}^+}{\operatorname{argmax}} \{\alpha_\lambda - \lambda R_P(h)\}$

Démonstration. Ce résultat découle du Théorème. 4.3. □

La première partie de notre borne supérieure correspond aux erreurs ayant lieu dans le support commun de P and Q. Dans ce cas, le taux d'erreur est contrôlé dans le domaine source, et est par conséquent également contrôlé dans le domaine cible. Le facteur d'amplification λ^* tient compte du fait que la masse commune entre P and Q est présente dans différentes amplitudes dans les deux domaines. La seconde partie correspond aux erreurs ayant lieu dans le domaine cible, mais hors du support du domaine source. On n'a aucun contrôle sur cette erreur, et devons par conséquent considérer le pire cas où h a toujours tort. Par conséquent, le seul moyen de maintenir ce terme sous contrôle est de faire un certain nombre d'hypothèses sur h, ainsi que sur la distribution des labels, c'est à dire la classe des hypothèses et la classe des concepts.

Commentaires

La dernière forme de notre borne (Eq. (4.19)) révèle un lien fort avec l'Eq. (4.17). En particulier, si $\lambda^* = 1$ alors, en notant que $\alpha_1 = 1 - \frac{1}{2}\|P - Q\|_{TV}$, alors les deux bornes sont quasiment identiques La seule différence réside dans un facteur $\frac{1}{2}$ qui joue en notre faveur et qui découle du fait que nous exploitons explicitement la positivité de l'erreur. De plus, de manière générale, 1 n'est pas le λ^* optimal et notre borne est en réalité encore plus fine. C'est à ce moment que l'utilisation d'une courbe de trade-off s'avère utile : nous obtenons virtuellement une borne pour chaque valeur de λ et nous pouvons choisir la plus fine. Considérons l'exemple simple de la Fig. 4.2 où l'on peut lire la valeur de α_λ sur l'axe y à l'emplacement où la courbe PR rejoint la ligne de l'équation $\alpha = \lambda\beta$. Dans cet exemple, $\alpha_1 \approx 0.38$ ce qui signifie que $\|P - Q\|_{TV} = 2(1 - \alpha_1) \approx 1.24$. Donc la borne de l'Eq. (4.17) est plus grande que 1 et est par conséquent non-informative. D'autre part, l'Eq. (4.11) avec $\lambda = 1$ donne $R_Q(h) \leq R_P(h) + (1 - \alpha_1) \approx R_P(h) + 0.62$ ce qui est informatif à partir du moment où $R_P(h) < 0.38$. Cette condition est facilement respectée dans des cas concrets puisque $R_P(h)$ est l'erreur dans le domaine source, que l'on peut maintenir sous contrôle à un terme de généralisation près grâce à la supervision. Plus important, on peut examiner comment le compromis optimal entre les deux termes de la borne peut mener à une borne bien plus fine. En fait, étant donné la convexité de $1 - \alpha_\lambda$, le λ optimal est caractérisé par la condition du point critique d'ordre 1, c'est-à-dire $R_P(h) \in \partial_\lambda \alpha_\lambda$. Notons qu'il n'est pas trivial de lire la dérivée de α_λ directement à partir de la courbe PR, mais dans cet exemple, cette dérivée est bien plus grande que 1 et donc plus grande que $R_P(h)$. Cela signifie que le λ optimal est éloigné de 1.

De plus, comme expliqué plus haut, notre borne peut facilement être comprise en termes de masse partagée vs masse séparée entre P et Q. Malgré ces avantages, il faut rappeler qu'elle souffre des mêmes limites en ce qui concerne son estimation par échantillons finis et sa nature pessimiste par-rapport à la régularité de l'ensemble des erreurs de h. Les courbes PR et de Lorenz présentent certaines similarités avec des outils développés pour l'adaptation de domaine, ce qui nous permet de faire quelques observations supplémentaires. Par exemple, on peut exprimer les courbes PR comme des courbes de compromis calculées à partir de ratios de pondération [3] à l'instar de Ben-David and Urner [2012]. Inspirés par leur travail, il apparaît naturel de vouloir restreindre la classe des espaces mesurables « admissibles » dans les courbes PR et obtenir des bornes plus utiles tout en retenant

3. Cette propriété découle du lien avec les frontières de divergence, que nous explicitons dans notre article Siry et al. [2020b].

la notion de compromis optimal. Un travail similaire pourrait être accompli dans la représentation duale en restreignant la classe de fonctions considérées dans le diagramme de Lorenz. De cette façon, on pourrait tirer parti des restrictions sur la classe d'hypothèse et obtenir des bornes similaires à celles dérivées des métriques de probabilité intégrales (IPM) [Redko et al., 2020]. Néanmoins, étant donné la capacité bien connue des réseaux de neurones profonds à sur-apprendre des labellisations aléatoires Zhang et al. [2016], exploiter les théories de complexité classiques n'est probablement pas suffisant pour obtenir des bornes qui sont représentatives des problèmes d'adaptation de domaine actuels. Il apparaît inévitable de devoir exploiter une sorte de « biais implicite » lié à la procédure d'optimisation. Faire ce travail tout en se reposant sur les courbes de compromis semble être une piste de recherche prometteuse qui sera certainement à l'origine de nombreuses bornes pratiques pour l'adaptation de domaine.

4.2.4 Deuxième borne basée sur les ϕ-divergences

On propose maintenant une seconde borne basée sur les ϕ-divergences (également appelées f-divergences). Nous commencerons naturellement par introduire cette notion classique en théorie des probabilités et en apprentissage. Nous verrons ensuite de quelle façon elle peut permettre d'étendre la notion de diagramme de Lorenz, ce qui nous permettra d'introduire une nouvelle borne d'adaptation de domaine. Une tentative de démontrer une telle borne a déjà été menée par Acuna et al. [2021], mais ces travaux comportent de nombreuses erreurs et ne peuvent en l'état pas être considérés comme valides.

Définition et propriétés des ϕ-divergences

Nous considérerons dans toute la suite des fonctions convexes $\phi : \mathbb{R} \to \mathbb{R}$, semi-continues inférieurement (l.s.c.) et vérifiant les conditions suivantes :

(A0) $\text{dom}(\phi) \cap]-\infty, 0[= \emptyset$ et $]0, +\infty[\subset \text{dom}(\phi)$

(A1) $\phi(1) = 0$

(A2) $0 \in \partial\phi(1)$ et $\partial\phi(1)$ est symétrique autour de 0 (ie $\phi'_-(1) = -\phi'_+(1)$)

(A3) ϕ est strictement convexe en 1

Nous appellerons Φ_1 l'ensemble des fonctions convexes l.s.c. vérifiant ces trois conditions.

Définition 10 (ϕ-divergence)**.** *Soient $\phi \in \Phi_1$ et μ et ν deux mesures positives. Alors,*

$$D_\phi(\mu\|\nu) := \int \phi\left(\frac{d\mu}{d\nu}\right) d\nu + \phi(0)\nu(\mu = 0) + \phi^\diamond(0)\mu(\nu = 0) \tag{4.20}$$

avec $\phi(0) := \lim_{u\to 0} \phi(u)$ et $\phi^\diamond(0) := \lim_{u\to 0} u\phi(\frac{1}{u})$

Remarque 4.4. *Prenons le temps de donner quelques éléments de réflexion sur les quatre conditions définissant Φ_1. La première partie de la condition (A0) restreint la ϕ divergence aux mesures positives (ou aux mesures de même signe), et enlève par conséquent tout choix subjectif sur la valeur de $\phi(\frac{d\mu}{d\nu})$ lorsque $\frac{d\mu}{d\nu} < 0$. Prises ensemble, les trois autres conditions définissant Φ_1 imposent $D_\phi(\mu\|\nu) \geq 0$ avec l'égalité ssi $\mu = \nu$. En effet, en utilisant les hypothèses (A1) et (A2), on remarque que $\forall u \in \mathbb{R}$, $\phi(u) \geq \phi(1) = 0$, et la dernière hypothèse implique $\phi(u) = 1$ ssi $u = 1$. La condition (A2) n'est pas toujours imposée dans la littérature lorsqu'on travaille avec des probabilités (puisque dans ce cas, $\mu = P, \nu = Q$ sont des distributions et $D_\phi(P\|Q)$ est invariants aux modifications de la forme $\tilde{\phi}(u) = \phi(u) - \frac{\phi'_+(1)+\phi'_-(1)}{2}(u-1)$).*

Remarque 4.5. *Le dual de Legendre-Fenchel de ϕ jouera un rôle important dans les formes variationnelles de D_ϕ [Nguyen et al., 2010]. Il est donc intéressant de commenter l'impact des hypothèses $\phi \in \Phi_1$ on ϕ^*. En fait, $\phi(1) = \min_u \phi(u)$ signifie exactement que $\phi^*(0) = -\phi(1)$. De*

Divergence	$\phi(u)$	$\phi'(u)$	$\mathrm{dom}^{v_0\rightarrow}(\phi^*)$	$\phi^{*\prime}(v)$	$\phi^*(v)$
KL	$u\log(u)-(u-1)$	$\log(u)$	$\mathbb{R}$	e^v	e^v-1
rKL	$-\log(u)+(u-1)$	$1-\frac{1}{u}$	$]-\infty,1[$	$\frac{1}{1-v}$	$-\log(1-v)$
JS	$-(u+1)\log\frac{1+u}{2}$ $+u\log u$	$\log(\frac{2u}{1+u})$	$]-\infty,\log(2)[$	$\frac{e^v}{2-e^v}$	$-\log(2-e^v)$
χ^2_{Pearson}	$(u-1)^2$	$2(u-1)$	$[-2,+\infty[$	$\frac{v}{2}+1$	$\frac{v^2}{4}+v$
Hellinger	$(\sqrt{u}-1)^2$	$1-\frac{1}{\sqrt{u}}$	$]-\infty,1[$	$\frac{1}{(1-v)^2}$	$\frac{v}{1-v}$
TV	$\lvert u-1\rvert$	$\mathrm{sign}(u-1)$	$[-1,1]$	1	v

TABLE 4.1 – Quelques ϕ-divergences standard. Pour chaque choix de ϕ on fournit les propriétés correspondantes dans l'ordre où elles sont naturellement déduites : on calcule d'abord $\phi'(u)$ puis on déduit le domaine "significatif" de ϕ^*, c'est-à-dire $\mathrm{dom}^{v_0\rightarrow}(\phi^*)$, ensuite $\phi^{*\prime}(v)$ est déterminée comme l'inverse de $\phi'(u)$ et $\phi^*(v)$ est déterminée comme étant l'anti-dérivée vérifiant $\phi^*(0)=0$. Notons que ϕ^* et sa dérivée sont uniquement données sur leur domaine significatif.

surcroît, puisque $\phi(1)=0$, cela signifie que $\phi^*(0)=0$. Symétriquement, lorsque $0\in\mathrm{dom}(\phi)$, choisissons $v_0\in\partial\phi(0)$, alors $\phi^*(v_0)=\min_v\phi^*(v)$. De plus, puisque ϕ est minimisée en $u=1$, alors elle est décroissante partout où $u<1$ et est croissante sinon (par convexité). En particulier, $\partial\phi(0)\subset[-\infty,0]$ et de ce fait, le minimum de ϕ^* peut uniquement être atteint en un point $v_0\leq 0$ (il est également possible que l'infimum ne soit pas atteint si $\partial\phi(0)=\{-\infty\}$ ou si $0\notin\mathrm{dom}(\phi)$). Pour conclure, on remarquera que la définition de ϕ sur $]-\infty,0]$ n'est pas importante pour $D_\phi(\mu\|\nu)$ (en effet, puisque μ et ν sont supposées positives, alors $\frac{d\mu}{d\nu}\geq 0$). Par conséquent, D_ϕ n'est également pas influencée par les valeurs prises par $\phi^*(v)$ pour $v<v_0:=\sup\{v\in\mathrm{argmin}(\phi^*)\}$. Dans tous les cas, puisque l'hypothèse (A0) impose $\phi(\frac{d\mu}{d\nu})=+\infty$ lorsque $\frac{d\mu}{d\nu}<0$ alors $\phi^*(v)=\phi^*(v_0)$ lorsque $v<v_0$. Pour simplifier les formules concrètes, comme par exemple dans la table 4.1, on se concentrera sur $\phi^*(v)$ sur la restriction suivante de son domaine $\mathrm{dom}^{v_0\rightarrow}(\phi^*):=\cup_{u\geq 0}\partial\phi(u)=\mathrm{dom}(\phi^*)\cap[v_0,+\infty[$. Il inclut au moins $]v_0,\phi'_+(+\infty)[$, et puisque ϕ est strictement convexe en 1, $\phi'_+(\infty)>0$ (en effet $\phi'_+(\infty)\geq\phi'(1)\geq 0$ et si par l'hypothèse de symétrie de (A2) $\phi'_+(1)=-\phi'_-(1)=0$ et ensuite la convexité stricte implique que $\phi'_+(\infty)>\phi'_+(1)$. Alors $\mathrm{dom}(\phi^*)\supset]v_0,0]$.

Diagrammes de Lorenz généralisés associés avec une f-divergence

Dans Nguyen et al. [2010], les auteurs montrent qu'étant donné une classe de fonctions mesurables, bornées $\mathcal{F}$,

$$\begin{aligned} D_\phi(P\|Q)&=\int\phi(\frac{dP}{dQ})dQ=\int\sup_{v\in\mathbb{R}}v\frac{dP}{dQ}-\phi^*(v)dQ\\ &\geq\sup_{f\in\mathcal{F}}\int fdP-\int\phi^*(f)dQ\end{aligned}\tag{4.21}$$

avec l'égalité ssi $\partial\phi(\frac{dP}{dQ})\cap\mathcal{F}\neq\emptyset$, ce qui signifie qu'il existe $f\in\mathcal{F}$ telle que $f\in\partial\phi(\frac{dP}{dQ})$ (ou de façon équivalent grâce à la dualité $\frac{dP}{dQ}\in\partial\phi^*(f)$). Notons qu'avec les conventions standard, $0\in\partial\phi(1)$ et puisque la convexité implique que la sous-différentielle de ϕ augmente, alors la condition $f\in\partial\phi(\frac{dP}{dQ})$ impose que $f\leq 0$ lorsque $\frac{dP}{dQ}\leq 1$ et $f\geq 0$ sinon. Cette forme variationnelle peut être utilisée pour créer une sorte de diagramme de Lorenz associé avec D_ϕ, défini comme l'ensemble des paires $\{(\int\phi^*(f)dP,\int fdQ)\}$. Ensuite, la frontière supérieure de cette région caractérise la proximité entre P et Q, et en particulier $D_\phi(Q\|P)$ est la distance verticale maximale entre la courbe et la diagonale. Pour des raisons techniques (principalement assurer un diagramme convexe), la définition du diagramme de Lorenz est légèrement plus compliquée (voir Figure 4.4 pour une illustration de sa construction [4]).

4. Noter que cette construction peut être adaptée à la formulation variationnelle plus fine de D_Φ proposée dans Ruderman et al. [2012] et Agrawal and Horel [2020]

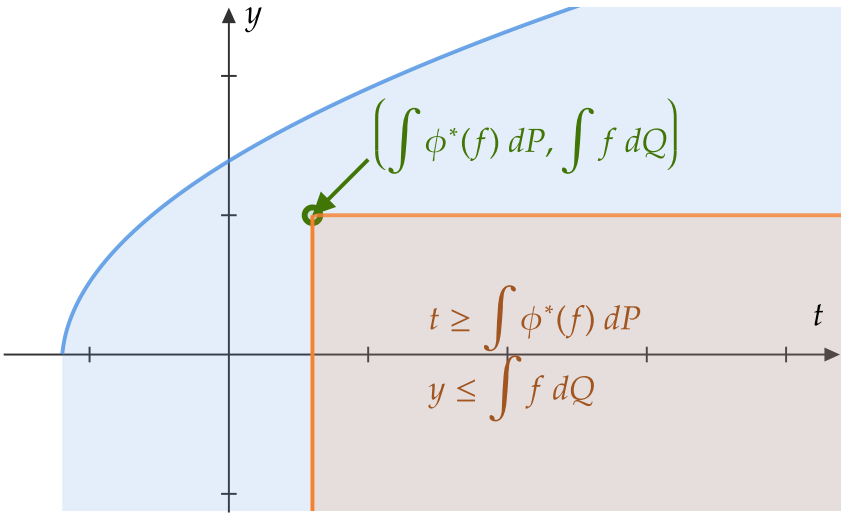

FIGURE 4.4 – Construction du diagramme de Lorenz généralisé.

Définition 11 (diagramme de Lorenz étendu)**.** *Soient $\phi \in \Phi_1$ et P, Q deux distributions. Le ϕ-diagramme de Lorenz entre P et Q est défini comme :*

$$LD_\phi(P,Q) = \left\{(t,y) \in \mathbb{R}^2 / \exists f : \Omega \to \mathbb{R}, t \geq \int \phi^*(f) dP, y \leq \int f dQ\right\}, \tag{4.22}$$

La courbe de Lorenz étendue est définie comme son enveloppe supérieure :

$$\bar{F}_\phi(t) := \sup\{y/(t,y) \in LD_\phi(P,Q)\} \tag{4.23}$$

Remarque 4.6. *Une tentative de démontrer une borne de généralisation pour l'adaptation de domaine basée sur la forme variationnelle des ϕ-divergences a été récemment publiée dans Acuna et al. [2021]. Ils proposent effectivement une borne similaire à l'équation (4.17) dans [Acuna et al., 2021, Lemme 1] pour une fonction de coût $\ell(\hat{y}; y) \in \mathrm{dom}(\phi^*)$. En notant,*

$$R^\ell_Q(h) := \int \ell(h(x); y) dQ(x,y) \tag{4.24}$$

leur borne est donnée comme suit,

$$R^{\phi^* \circ \ell}_Q(h) := \int \phi^*(\ell(h(x); y)) dQ(x,y) \leq R^\ell_P(h) + D_\phi(P\|Q) \tag{4.25}$$

en remarquant que sous les hypothèses standard $\phi(1) = 0$ alors $\phi^*(v) \geq v$, la borne précédente implique alors :

$$R^\ell_Q(h) \leq R^{\phi^* \circ \ell}_Q(h) \leq R^\ell_P(h) + D_\phi(P\|Q) \tag{4.26}$$

Malheureusement, cette borne est erronée. On peut le montrer en considérant la famille des ϕ-divergences associées avec $\phi_s = s\phi$ où $s > 0$. En effet, si la borne était vraie, puisque $D_{\phi_s}(P\|Q) = sD_\phi(P\|Q)$, dans le cas où $s \to 0$, cette borne impliquerait que $R^\ell_Q(h) \leq R^\ell_P(h)$, ce qui est évidemment faux en général[5].

5. L'origine de l'erreur est une substitution fallacieuse de valeurs absolues par un suprémum : $D_\phi(P\|Q) = \sup_{f \in \mathrm{dom}(\phi^*)} \int f dP - \int \phi^*(f) dQ = |\sup_{f \in \mathrm{dom}(\phi^*)} \int f dP - \int \phi^*(f) dQ| \leq \sup_{f \in \mathrm{dom}(\phi^*)} |\int f dP - \int \phi^*(f) dQ|$: la dernière inégalité est en général stricte (et dans les faits la partie droite est souvent infinie), alors que Acuna et al. [2021] ont utilisé une égalité.

A l'instar de ce qui a été fait dans la Proposition 4.1, il est possible de corriger cette borne en utilisant le diagramme de Lorenz associé à la ϕ-divergence.

Proposition 4.3. *Let $\phi \in \Phi_1$, $\ell(\hat{y}; y) \in \mathrm{dom}(\phi^*)$. La courbe de Lorenz permet d'obtenir la borne suivante pour l'adaptation de domaine :*

$$R^{\ell}_{Q}(h) := \int \ell(h(x); y) dQ(x, y) \leq R^{\phi^* \circ \ell}_{\lambda P}(h) + D_{\phi}(Q \| \lambda P) \tag{4.27}$$

Démonstration. On commence par proposer une borne pour l'enveloppe supérieure du diagramme de Lorenz. Pour $t \in \mathrm{dom}(\phi^*)$

$$\begin{aligned}
\bar{F}_{\phi}(t) =& \sup\{y \in \mathbb{R} / \exists f \in \mathrm{dom}(\phi^*), y \leq \int f dQ, t \geq \int \phi^*(f) dP\} \\
=& \sup_{f / \int \phi^*(f) dP \leq t} \int f dQ = \sup_{f} \inf_{\lambda \geq 0} \int f dQ - \lambda (\int \phi^*(f) dP - t) \\
\leq& \inf_{\lambda \geq 0} \sup_{f} \int f dQ - \lambda (\int \phi^*(f) dP - t) \\
=& \inf_{\lambda \geq 0} \lambda t + \sup_{f} \int f dQ - \int \phi^*(f) d(\lambda P) \\
=& \inf_{\lambda \geq 0} \lambda t + D_{\phi}(Q \| \lambda P)
\end{aligned} \tag{4.28}$$

où l'inégalité dans la troisième ligne est triviale. Notons également que dans la dernière identité, on a besoin d'étendre la forme variationnelle de D_ϕ de Nguyen et al. [2010] lorsque appliquée à Q et λP (qui est une mesure positive, mais pas une distribution de probabilité en général). La démonstration de Nguyen et al. [2010] peut facilement être étendue à cette situation. Par conséquent,

$$\begin{aligned}
R^{\ell}_{Q}(h) :=& \int \ell(h(x); y) dQ(x, y) \\
\leq& \bar{F}_{\phi}(R^{\phi^* \circ \ell}_{P}(h)) \\
\leq& \inf_{\lambda > 0} \lambda R^{\phi^* \circ \ell}_{P}(h) + D_{\phi}(Q \| \lambda P)
\end{aligned} \tag{4.29}$$

□

Remarque 4.7. *Pour ce qui est du diagramme de Lorenz standard, cette borne permet de faire un compromis entre l'erreur faite dans le domaine source et une meilleure couverture entre la distribution cible Q et la distribution source non-normalisée λP. En utilisant le choix sous-optimal $\lambda = 1$, on retrouve une borne similaire à l'Equation erronée (4.26) tirée de Acuna et al. [2021] :*

$$R^{\ell}_{Q}(h) \leq R^{\phi^* \circ \ell}_{P}(h) + D_{\phi}(Q \| P) \tag{4.30}$$

Dans cette borne corrigée, la fonction de coût source ℓ est remplacée par $\phi^* \circ \ell$, qui est toujours plus grande. Ceci étant dit, dans un scénario typique d'adaptation de domaine, ℓ est en moyenne faible dans le domaine source (distribution P). Puisqu'en pratique, ℓ est également positive, être en moyenne faible signifie être faible avec une probabilité élevée sous P (ce qui peut être montré en utilisant par exemple l'inégalité de Markov). De plus, puisque $\phi \in \Phi_1$, $0 \in \partial\phi(1)$ ce qui par dualité implique que $1 \in \partial\phi^*(0)$ ce qui montre que $\phi^*(0) = -\phi(1) = 0$ (pour peu qu'on ait bien sûr $0 \in \mathrm{dom}(\phi^*)$). De plus, ϕ^* est souvent dérivable en 0, donc son développement de Taylor est $\phi^*(v) = v + o(v)$, par conséquent, $R^{\phi^* \circ \ell}_{P}(h)$ et $R^{\ell}_{P}(h)$ sont similaires (pour peu que $R^{\ell}_{P}(h)$ soit faible). Pour étayer cette assertion plus précisément, intéressons-nous au développement de Taylor

avec reste intégral. Si $\phi^{*\prime}(0) = 1$ *alors, on a le développement de Taylor suivant (voir [Liese and Vajda, 2006, Théorème 1]) :*

$$\phi^*(v) = v + \int_0^v (v-s) d\phi_+^{*\prime}(s)$$

En particulier, si la courbure de $\phi^*(v)$ *est bornée :* $\forall v \in [0,1], \phi^{*\prime\prime}(v) \in [\underline{\kappa}, \bar{\kappa}]$*, alors on a :*

$$\frac{1}{2}\underline{\kappa}\sigma^2 \leq R_P^{\phi^* \circ \ell}(h) - R_P^{\ell}(h) \leq \frac{1}{2}\bar{\kappa}\sigma^2$$

où $\sigma^2 := \int \ell(h(x), y)^2 dP$ *est généralement assez faible dans un scénario concret. Noter que le raisonnement précédent est uniquement valable si la courbure de* ϕ^* *est contrôlée. Le contre-exemple consistant à remplacer* ϕ *par une version remise à l'échelle* $\phi_s = s\phi$ *ne respecterait plus la condition si* $s \to 0$*. En effet,* $\phi_s^*(v) = s\phi^*(\frac{v}{s})$*, et donc* $\phi_s^{*\prime\prime}(v) = \frac{1}{s}\phi^{*\prime\prime}(\frac{v}{s})$*.*

Comme indiqué dans la littérature de l'adaptation de domaine Ben-David et al. [2010b], une borne de généralisation doit, pour être utile :

— dépendre de quantités que l'on peut estimer à partir d'un nombre fini d'échantillons (cette condition n'est pas spécifique à l'adaptation de domaine)

— dépendre le moins possible de la loi d'étiquetage du domaine cible $Q_{Y|X}$ (en particulier si on cherche à dériver un algorithme non supervisé (UDA) de cette borne)

Pour rendre notre borne compatible avec de telles exigences, nous allons utiliser les inégalités triangulaires.

Définition 12. *Soit* $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ *une fonction de coût. Nous noterons que*

— $\ell \in TI$ *ssi* $\forall y_a, y_b, y_c \in \mathcal{Y}, \ell(y_a; y_c) \leq \ell(y_a; y_b) + \ell(y_b; y_c)$*.*

— $\ell \in TI'$ *ssi* $\forall y_a, y_b, y_c \in \mathcal{Y}, \ell(y_a; y_c) \leq \ell(y_a; y_b) + \ell(y_c; y_b)$*.*

Remarque 4.8. *Ces deux inégalités triangulaires sont valides pour la fonction de coût de marge (margin loss) de Zhang et al. [2019] :*

$$\ell(\hat{y}; y) = \left[1 - \frac{1}{\rho}[\delta(\hat{y}; y)]_+\right]_+ = \begin{cases} 1 & \text{si } \operatorname{argmax}(\hat{y}) \neq \operatorname{argmax}(y) \\ \max(0, 1 - \frac{\delta(\hat{y};y)}{\rho}) & \text{sinon} \end{cases} \tag{4.31}$$

où $[x]_+ := \max(0, x)$ *et* $\delta(\hat{y}; y) = \frac{1}{2}\min_{k \neq k_y} \hat{y}_{k_y} - \hat{y}_k$ *est une marge multi-classe entre le score* $\hat{y}_{k_y}$ *attribué à la plus grande composante de* y *(*$k_y = \operatorname{argmax}_k y_k$*) et le second meilleur score* $\max_{k \neq k_y} \hat{y}_k$*.*

Le lemme suivant fait intervenir des raisonnements assez habituels lorsque l'on cherche à transformer une borne comme celle de l'Equation (4.27) en une borne utile en pratique (au sens des remarques faites plus haut).

Lemme 4.2. *Soient* $h, h' \in \mathcal{H}$ *deux hypothèses,* ℓ *une fonction de coût et* μ *une mesure positive*[6] *sur les paires* (x, y)*. Alors, en notant* μ_X *la loi marginale de* μ *par-rapport à* x*,*

— *si* $\ell \in TI$

$$R_\mu^\ell(h) \leq R_\mu^\ell(h') + R_{\mu_X}^\ell(h; h') \tag{4.32}$$

où $R_{\mu_X}^\ell(h; h') = \int \ell(h(x); h'(x)) d\mu_X$*.*

— *De la même façon, si* $\ell \in TI'$

$$R_{\mu_X}^\ell(h; h') \leq R_\mu^\ell(h) + R_\mu^\ell(h') \tag{4.33}$$

6. μ pourrait être P ou Q ou même λP.

Démonstration. En effet, si $\ell \in TI$

$$\begin{aligned} R^{\ell}_{\mu}(h) =& \int \ell(h(x); y) d\mu(x, y) \leq \int \ell(h(x); h'(x)) + \ell(h'(x); y) d\mu(x, y) \\ =& R^{\ell}_{\mu}(h') + R^{\ell}_{\mu_X}(h; h') \end{aligned} \tag{4.34}$$

de la même façon, si $\ell \in TI'$

$$\begin{aligned} R^{\ell}_{\mu_X}(h; h') =& \int \ell(h(x); h'(x)) d\mu_X(x) = \int \ell(h(x); h'(x)) d\mu(x, y) \\ \leq& \int \ell(h(x); y) + \ell(h'(x); y) d\mu(x, y) \\ =& R^{\ell}_{\mu}(h) + R^{\ell}_{\mu}(h') \end{aligned} \tag{4.35}$$

□

En suivant Acuna et al. [2021], et en prenant en compte la correction que nous avons pu apporter à leur borne basée sur les ϕ divergences, nous proposons la notion de divergence suivante.

Définition 13 ($D^{\phi,\ell}_{h,\mathcal{H}}(P_X, Q_X)$)**.** *Soient P_X, Q_X deux distributions sur x et $\lambda > 0$. Soient $\phi \in \Phi_1$, et ℓ une fonction de coût. Soit $\mathcal{H}$ une classe d'hypothèses et $h \in \mathcal{H}$. Alors, nous définissons,*

$$D^{\phi,\ell}_{h,\mathcal{H}}(Q_X \| \lambda P_X) := \sup_{h' \in \mathcal{H}} R^{\ell}_{Q_X}(h; h') - R^{\phi^* \circ \ell}_{\lambda P_X}(h; h') \tag{4.36}$$

où par extension $R^{\phi^* \circ \ell}_{\lambda P_X}(h; h') := \int \phi^*(\ell(h; h')) d\lambda P_X = \lambda R^{\phi^* \circ \ell}_{P_X}(h; h')$.

Corollaire 4.1. *Soit $\phi \in \Phi_1$. Soient également $\ell \in TI$ et $\ell_\phi \in TI'$ telles que $\phi^* \circ \ell \leq \ell_\phi$. Étant donné une hypothèse $h \in \mathcal{H}$, nous avons la borne de généralisation suivante :*

$$R^{\ell}_{Q}(h) \leq \inf_{\lambda > 0} R^{\ell_\phi}_{\lambda P}(h) + D^{\phi,\ell}_{h,\mathcal{H}}(Q_X \| \lambda P_X) + \gamma^*_\lambda \tag{4.37}$$

*où $\gamma^*_\lambda := \inf_{h' \in H} R^{\ell}_{Q}(h') + R^{\ell_\phi}_{\lambda P}(h')$ quantifie l'*adaptabilité *de la tâche entre les domaines P et Q.*

Démonstration. À partir du lemme 4.2 et de la Définition 13, pour tout $h' \in \mathcal{H}$

$$\begin{aligned} R^{\ell}_{Q}(h) \leq& R^{\ell}_{Q}(h') + \underbrace{R^{\ell}_{Q_X}(h; h')}_{\leq R^{\phi^* \circ \ell}_{\lambda P_X}(h; h') + D^{\phi,\ell}_{h,\mathcal{H}}(Q_X \| \lambda P_X)} \\ \leq& R^{\ell}_{Q}(h') + \underbrace{R^{\phi^* \circ \ell}_{\lambda P_X}(h; h')}_{\leq R^{\ell_\phi}_{\lambda P_X}(h; h') \leq R^{\ell_\phi}_{\lambda P}(h) + R^{\ell_\phi}_{\lambda P}(h')} + D^{\phi,\ell}_{h,\mathcal{H}}(Q_X \| \lambda P_X) \\ \leq& R^{\ell_\phi}_{P}(h) + D^{\phi,\ell}_{h,\mathcal{H}}(Q_X \| \lambda P_X) + (R^{\ell}_{Q}(h') + R^{\ell_\phi}_{\lambda P}(h')) \end{aligned}$$

Étant donné que h' a été à l'origine choisie arbitrairement dans $\mathcal{H}$, le dernier terme peut être remplacé par son infimum sur $h' \in \mathcal{H}$, donnant γ^*_λ. □

Remarque 4.9. *Le lecteur attentif pourrait se demander légitimement pourquoi nous introduisons une fonction de coût alternative ℓ_ϕ plutôt qu'utiliser $\phi^* \circ \ell$ qui semble être le choix naturel. Cela est fait pour apporter plus de flexibilité sur le choix de ℓ et ϕ dans la mesure où, lorsque $\phi \in \Phi_1$ et $\ell \in TI$, alors en général $\phi^* \circ \ell \notin TI'$. C'est par exemple le cas pour la divergence KL et la fonction de coût de marge définie dans l'Equation (4.31). À l'inverse, puisque dans ce cas $\ell(\hat{y}; y) \in [0, 1]$, alors on peut remarquer que $\phi^* \circ \ell \leq \ell_\phi := \phi^*(1)\ell$, ce qui respecte les hypothèses du théorème.*

Remarques et considérations pratiques

Le résultat présenté dans le Corollaire 4.1 laisse une certaine liberté pour le choix de l'ensemble ϕ, ℓ et ℓ_ϕ. Cependant, ϕ, ℓ, ℓ_ϕ devraient respecter les conditions suivantes.

(P1) $\ell \in TI$, $\ell_\phi \in TI'$ et $\phi^* \circ \ell \leq \ell_\phi$ (pour assurer la validité de la borne)

(P2) ℓ doit être calibrée ou « bayes consistent » (voir Steinwart [2007], Tewari and Bartlett [2007])

(P3) ℓ_ϕ est convexe et bornée inférieurement (une propriété souhaitable pour l'optimisation)

Par exemple, en suivant les étapes de Zhang et al. [2019] et en choisissant pour ℓ la fonction de coût de marge définie Eq (4.31), en choisissant par exemple $\rho = 1$: puisque range$(\ell) = [0, 1]$, en utilisant $\ell_\phi = \phi^*(1)\ell$, alors (P1) est valide. ℓ est une borne supérieure de la loss $0-1$[7], et on est toujours assurés que contrôler le ℓ-risque assure un risque $0-1$ faible. Finalement, ℓ_ϕ est bornée inférieurement, mais pas convexe. Ce n'est pas un problème dans la mesure où il est toujours possible de la remplacer par une fonction de coût convexe plus grande, telle que la fonction de coût de marge.

4.3 Lien entre théorie et pratique

Nos bornes présentent certains avantages théoriques par-rapport aux travaux antérieurs. Pourtant, dans cette section, nous allons montrer que dans le cas d'un problème d'adaptation de domaine *réaliste* de vision par ordinateur, il n'y a généralement pas de raison de penser qu'on puisse en tirer un bénéfice significatif. De façon plus générale, nous allons mettre en évidence deux hypothèses souvent invoquées dans la littérature de l'adaptation de domaine lorsque l'on cherche à justifier l'efficacité de tel ou tel algorithme pour optimiser l'une de ces bornes, et montrer que ces hypothèses sont en pratique infondées dans la plupart des cas, ce qui laisse peu de garanties théoriques aux algorithmes présentés dans la littérature.

Nous allons caractériser le lien qui existe entre la théorie vue précedemment dans ce chapitre et les algorithmes d'UDA vus au chapitre 3. Soient deux domaines $\mathcal{S}$ et $\mathcal{T}$ définis sur $X \times Y$. Étant donnés des échantillons annotés de $\mathcal{S}$ et des échantillons non-annotés de $\mathcal{T}$, on cherche à minimiser le risque dans le domaine cible $R_\mathcal{T}$. La plupart des algorithmes d'adaptation de domaine appliqués à l'apprentissage profond exploitent l'hypothèse du covariate shift. Certains de ces algorithmes sont basés sur les bornes supérieures majorant le risque dans le domaine cible. C'est par exemple le cas de DANN, dont les auteurs justifient le fonctionnement en invoquant la borne de Ben-David (Eq. 4.2).

La plupart des bornes vues dans la littérature épousent la forme suivante (voir par exemple Ben-David et al. [2010b], Mansour et al. [2009], Shen et al. [2017] ou Redko et al. [2020] pour une liste exhaustive des bornes).

$$R_\mathcal{T}(h) \leq R_\mathcal{S}(h) + \delta(\mathcal{S}_X, \mathcal{T}_X) + \gamma \tag{4.38}$$

Le deuxième terme de la partie droite de l'équation, $\delta(\mathcal{S}_X, \mathcal{T}_X)$, représente une divergence entre les distributions marginales sur X des deux domaines. De plus, cette divergence dépend d'une classe d'hypothèses $\mathcal{H}$. Le dernier terme de l'équation (4.38), γ, quantifie la notion d'adaptabilité, c'est-à-dire l'existence d'une unique hypothèse $h^\star \in \mathcal{H}$ avec une bonne performance sur les deux domaines à la fois. Ce terme ne peut pas être estimé en l'absence d'annotations dans le domaine cible, mais on s'attend à ce qu'il soit à-priori faible si on fait l'hypothèse du covariate shift et si $\mathcal{H}$ contient des fonctions suffisamment expressives.

Le premier terme $R_\mathcal{S}(h)$ peut-être facilement minimisé vers zéro, à un terme de généralisation près, en optimisant h sur un objectif de classification (par exemple l'entropie croisée) puisque le domaine source est supposé complètement annoté. Cependant, pour un espace d'hypothèses donné, $\delta(\mathcal{S}_X, \mathcal{T}_X)$ ne peut pas être minimisé, puisque les distributions d'entrée $\mathcal{S}_X$ et $\mathcal{T}_X$ sont fixées. De

7. Plus précisément $\ell(\hat{y}; y) \geq \mathbf{1}_{\text{argmax}(\hat{y}_k) \neq \text{argmax}(y_k)}$

plus, dans la majorité des transferts considérés dans la littérature de vision par ordinateur, les domaines source et cible ont justement des supports disjoints, i.e. $\text{supp}(\mathcal{S}_X) \neq \text{supp}(\mathcal{T}_X)$. Par exemple, les images d'un objet pris en plein jour (le domaine source) ont une probabilité nulle sur la distribution des images prises de nuit (le domaine cible), et vice-versa. Un autre exemple est la paire de domaines MNIST / SVHN. Ces deux domaines sont des datasets de classification de chiffres connus : alors que MNIST contient des chiffres manuscrits blancs sur fond noir, avec une variété très faible de styles et de taille, SVHN, plus riche, est un jeu de données de numéros de maisons. Il offre des images avec une grande variété de couleurs, de formes et de tailles de chiffre. Dans ce cas également, il est donc très facile de déterminer avec une certitude totale duquel de ces deux jeux de données une image aurait été tirée. Dans de tels cas, la notion de distribution conditionnelle, par exemple $\mathcal{S}_{Y|X}$, n'est pas définie en-dehors du support de $\mathcal{S}$. De ce fait, la contrainte $\mathcal{S}_{Y|X} = \mathcal{T}_{Y|X}$ n'a plus d'influence, et **la notion de covariate shift n'a plus du tout de sens**. Il s'agit là de la première erreur courante faite dans la littérature que nous souhaitions montrer.

Remarque 4.10. *Notons qu'on peut toujours donner un sens qualitatif à la notion de covariate shift : les domaines source et cible contiennent les mêmes classes, annotées de la même façon. Seulement, quand les supports sont disjoints, il n'est plus possible de formaliser cette hypothèse avec des probabilités conditionnelles.*

Même si cet état de fait est souvent négligé dans les travaux théoriques de l'adaptation de domaine faisant l'hypothèse du covariate shift, la pratique usuelle consiste à introduire un extracteur de caractéristiques $\Psi : X \to Z$, qui réalise un plongement des distributions $\mathcal{S}_X, \mathcal{T}_X$ vers un espace latent Z où les supports peuvent se superposer (e.g. [Sun and Saenko, 2016, Long et al., 2015, Ganin et al., 2016, Häusser et al., 2017, Zhang et al., 2019]). Lorsque la borne est appliquée sur un espace de descripteurs, $\delta(\mathcal{S}_X, \mathcal{T}_X)$ peut être trivialement minimisée : pour amener le terme de divergence près de zéro, il suffit que Ψ produise une représentation latente qui soit invariante au domaine ; en effet, une représentation est invariante au domaine si et seulement si les caractéristiques issues du domaine source suivent la même distribution que les caractéristiques issues du domaine cible. La plupart des méthodes entraînent donc l'encodeur à satisfaire un double objectif : i) donner des descripteurs suffisamment représentatifs pour obtenir une bonne performance de classification sur le domaine source et ii) minimiser une mesure de divergence entre les distributions latentes source et cible. la méthode d'alignement la plus prépondérante est DANN, qui utilise l'entraînement adversaire : il oppose l'extracteur de caractéristiques à un classifieur de domaines dans un jeu minimax ; le classifieur de domaine s'entraîne à reconnaître le domaine depuis lequel les features ont été encodées en minimisant une erreur de classification binaire. Ensuite, l'extracteur de caractéristiques s'entraîne pour maximiser cette erreur afin de « tromper » le classifieur et rapprocher les distributions l'une de l'autre. Après quelques étapes d'optimisation alternée, l'algorithme atteint un point d'équilibre et les distributions latentes sont alignées. Quand cela arrive, la divergence $\mathcal{H}\Delta\mathcal{H}$ proposée par Ben-David, de laquelle DANN tire son inspiration, est proche de zéro. Puisque le dernier terme γ ne peut pas être évalué sans connaître les labels du domaine cible, seuls les deux premiers termes de la borne sont considérés dans la minimisation. Cependant, quand on applique la borne dans un espace latent, on ne peut plus considérer à-priori que γ est faible. Cette deuxième observation a déjà été relevée par au moins trois papiers récents : respectivement [Zhao et al., 2019, Johansson et al., 2019, Bouvier et al., 2020] sous la forme d'un théorème de type *no-free-lunch*. La conséquence principale de cette observation est que **alors que les deux premiers termes peuvent être minimisés, le troisième peut devenir hors de contrôle**. Plus spécifiquement, l'extracteur de caractéristiques peut projeter différentes régions de $\mathcal{T}_X$ appartenant à différentes classes vers la même région de de Z, ce qui augmente le risque de Bayes dans le domaine cible puisque la fonction de labeling définie sur Z n'est plus considérée comme déterministe. Cette confusion des labels peut être encore aggravée en minimisant la divergence entre les distributions latentes dans les scénarios où les classes n'ont pas le même équilibrage dans le domaine source et le domaine cible : ce cas de figure, appelé *prior shift*, n'est pas incompatible avec l'hypothèse du covariate shift. Pire encore, rien n'empêche une région de $\mathcal{T}_X$ appartenant à une seule classe d'être encodée vers des régions appartenant à

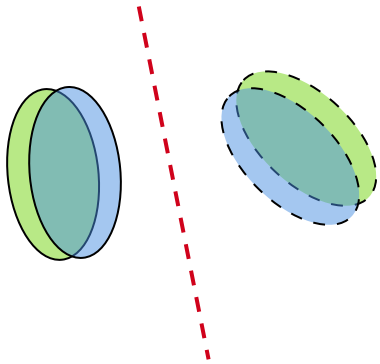

FIGURE 4.5 – Alignement des domaines bien conditionné : la divergence entre la distribution marginale des desccripteurs et γ sont faibles. Les couleurs indiquent le domaine. Les contours continus indiquent la classe 1, les contours en pointillés la classe 2.

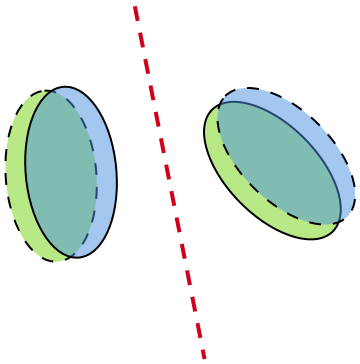

FIGURE 4.6 – Cas dégénéré d'alignement des domaines : la divergence entre les distributions marginales des descripteurs est faible, mais l'erreur jointe γ est élevée. Les couleurs indiquent le domaine. Les contours continus indiquent la classe 1, les contours en pointillés la classe 2.

différentes classes de $\mathcal{S}_X$. Nous illustrons ce cas dans la Figure 4.6. Ces deux phénomènes sont la conséquence de la non-inversibilité de Ψ [Johansson et al., 2019], qui est pourtant indispensable pour que les supports de $\mathcal{S}_Z$ et $\mathcal{T}_Z$ se superposent.

Malgré les problèmes susmentionnés, les méthodes d'alignement de domaine donnent parfois souvent de bons résultats en pratique. Cela peut-être expliqué d'une part par la manière dont les résultats publiés ont pu être sélectionnés. Par exemple, alors que le transfert de SVHN vers MNIST est souvent montré, ce n'est quasiment jamais le cas du transfert inverse (MNIST vers SVHN – voir Figure 4.7 pour une comparaison visuelle des représentations latentes dans chacun des deux cas). En effet, le transfert de SVHN vers MNIST devrait être considéré comme étant le sens « facile » : puisque SVHN est plus riche que MNIST en termes de degrés de liberté (polices, échelles, couleurs, textures de fond), les descripteurs appris grâce à la supervision sur SVHN sont déjà suffisamment robustes pour généraliser quelque peu sur MNIST, et grâce à l'ajustement apporté par l'alignement de domaine, on peut observer dans la partie gauche de la Figure 4.7 que non seulement les distributions marginales de descripteurs sont alignées, mais qu'il en est également de même pour les distributions jointes descripteur-label ; sur chaque cluster correspondant à une classe source se trouve un cluster du domaine cible constitué d'échantillons de la même classe.

Le cas MNIST vers SVHN est bien plus difficile : la supervision sur MNIST ne produit pas de descripteurs qui généralisent naturellement vers SVHN, et aligner les domaines fait converger le modèle vers un cas dégénéré dans lequel les distributions marginales sont alignées mais pas

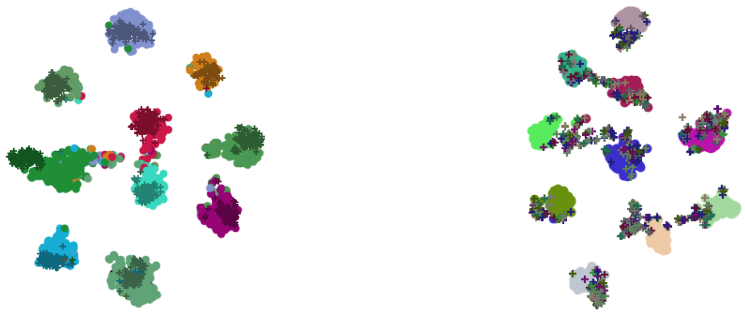

FIGURE 4.7 – Plot t-SNE de l'espace des descripteurs après l'adaptation avec DANN (les labels sont représentés avec les couleurs). Gauche : SVHN→MNIST, Droite : MNIST→ SVHN. SVHN a beaucoup plus de degrés de liberté (digit fonts, scales, colors) que MNIST, il n'est donc pas surprenant que les descripteurs appris par simple supervision sur SVHN transfèrent vers MNIST de manière robuste, ce qui a pour conséquence de maintenir les distributions jointes descripteur/label relatives à chaque domaine proches. Le transfert inverse MNIST → SVHN est bien plus difficile : Les descripteurs appris à partir de la supevision sur MNIST n'extrapolent pas de manière fiable et consistante vers SVHN. Dans ce cas, appliquer l'algorithme DANN mène à un alignement dégénéré.

les distributions jointes descripteur-label. Dans la partie droite de la Figure 4.7, on observe que chaque cluster du domaine source correspondant à une classe est superposé à un cluster constitué d'échantillons du domaine cible, mais provenant de classes arbitraires. Cela indique que l'encodeur a pu utiliser sa capacité importante de réseau de neurones pour exploiter des sources d'information arbitraires issues du domaine cible dans l'espace des images pour reproduire une distribution latente similaire à la distribution latente du domaine source. Les quelques travaux réalisant le transfert MNIST→SVHN utilisent systématiquement des astuces ad-hoc : par exemple augmenter MNIST avec des couleurs aléatoires [Häusser et al., 2017] (ce qui le rapproche qualitativement de SVHN). On peut également mentionner l'utilisation d'une couche de normalisation d'instance (InstanceNorm) [Kumar et al., 2018], qui aide le modèle d'encodage à ignorer les couleurs et les contrastes, ce qui introduit une bonne invariance faite « à la main » qui aide l'alignement conditionnel. En réalité, ces astuces sont constituantes du *biais inductif*, auquel nous consacrerons le Chapitre 5.

4.4 Conclusion

Après une brève présentation des principaux résultats théoriques existants, nous avons dans ce chapitre proposé deux nouvelles bornes pour l'adaptation de domaine. La première est basée sur une équivalence mathématique entre les courbes PR et les courbes de Lorenz : nous avons montré qu'elle est compétitive parmi les bornes ne dépendant pas d'une classe d'hypothèses h. La seconde borne incorpore la notion de Φ-divergence. Elle est relativement flexible, puisqu'il est possible de choisir une Φ-divergence en fonction de la façon dont on souhaite pénaliser les différents écarts d'amplitude entre P et Q.

Cependant, nous avons également rappelé que nos bornes, ainsi que celles présentées dans la littérature, ne sont pas suffisantes pour gérer les problèmes réalistes de vision par ordinateur, car dans ce cas il n'est pas possible de les minimiser entièrement. Par conséquent, les algorithmes qui ont pu être dérivés de ces bornes ne prennent pas en compte l'impact potentiellement néfaste de l'encodeur. Ces éclaircissements apportés par la théorie permettent en outre d'expliquer pourquoi les méthodes d'alignement que nous avons évalué dans le chapitre 3 pouvaient échouer dans certains cas. Enfin, ils montrent que le caractère *no-free-lunch* de ces bornes est un verrou scientifique important.

Chapitre 5

Adaptation de domaine et biais inductif

Sommaire

Nous avons pu démontrer dans le chapitre 4 que les bornes théoriques proposées pour d'adaptation de domaine ne permettaient pas d'expliquer entièrement le succès des méthodes d'adaptation de domaine telles que DANN. Dans ce chapitre, nous défendons l'hypothèse que le bon fonctionnement des méthodes d'adaptation de domaine est en réalité fortement conditionné par l'existence d'un biais inductif résultant d'un certain nombre de choix d'implémentation quasi systématiques dans la littérature.

On appelle « biais inductif » l'ensemble des hypothèses (implicites ou explicites), qui, additionnées aux données d'entraînement, conditionnent le comportement d'un modèle sur les échantillons de test. On peut citer comme exemple l'utilisation de modèles pré-entraînés comme point de départ, un choix de design omniprésent dans la littérature [Ganin et al., 2016, Kang et al., 2019, Tzeng et al., 2017, Raab et al., 2020, Lv et al., 2021], en particulier pour traiter les transferts complexes tels que Office-31 [Gong et al., 2012] et PACS [Li et al., 2017]. Pourtant, dans les publications liées à l'adaptation de domaine, ces choix expérimentaux sont souvent relégués en arrière-plan et présentés comme des détails d'implémentation [Häusser et al., 2017, Ganin et al., 2016, Kang et al., 2019, Tzeng et al., 2017, Raab et al., 2020, Lv et al., 2021]. Leur impact sur le succès du transfert est rarement mentionné et presque jamais comparé à l'impact de la méthode d'adaptation de domaine en elle-même.

Ce manque de clarté introduit de nombreuses incertitudes lorsqu'il s'agit de reproduire les résultats ou de comparer les méthodes d'adaptation de domaine entre elles. Nous proposons donc dans cette partie un ensemble d'expériences illustrant l'importance du biais inductif. Notons que cette notion de biais inductif a déjà été mentionnée dans le cadre de l'adaptation de domaine par Bouvier et al. [2020] dans le cadre du prior shift : nous étendons ici l'étude du biais inductif à une grande variété de dérives de domaines.

5.1 Hypothèses implicites de l'adaptation de domaine en vision artificielle

S'il existe des mécanismes de biais inductif bénéfiques à la transférabilité, c'est qu'il existe des propriétés implicites partagées par tous les problèmes d'adaptation de domaine, en particulier dans le cas de la vision par ordinateur. Nous avons, jusqu'à maintenant, analysé le problème de l'adaptation de domaine sous la contrainte d'hypothèses très peu restrictives sur les transferts considérés : à savoir l'existence d'une unique hypothèse $h^\star \in \mathcal{H}$ ayant une bonne performance de classification sur les deux domaines. Faisons d'abord remarquer que dans le cadre de la vision par ordinateur, nous pouvons définir de nombreux problèmes de transfert respectant cette hypothèse, mais que l'on considérerait comme ambigus voire vides de sens :

Considérons par exemple le transfert d'un problème de classification binaire sur des images. Le domaine source comporte des images de fleurs et de personnes, labellisées 0 et 1 respectivement. Le domaine cible comporte des images de chats et de lapins, labellisées 0 et 1 respectivement. Du point de vue de l'être humain, ce problème serait qualifié de « mal défini », puisqu'on s'attend à avoir les mêmes classes, labellisées de la même manière dans le domaine source et le domaine cible. Pourtant, comme les supports de X_S et X_T sont disjoints, il existe un $h^\star$ performant sur les deux domaines, en supposant $\mathcal{H}$ suffisamment riche. La théorie existante n'est donc pas suffisante pour définir, identifier et écarter de tels cas aberrants.

Dans le chapitre 4, nous avons montré l'aspect no-free-lunch des bornes basées sur cette hypothèse très générale, qui limite fortement leur intérêt pratique. Dans cette section, nous tentons donc de décrire les spécificités des écarts de domaines caractérisant les transferts de vision par ordinateur « bien définis ». Ces spécificités pouvant aider à l'élaboration d'algorithmes plus spécifiques, adaptés aux invariances des problèmes de vision par ordinateur.

Dans le cas des problèmes de classification d'objets en vision par ordinateur, les domaines $\mathcal{D}$ ne sont pas des distributions quelconques sur $X \times Y$. Ces distributions jointes obéissent à un certain nombre d'à priori implicites. Par exemple, les images de $\mathcal{D}$ doivent comporter au maximum un

seul objet, ou du moins un objet bien mis en évidence et présenté comme l'objet à classer. Ne pas respecter cet à priori introduit de la confusion sur la tâche à accomplir, notamment dans le domaine cible où les annotations ne sont pas fournies. Cela pose un problème dans la mesure où la tâche à accomplir dans le domaine cible doit être naturellement *devinée* par le modèle par l'intermédiaire du biais inductif, étant donné des échantillons source annotés et éventuellement des échantillons cible non annotés.

Dans le cas de la vision par ordinateur, pour une paire de domaines issue d'un transfert de domaine « bien défini », nous pouvons en principe définir une fonction de transformation aléatoire T sur l'espace des images (définissant la conditionnelle $p_T(x' \mid x)$, avec $x, x' \in X$) laissant la classe invariante quel que soit x. Pour que cette fonction caractérise entièrement la dérive de domaine considérée, nous imposons également $\text{supp}(\mathcal{T}_X) \subset \text{supp}(T(\mathcal{S}_X))$. Il n'est pas possible de décrire analytiquement une telle fonction dans le cas général, mais nous pouvons l'approcher qualitativement par une combinaison de transformations typiques sur les images.

Nous distinguons donc deux types d'à priori qui restreignent l'ensemble des transferts possibles. Le premier type restreint les domaines individuellement, tandis que le second restreint les paires de domaines que l'on peut associer pour définir un transfert « sensé ».

Nous décrivons maintenant un certain nombre de ces transformations qui pourraient poser un problème aux modèles d'apprentissage profond. **Transformation locale de l'image :** Une transformation « locale » est une transformation qui ne fait interagir chaque pixel qu'avec ses voisins proches. Par exemple, les fonctions de flou gaussien (avec un noyau de faible variance) et de passage en noir et blanc sont des transformations locales. Les transformations locales laissent par-ailleurs inchangé le contenu global de l'image (entre autres la scène, la forme des objets et donc la classe de l'image). Ces transformations sont à priori assez faciles à prendre en compte en adaptation de domaine, puisque leur impact est limité à de faibles champs réceptifs. On peut par exemple utiliser la méthode PixelDA [Bousmalis et al., 2017] décrite Chapitre 2, avec un réseau de traduction d'image à champ réceptif local, pour traduire une image source en image cible en étant assuré que la classe de l'image ne sera pas affectée. **Transformation géométrique :** Une transformation géométrique résulte d'un changement de point de vue de l'objet à classer. Il peut s'agir d'une translation, d'une rotation, d'un changement d'échelle, voire d'une combinaison de tous ces facteurs. Les transformations géométriques laissent la classe de l'image invariante, à l'exception près des objets présentant certaines symétries (par exemple un '6' devient un '9' lorsque tourné de plus de 90°. La plupart des modèles d'apprentissage profond existants ne modélisent pas naturellement l'invariance-équivariance associée à ces transformations. Par exemple, dans le cas du réseau à convolutions, l'image d'un objet et cette même image tournée de 90° n'activeront pas les mêmes filtres. Pour généraliser, voire extrapoler efficacement vers le domaine cible, il serait préférable d'avoir une représentation *équivariante*, c'est-à-dire une représentation dans laquelle l'information relative à la nature de l'objet est décorrélée de l'information relative à sa pose. Des travaux ont déjà montré le mauvais comportement des architectures habituelles lorsqu'on extrapole les transformations géométriques par rapport aux données d'entraînement [Schott et al., 2022]. Quelques solutions ont été proposées, mais aucune ne passe à l'échelle dans le cas général [Cohen and Welling, 2016, Sangalli et al., 2021]. Appliquer des techniques d'augmentation sur les images d'entraînement peut se révéler très utile pour couvrir une plus grande diversité de poses, mais les augmentations géométriques synthétisables restent limitées au repère 2D au sein duquel l'image a été projetée : difficile de synthétiser un changement de point de vue en 3D par exemple.

Changement de style non-local : Malgré tous les efforts que l'on peut fournir pour caractériser mathématiquement les transformations visuelles préservant la classe, certains écarts de domaine demeurent relativement difficiles à décrire. Prenons pour exemple la différence entre le dessin enfantin et approximatif d'un avion et une photographie de ce même avion : l'avion dessiné a des proportions très exagérées, si bien qu'on ne le reconnaît que comme un assemblage de ses parties (ailes, fuselage, etc).

Changement de contexte/fond : Dans le domaine cible, l'objet d'intérêt peut avoir un aspect similaire à celui observé en source, mais être présenté avec un fond, un contexte différents

du domaine source. Il y a un risque que ce changement de contexte agisse comme une source d'information parasite, qui, si elle n'est pas filtrée ou démêlée de l'information relative à l'objet d'intérêt, introduise de la confusion lors du processus de classification, similairement à ce qui se produit dans nos expériences avec MNIST Algebra (voir chap. 3).

Dans la partie qui suit, on identifie les choix expérimentaux qui, en pratique, permettent aux modèles de vision d'être robustes à ces transformations par l'intermédiaire du biais inductif.

5.2 Biais inductif existant dans la pratique

On classe les cas de biais inductif en quatre catégories : i) les biais relatifs au mécanisme d'alignement des domaines, ii) les biais résultant de l'augmentation des données, iii) les biais résultant de l'initialisation et du fine-tuning de l'extracteur et iv) les biais résultant de son architecture.

5.2.1 Biais inductif et alignement de domaine

La première expérience est conçue pour évaluer le comportement de quelques méthodes d'alignement de domaine en présence d'un cas de prior shift. On a pu démontrer qu'aligner les distributions des descripteurs source et cible n'est pas une condition suffisante pour aligner les domaines ; montrons maintenant qu'elle n'est pas non plus nécessaire. Et pire, qu'aligner les descripteurs peut avoir un impact négatif par-rapport à la référence source-only. On construit deux versions du jeu de données MNIST, faisant office de domaine source et de domaine cible respectivement. Dans le domaine source, les 10 classes sont représentées de façon équilibrée, c'est-à-dire 10% d'images de chaque classe. Dans le domaine cible, en revanche, la représentation des classes est déséquilibrée, avec des partitions allant de 5% à 20% du total des images. Notons que ce transfert vérifie strictement l'hypothèse du covariate shift, à savoir $p_s(y \mid x) = p_t(y \mid x)$. On nomme ce jeu de données MNIST-Imbalanced (MI).

Dans cette expérience, on évalue trois méthodes d'adaptation de domaine. La première consiste à entraîner $g \circ \Psi$ sans alignement (référence source-only, SO). La seconde et la troisième approche sont respectivement les algorithmes DANN [Ganin et al., 2016] et Associative-DA [Häusser et al., 2017]. Associative-DA est une méthode d'alignement qui n'aligne que partiellement les distributions source et cible dans l'espace des descripteurs. En effet, si la distribution marginale des classes n'est pas la même dans le domaine source et le domaine cible, alors aligner parfaitement les descripteurs source et cible, avec une méthode adversaire comme DANN par exemple, risque de dégrader les performances : dans ce cas précis, pour satisfaire parfaitement un critère de divergence entre les distributions, des échantillons appartenant à une classe (et donc un mode) seraient déplacés vers un autre mode pour rétablir l'équilibre des classes observé en source, et seraient par conséquent mal classés. Pour éviter cela, Associative-DA se base sur l'hypothèse que, au sein de la distribution source comme de la distribution cible, les descripteurs d'une même classe vont naturellement former des clusters. Elle utilise cette hypothèse pour aligner les *centroïdes* des clusters sans aligner leur population. On s'attend donc à ce qu'Associative-DA soit plus robuste au prior shift. La table 5.1 illustre les résultats de cette expérience. Elle confirme que dans le cas du prior shift, aligner les distributions marginales dans l'espace des représentations déteriore les performances. Dans le cas source-only, la performance en cible est sans surprise proche de la perfection. Dans le cas de DANN, la performance est largement déteriorée. Enfin, dans le cas d'Associative-DA, l'alignement partiel tolérant au prior shift restaure une performance inférieure, mais proche de source-only. En définitive, si cette forme d'alignement dégrade légèrement les performances par-rapport à source-only, elle montre une instance de biais inductif qui rend l'alignement plus robuste au prior shift.

	SO	DANN	ASSO-DA
M→ MI	0.99	0.67	0.97

TABLE 5.1 – Effets de l'alignement dans un transfert comprenant un prior-shift ; M = MNIST équilibré, MI = MNIST avec déséquilibre de classe

5.2.2 Augmentation des données

Un moyen connu et efficace d'améliorer la capacité de généralisation des descripteurs et d'augmenter explicitement les images d'entraînement avec une famille de transformations préservant la classe. Par exemple, une telle famille peut être construite en composant des symétries, rotations, mises à l'échelle, flous ou distorsion des couleurs aléatoires. Ces transformations synthétisables représentent un sous-ensemble de l'ensemble des transformations préservant la classe, et peuvent donc expliquer en partie les écarts de domaine que l'on trouve dans la réalité. L'augmentation des images d'entraînement est donc une autre instance de biais inductif et son effet sur la transférabilité des descripteurs doit être mesuré dans le cadre de l'adaptation de domaine. Sur la table 5.2, on montre qu'une simple randomisation des couleurs du chiffre et du fond de MNIST, utilisé comme jeu de données source, apporte un gain significatif dans la résolution du transfert MNIST→SVHN, pourtant difficile et hors de portée des modèles pré-entraînés, avec un gain en performance de 45%. Cette expérience montre à quel point une transformation simple, mais ad hoc, peut aider à adapter l'écart de domaine complexe existant entre MNIST et SVHN.

	DANN w/o augment	DANN w/ augment
M→ S	0.15	0.6
P→Sk	0.38	0.42
P→C	0.22	0.24
P→A	0.58	0.64

TABLE 5.2 – Effets de l'augmentation des données ; M=MNIST, S=SVHN, P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart ; Le modèle est pré-entraîné

5.2.3 Biais inductif lié au pré-entraînement et au fine-tuning

La question suivante est probablement la plus subtile. Il a été expliqué dans le chapitre 3 et dans notre publication [Siry et al., 2020a] que la performance en cible d'un modèle entraîné en source-only est généralement annonciatrice du succès d'un algorithme d'alignement. Cette observation peut mener à plusieurs interprétations lorsque considérée sous l'aspect du biais inductif. Effectivement, la plupart des problèmes de classification d'image possèdent des caractéristiques communes : les échantillons contiennent un seul objet à classer. De la même façon, les écarts de domaines observés dans des situations concrètes sont construits à partir de transformations préservant la classe, qui sont un sous-ensemble très restreint de toutes les transformations pouvant être définies sur l'espace des images (se référer à la section précédent pour plus de détails). Un écart de domaine réaliste peut donc être modélisé comme une combinaison aléatoire de ces transformations. Prendre ces à priori en compte permet de définir une famille plus restreinte de domaines et de transferts, pour lesquels il est plus facile de construire des fonctions d'encodage invariantes au domaine. Le pré-entraînement ImageNet est couramment utilisé par la communauté Deep Learning. Il fournit un espace de descripteurs informatif, général et robuste, ainsi qu'une paramétrisation de départ qui accélère considérablement la convergence du modèle vers à peu près n'importe quelle nouvelle tâche d'intérêt. Enfin, le pré-entraînement augmente considérablement les capacités de généralisation du modèle sur cette tâche d'intérêt, en particulier lorsque les données d'entraînement pour celle-ci sont

Transfer	NoPT+FT	PT	PT+FT
M→ MM	0.13	0.19	0.56
MM→ M	0.98	0.29	0.97
S→ M	0.56	0.25	0.84
M→ S	0.06	0.19	0.23
P→ Sk	0.16	0.17	0.41
P→ C	0.16	0.17	0.22
P→ A	0.24	0.34	0.58

TABLE 5.3 – Performances dans le domaine cible du classifieur KNN sur des descripteurs pré-entraînés statiques (PT) et sur ces mêmes descripteurs après fine-tuning (PT+FT). On montre également le cas des descripteurs non-pré-entraînés, mais fine-tunées comme référence (NoPT+FT) ; P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart, M=MNIST, MM=MNIST-M, S=SVHN

en faible nombre. Cette tendance à généraliser finalement s'étend naturellement à l'adaptation de domaine : en pratique, partir d'un modèle pré-entraîné est indispensable pour adapter correctement des transferts low-shot comme ceux du benchmark Office-31, et facilite grandement la résolution de transferts plus simples tels que SVHN→ MNIST. Sur une majorité de transferts, Source-Only, la baseline la plus simple, donne des résultats convaincants quand le pré-entraînement est utilisé. En prenant la même architecture, mais en partant d'une initialisation aléatoire, la performance en cible chuterait fortement. Dans les expériences qui suivront dans ce chapitre, le pré-entraînement sera mené à l'aide du jeu de données ImageNet, extrêmement connu. Il contient 1000 catégories et 1,4 million d'échantillons. Résoudre ce problème de classification de grande échelle encourage la découverte de descripteurs invariants à toute une variété de transformations préservant la classe. Il n'est donc pas surprenant que les modèles obtenus de cette façon puissent plus facilement adapter un domaine vers un autre. Pour caractériser le rôle du pré-entraînement dans le cadre de l'adaptation de domaine, on envisage les deux hypothèses suivantes (la deuxième étant une version plus raffinée de la première) : i) le pré-entraînement mène à un point de fonctionnement où les échantillons source et cible sont regroupés de façon cohérente dans l'espace latent ii) le pré-entraînement, suivi d'un biais d'optimisation de l'entropie croisée sur le domaine source mène à cet espace de descripteurs où les distributions source et cibles sont clusterisées et rapprochées. Afin de vérifier la vraisemblance de chacune de ces hypothèses, on propose de réaliser deux expériences.

Montrer la transférabilité des descripteurs avec les k plus proches voisins : Afin de valider la première hypothèse, on élabore un simple classifieur basé sur les K plus proches voisins (KNN) pour prédire le label des descripteurs produits par le modèle pré-entraîné à partir du domaine cible. Le classifieur fonctionne comme suit : étant donné un descripteur du domaine cible non labellisé, il cherche les K (K=50) plus proches descripteurs annotés du domaine source au sens d'une métrique de base dans cette espace de descripteurs (par exemple la norme L2), il prédit ensuite le label du descripteur cible à partir du vote majoritaire des K labels source. Une bonne performance de ce classifieur montrerait que les représentations des classes dans le domaine cible sont proches des représentations de ces mêmes classes dans le domaine source. Autrement dit, une bonne performance du classifieur KNN signifie que l'espace des descripteurs pré-entraînés est naturellement équivariant à la classe et invariant au domaine. Cependant, comme mentionné plus haut, attendre que cette propriété soit vérifiée immédiatement par les descripteurs figés du pré-entraînement ImageNet est peut-être un peu trop restrictif étant donné que l'entraînement source-only implique également un ajustement de bout en bout de l'ensemble de l'encodeur sur la tâche de classification du domaine source, ce qui peut en conséquence modifier significativement l'espace des descripteurs. Pour tenir compte de cette seconde hypothèse, les effets du fine-tuning source-only ont également été mesurés. De ce fait, on évalue la performance du classifier KNN sur les descripteurs produits par un ResNet-18

pré-entraîné sur ImageNet, avant (PT) et après (PT+FT) le fine tuning Source-Only. Pour avoir une référence, on évalue également la performance du classifieur KNN sur les descripteurs d'un modèle entraîné sur source, mais sans pré-entraînement (NoPT+FT). On reproduit l'expérience sur quelques transferts, et présentons les résultats dans la table 5.3. Sur les transferts de classification de chiffres, on observe une performance faible, mais meilleure que le hasard lorsque le modèle n'est pas fine-tuné sur source, par exemple 19% pour MNIST→ MNIST-M et 25% pour SVHN→ MNIST. Cependant, les distributions de descripteurs source et cible deviennent significativement plus proches à mesure que l'encodeur est fine-tuné sur source, avec des performances KNN montant à 56% et 84% respectivement sur ces deux mêmes transferts. La même tendance peut-être observée sur les transferts de PACS. Les bénéfices du pré-entraînement ne peuvent donc pas être uniquement expliqués par la géométrie des descripteurs figés après pré-entraînement : les couches cachées, pré-entraînées du ResNet-18 conditionnent également les dynamiques du fine-tuning source-only, qui mène la plupart du temps à une bonne représentation pour le domaine source et pour le domaine cible grâce à un biais implicite intervenant lors de l'optimisation.

Jeu de données Dead Pixel CIFAR : Pour compléter cette première expérience portant sur l'influence du pré-entraînement et du fine-tuning sur source, on propose une autre expérience afin de montrer qu'introduire certains facteurs de confusion bien choisis peuvent rendre le pré-entraînement désavantageux. Notons que cette expérience est délibérément extrême, puisqu'elle ne respecte pas les hypothèses implicites décrivant les problèmes de classification de vision par ordinateur énoncés dans la précédente section, au point qu'elle fasse échouer le biais inductif du pré-entraînement ImageNet, habituellement utile pour cette famille de problèmes. Dans cette expérience, on construit un transfert synthétique pensé pour rendre le pré-entraînement moins efficace que l'initialisation aléatoire : dans les deux domaines, on construit une image en piochant un échantillon au hasard dans CIFAR-10, et en remplaçant la couleur du $i^{ème}$ pixel en haut à gauche par un gris uniforme i.e. $RGB = (0.5, 0.5, 0.5)$. L'indice i du pixel est choisi entre 1 et 10. Dans le domaine source, i correspond à la classe d'origine de l'échantillon CIFAR-10. Dans le domaine cible, au contraire, l'indice i du « pixel mort » est choisi indépendamment de l'étiquette d'origine de l'image CIFAR-10, selon la distribution uniforme. Il en résulte que dans le domaine cible, l'indice du pixel et l'objet CIFAR-10 observé sont deux sources d'information complètement décorrélées. Dans les deux domaines, la classe d'un échantillon correspond à l'indice i de son pixel mort. Le but de cette expérience est de mesurer à quel point le classifier exploite le pixel mort comme source information pour la classification. Cette expérience illustre un des obstacles principaux en adaptation de domaine, à savoir la présence de facteurs de confusion présents dans le domaine source, mais absents dans le domaine cible. En cela, elle est réminiscente de ce qu'on a pu faire avec MNIST-Algebra dans le chapitre 3. Lors de la supervision sur le domaine source, on s'attend à ce qu'un réseau pré-entraîné sur ImageNet associe immédiatement l'information de label avec l'information du contenu originel de l'image CIFAR-10, à savoir la nature de l'objet observé sur l'image, et ignore l'information venant du pixel mort au profit de l'information produite par les descripteurs de l'objet observé, puisque le pré-entraînement rend cette dernière plus saillante et filtre les détails de la taille d'un pixel. Dans le domaine cible, où l'objet de CIFAR-10 n'est plus utile pour prédire le label de l'image, le modèle pré-entraîné devrait donc échouer même après un fine-tuning source-only.

Au contraire, un classifieur initialisé aléatoirement devrait rapidement identifier le pixel mort comme le moyen le plus sûr et le plus facile de prédire le label de l'image, et devrait davantage ignorer l'objet de CIFAR-10. De ce fait, ce classifier entraîné de zéro devrait atteindre une bonne performance dans le domaine cible. On baptise ce transfert « dead-pixel CIFAR » et on en présente les résultats dans la table 5.4 : comme attendu, le pré-entraînement atteint une moins bonne performance, mais parvient quand même à exploiter une partie de l'information venant du pixel mort, ce qui mène à une performance raisonnable dans le domaine cible. Cependant, le modèle entraîné à partir de zéro fonctionne parfaitement dans les deux domaines, et atteint la performance maximale dans le domaine cible. De plus, aligner explicitement avec DANN ne semble pas apporter

	sans pré-entraînement	
	SO	DANN
M→ MM	0.17	0.63
MM→ M	0.98	0.98
S→ M	0.65	0.71
M→ S	0.07	0.1
P→ Sk	0.15	0.21
P→ C	0.17	0.26
P→ A	0.23	0.22
D-Pix	0.95	0.99

	avec pré-entraînement	
	SO	DANN
M→ MM	0.6	0.99
MM→ M	0.98	0.98
S→ M	0.83	0.88
M→ S	0.25	0.15
P→ Sk	0.42	0.6
P→ C	0.22	0.67
P→ A	0.58	0.76
D-Pix	0.75	0.76

TABLE 5.4 – Performances dans le domaine cible de la baseline source-only et de l'algorithme d'alignement DANN ; **Gauche** : Pas de pré-entraînement, **Droite** : pré-entraînement ; P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart, M=MNIST, MM=MNIST-M, S=SVHN, D-Pix=transfert Dead Pixel CIFAR

d'amélioration par-rapport à source-only dans le cas non pré-entraîné.

On propose un ensemble d'expériences supplémentaires dans la table 5.4 pour résumer l'influence du pré-entraînement dans des transferts qui cette fois satisfont les propriétés d'un shift de domaine de vision par ordinateur « bien défini ». On constate un gain en performance significatif et systématique lorsque le pré-entraînement est utilisé, que ce soit dans le cas du simple fine-tuning source-only comme dans le cas de l'alignement de domaines avec DANN. Sur la plupart des transferts impliquant des jeux de données de chiffres, DANN apporte un gain de performance supplémentaire, même dans le cas non pré-entraîné (+46% sur MNIST → MNIST-M, +6% sur SVHN→MNIST sans pré-entraînement). Ces transferts tombent toutefois dans la catégorie des transferts « faciles », pour lesquels la supervision en source fait naturellement émerger des descripteurs transférables vers le domaine cible sans pré-entraînement. La seule exception notoire est le difficile MNIST → SVHN, face auquel DANN échoue à la fois dans le régime pré-entraîné et non pré-entraîné. Ce cas dégénéré a déjà été analysé dans le chapitre 4. Sur les transferts impliquant le jeu de données PACS, on observe qu'utiliser DANN est souvent utile dans le cas pré-entraîné, avec un gain significatif en performance sur cible. Cependant, DANN améliore en moyenne légèrement la performance dans le cas non pré-entraîné.

Dans cette section, on a pu confirmer que les représentations pré-entraînées permettent généralement d'obtenir des descripteurs particulièrement bons pour la transférabilité. On a également montré que le pré-entraînement seul n'est pas suffisant pour obtenir une bonne performance en cible, mais qu'il doit être combiné avec du fine-tuning source-only. Enfin, l'expérience synthétique sur dead-pixel CIFAR semble confirmer l'idée que le pré-entraînement doit être considéré comme une instance de biais inductif aidant pour les shifts de domaine habituels en computer vision, mais qui échoue dans les cas où l'introduction de facteurs de confusion contraires au « bon sens » rendent ce biais contre-productif.

5.2.4 biais inductif et architecture du réseau

L'évolution de la nature des descripteurs au cours du fine-tuning source-only est un phénomène complexe, non-linéaire, qui ne dépend pas seulement de l'initialisation de l'encodeur. Pour comprendre plus finement les conditions qui font émerger des descripteurs à la fois transférables et utiles pour la tâche d'intérêt, il faut également prendre en compte l'architecture de l'encodeur. Dans cette section, on propose quelques expériences supplémentaires pour montrer la sensibilité de l'apprentissage par transfert à ces composantes architecturales.

La BatchNorm [Ioffe and Szegedy, 2015] est principalement utilisée pour accélérer l'entraînement des modèles profonds : elle centre et réduit l'amplitude de toutes les activations en sortie d'une couche cachée, afin que ces dernières restent dans la zone sensible des non-linéarités, ce qui évite les points de fonctionnement où la surface de loss est plate, empêchant l'entraînement. Cependant, elle n'affecte pas que l'optimisation des réseaux, mais aussi leur inférence : l'opération multiplicative au sein de la BatchNorm permet de faire émerger plus facilement des descripteurs davantage invariants à certaines transformations, par exemple des modifications de colorisation ou de contraste. Pour illustrer cette propension à ce genre d'invariance, on propose un transfert très simple, dont le but est d'adapter MNIST vers sa contrepartie dans laquelle les couleurs sont inversées (noir sur blanc au lieu de blanc sur noir), que l'on appellera Inv-MNIST. On montre dans la table 5.5 que dans le cas d'une architecture non pré-entraînée, la présence d'une BatchNorm conditionne entièrement la performance dans le domaine cible, que l'on utilise l'algorithme source-only ou bien DANN.

M→ InvM	SO	DANN
Encoder-BN	0.5	0.95
Encoder-NoBN	0.05	0.04

TABLE 5.5 – Performance Source-only pour deux architectures d'encodeur différentes (avec et sans batch normalization) ; M=MNIST, InvM=MNIST avec couleurs inversées

Le pooling global est une opération qui réduit une feature map de taille $C \times H \times W$ en un simple tenseur de taille $C \times 1 \times 1$. Le choix le plus courant est de faire la moyenne canal par canal sur l'ensemble des dimensions spatiales. L'opérateur « moyenne » ne retient pas la position spatiale d'origine des descripteurs et possède de ce fait la propriété d'invariance par translation. La plupart des backbones populaires dans la littérature tels que VGG-16, ResNet-18 ou DenseNet incorporent une fonction de pooling global après leurs étages de convolution, elle est généralement appliquée sur une feature map de dimensions spatiales 7×7. Pour évaluer la capacité d'un modèle à apprendre l'invariance par translation sans avoir été explicitement supervisé pour cette tâche, on propose un autre transfert. Le domaine source est encore une fois MNIST, le domaine cible, que l'on appellera MNIST-T, est une variante de MNIST où les images ont subi une augmentation géométrique de translation/rotation/mise à l'échelle 2D. On reporte les performances enregistrées pour ce transfert dans la table 5.6. Les résultats montrent une fois de plus qu'une composante architecturale simple, mais appropriée comme l'average pooling peut modifier significativement le comportement en extrapolation d'un modèle vers un domaine cible jamais observé. Il est intéressant de remarquer que lorsque ce biais inductif bénéfique est utilisé, DANN améliore encore plus la performance de source-only, alors que sans global pooling, DANN dégrade les performances par-rapport à source-only.

Le dropout [Srivastava et al., 2014] est une technique simple, mais très efficace pour régulariser la capacité des réseaux de neurones profonds. Elle consiste à mettre à zéro les activations en sortie d'une couche pendant l'entraînement avec une certaine probabilité p. Cela permet d'éviter

M→ MT	SO	DANN
NoGlobalAvgPool	0.51	0.48
GlobalAvgPool	0.8	0.95

TABLE 5.6 – Performances Source only pour deux architectures d'encodeur différentes (avec et sans pooling global) ; M=MNIST, MT=MNIST augmenté aléatoirement avec des rotations/translations/mises à l'échelle

	san pré-entraînement	
	SO	DANN
M→ MM	0.17 / 0.13	0.63 / 0.7
MM→ M	0.98 / 0.98	0.98 / 0.985
S→ M	0.65 / 0.65	0.71 / 0.74
M→ S	0.07 / 0.07	0.1 / 0.1
P→ Sk	0.15 / 0.17	0.21 / 0.22
P→ C	0.17 / 0.175	0.26 / 0.26
P→ A	0.23 / 0.23	0.22 / 0.22

	avec pré-entraînement	
	SO	DANN
M→ MM	0.6 / 0.7	0.99 / 0.98
MM→ M	0.98 / 0.98	0.98 / 0.984
S→ M	0.83 / 0.8	0.88 / 0.91
M→ S	0.25 / 0.22	0.15 / 0.14
P→ Sk	0.42 / 0.43	0.6 / 0.6
P→ C	0.22 / 0.22	0.67 / 0.67
P→ A	0.58 / 0.595	0.76 / 0.77

TABLE 5.7 – Performances dans le domaine cible sans / avec dropout ; **Table de gauche :** Non-pré-entraîné **Table de droite :** Pré-entraîné ; P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart, M=MNIST, MM=MNIST-M, S=SVHN

la coadaptation des neurones et ainsi d'augmenter la robustesse des représentations. Le dropout peut également être vu comme une façon implicite de moyenner une infinité de sous-réseaux et donc être catégorisé comme une méthode d'ensemble. Puisque le dropout est omniprésent dans les solutions de classifications standard et est reconnu pour son impact bénéfique sur la généralisation, on souhaiterait mesurer sa contribution dans le cadre plus difficile de l'adaptation de domaine. Pour ce faire, on répète une fois de plus toutes les expériences de la table 5.4 avec et sans dropout sur l'avant-dernière couche. On rassemble dans la table 5.7 l'ensemble des résultats, avec et sans dropout et avec et sans pré-entraînement. On observe dans tous les cas un impact négligeable du dropout à la fois dans le cas non pré-entraîné et dans le cas pré-entraîné.

5.3 Conclusion

Afin d'expliquer pourquoi les méthodes d'alignement de domaine fonctionnaient en l'absence de garanties théoriques, on a proposé dans ce chapitre une approche différente pour décrire le problème de l'adaptation de domaine. En effet, bien que l'on puisse voir dans les bornes théoriques de la littérature une prescription pour aligner les domaines, c'est en réalité une interprétation abusive : la performance dans le domaine cible ne bénéficie pas systématiquement de l'alignement de domaine. Ce constat oblige à trouver un autre angle d'analyse pour expliquer le relatif succès des algorithmes d'adaptation de domaine et identifier les facteurs indispensables à leur bon fonctionnement.

On a d'abord commencé par décrire les spécificités des problèmes d'adaptation de domaine posés en computer vision et montré qu'ils ne constituaient qu'un ensemble très restreint de tous les transferts possibles. On défend l'hypothèse que ces spécificités peuvent être exploitées par un biais inductif bien choisi.

On a pu étudier empiriquement le rôle d'un certain nombre d'instances du biais inductif afin d'apporter des éléments de réflexion sur les mécaniques inhérentes au transfert de domaine. On classe ces instances du biais en quatre catégories, allant des biais inhérents à l'alignement de domaine à ceux, plus évidents, comme l'architecture du réseau ou l'augmentation de données pendant l'entraînement. On a pu relever que le pré-entraînement des modèles sur un jeu de données tiers très fourni comme ImageNet était particulièrement important et ses effets difficiles à décrire :

D'une part, les méthodes d'alignement actuelles échouent très souvent sans pré-entraînement, ce qui amène à penser que le pré-entraînement aide à rapprocher sémantiquement les descripteurs source et cible de telle sorte qu'un simple ajustement est nécessaire pour les aligner correctement. D'autre part, on a pu remarquer avec l'expérience du classifieur KNN que le rôle du pré-entraînement ne se limitait pas à la seule représentation figée de départ. Pour résumer, c'est l'effet combiné du pré-entraînement et du fine-tuning sur le domaine source qui aide à aligner correctement les descripteurs.

Cette étude des instances de biais inductif trouve toutefois rapidement ses limites. D'abord, elle n'est pas exhaustive : nous nous sommes limités aux cas les plus prototypiques, les possibilités de conception étant innombrables en deep learning. Ensuite, le rôle de chaque biais inductif a été étudié de façon isolée, sur la base de problèmes jouets construits explicitement pour qu'ils en constituent une solution ad hoc. Étant donné un transfert réaliste, on ne peut en réalité pas prédire quelle combinaison de biais inductifs permettra de le résoudre. On ne peut par-ailleurs pas garantir que la combinaison de plusieurs de ces biais soit toujours bénéfique.

Les conclusions de ce chapitre constituent donc plutôt un nouvel état des lieux de ce que l'on peut comprendre des problèmes de transfert de domaine et ne sont pas directement prescriptives. Les éléments de réflexions apportés invitent tout de même à imaginer de nouveaux algorithmes construits autour de cette notion de biais inductif, ce que l'on fera dans les deux chapitres à venir.

Chapitre 6

Optimisation du biais inductif

Sommaire

Dans le chapitre précédent, nous avons considéré le problème de l'adaptation de domaine sous l'angle de la recherche d'un biais inductif adapté. Toutefois, jusqu'à maintenant, nous avons uniquement étudié des biais inductifs contrôlés par un ensemble d'hyperparamètres rigide et fixé (par exemple la présence ou non d'un élément d'architecture dans le réseau). Dans la plupart des expériences présentées, les transferts étaient par-ailleurs artificiels et avaient pour but de montrer l'influence de tel ou tel biais inductif choisi comme une solution ad hoc. Bien que certains d'entre eux, comme le pré-entraînement, semblent apporter une augmentation conséquente et systématique de la performance dans le domaine cible, il n'existe pas dans le cas général de moyen clair de choisir, ajuster et combiner ces éléments de design pour résoudre un transfert de domaine donné. De surcroît, il n'y aucune garantie pour qu'une combinaison d'à prioris et hypothèses rigides produise une solution optimale.

Dans cette section, nous explorons plusieurs méthodes ajoutant un degré de flexibilité au modèle en autorisant l'optimisation de biais inductifs paramétriques. Nous verrons d'abord que le biais inductif peut être optimisé en maximisant un objectif heuristique intermédiaire. Nous verrons ensuite que sous certaines conditions, il est également possible de maximiser directement le véritable objectif d'intérêt, qui est le risque du modèle dans le domaine cible. Dans ce dernier cas, on parle alors de méta-apprentissage.

Un certain nombre de méthodes basées sur l'optimisation d'une forme de biais inductif existent déjà : pour ce qui est de l'optimisation heuristique, on pense à la méthode SDA (pour *Select Data Augmentation*) [Ilse et al., 2020], dont le but est de trouver une fonction d'augmentation idéale pour l'adaptabilité. Pour ce qui est du méta-apprentissage, plusieurs travaux ont déjà été proposés pour l'adaptation de domaine, en particulier dans le cas de la généralisation de domaine multi-source (MS-DG) : Li et al. [2018] méta-apprend, à l'aide de plusieurs domaines sources, une paramétrisation initiale de l'encodeur qui généralise vers le domaine cible. Cette méthode a ensuite été améliorée par Balaji et al. [2018] qui propose plutôt de méta-apprendre un poids de régularisation (L1 ou L2) pour chaque paramètre scalaire du classifieur. Ces poids sont optimisés pour maximiser la performance dans le domaine cible dans un scénario de fine tuning régularisé. Enfin, Wei et al. [2021] utilise le méta-apprentissage pour apprendre une paramétrisation de l'encodeur qui synchronise naturellement les dynamiques d'apprentissage de la supervision en source et de l'alignement de domaine pour accroître la stabilité et la performance d'un scénario de type DANN, cette dernière méthode couvrant les cas de l'UDA et de la MS-DG. Le contenu de ce chapitre est rapporté de la seconde partie de notre papier [Siry et al., 2021] qui analyse la pratique de l'adaptation de domaine à travers la notion de biais inductif.

6.1 Évaluation de la méthode SDA

Dans les conditions expérimentales habituelles de l'adaptation de domaine, le véritable objectif ne peut être ni évalué, ni optimisé directement, puisque les annotations ne sont pas disponibles dans le domaine cible. De ce fait, il est nécessaire de choisir un objectif intermédiaire ou « proxy » que l'on optimisera par-rapport aux hyperparamètres contrôlant le biais inductif. La méthode SDA [Ilse et al., 2020] tombe dans cette catégorie : elle consiste à apprendre les paramètres de fonctions d'augmentation qui, si elles étaient appliquées au domaine source, généraliseraient le plus possible vers le domaine cible. Étant donné un ensemble de fonctions d'augmentations (par exemple les rotations aléatoires, flous gaussiens, distortion des couleurs, etc), SDA choisit de façon gloutonne la fonction d'augmentation qui maximise la confusion entre les domaines. La méthode procède en entraînant un classifieur à distinguer entre les deux domaines augmentés, et sélectionne la fonction d'augmentation pour laquelle la performance du classifieur est minimale (et donc la confusion entre domaines maximale). Ce procédé est ensuite répété avec toutes les augmentations restantes, composées avec la précédente, et ainsi de suite jusqu'à ce qu'un certain critère d'arrêt soit déclenché. On évalue ensuite la chaîne d'augmentations ainsi déterminée en entraînant le modèle sur le domaine source augmenté avec ladite chaîne. Enfin, on teste la performance sur le domaine cible.

Pour étudier le bien fondé de cet objectif « proxy », on évalue dans la Figure 6.1 la confusion de

domaine induite par 4 types d'augmentations (rotation aléatoire, distortion des couleurs aléatoire, bruit gaussien et crop aléatoire) sur 5 transferts. Dans chacune des configurations (augmentation, transfert), on augmente progressivement le paramètre contrôlant l'intensité de l'augmentation, de sa valeur minimale (l'augmentation est alors l'identité) à sa valeur maximale. Par exemple, dans le cas des rotations aléatoires, on augmente la portée d'échantillonage de l'ange de $\pm 0°$ à $\pm 180°$. Pour chaque degré d'intensité, nous mesurons la performance de classification de domaine comme mesure de confusion, ainsi que la performance de classification de classe pour mesurer à quel point l'augmentation dégrade l'information de classe utile.

A l'exception du transfert pacs-photo vers pacs-painting, dans lequel on observe une légère décrue dans la performance de classification de domaine à mesure que l'augmentation devient trop sévère, les domaines peuvent être parfaitement distingués avec un simple ResNet-18 pré-entraîné. On mesure cependant une chute significative de la performance de classification de classe dans la plupart des transferts.

Pour rendre la confusion des domaines plus facile, on essaie d'affaiblir le classifieur de domaines en utilisant l'architecture AlexNet, plus rudimentaire, ainsi que son optimiseur associé, en choisissant SGD plutôt que Adam, et reportons les nouvelles caractéristiques dans les figures 6.2 et 6.3. On observe que même pour ces classifieurs régularisés, la classification des domaines reste une tâche triviale, y compris lorsque les images sont largement dégradées par l'augmentation. En revanche, la performance de classification des classes est significativement plus faible qu'avec ResNet-18.

La conclusion que nous tirons de nos expériences sur SDA est donc double : 1) la performance de classification des domaines donnée par un modèle profond n'est pas une mesure informative pour mesurer la confusion des domaines, puisque les domaines source et cible ne se chevauchent pas même lorsque augmentés. 2) l'heuristique choisie par SDA pour choisir la bonne chaîne d'augmentations ne tient pas compte de la corruption éventuelle de l'information utile à la classification des classes, et ne peut donc pas garantir de gain en performance fiable dans le domaine cible.

6.2 Méta-Apprentissage

6.2.1 Définition et formalisme

Dans le chapitre précédent, nous nous sommes intéressés aux mécanismes de biais inductif pouvant avoir un impact significatif sur la capacité des modèles à transférer d'un domaine source vers un domaine cible. Cependant, cette étude est restée limitée à des biais résultant de choix purement heuristiques, conçus et choisis « à la main » (choix de l'architecture du réseau, définition d'un protocole de fine-tuning...). En procédant de la sorte, nous ne pouvons pas être sûrs que ces choix soient optimaux par-rapport à l'objectif d'intérêt, qui est la maximisation de la performance dans le domaine cible.

Le méta-apprentissage est une branche récente de l'apprentissage machine qui consiste à « apprendre à apprendre ». Plus précisément, elle cherche à apprendre le biais inductif grâce aux données elles-mêmes, en l'optimisant par-rapport au but recherché. Dans ce chapitre, on présente diverses applications du principe de méta-apprentissage au problème de l'adaptation de domaine.

On définit une tâche T comme étant un problème de classification à N classes, muni de son ensemble d'entraînement et de son ensemble de test. Étant donné une distribution de tâches $p(T)$, l'objectif d'un modèle de méta-apprentissage est d'apprendre à apprendre correctement, en moyenne, l'ensemble des tâches $T \sim p(T)$, c'est-à-dire avoir une bonne performance de test après avoir vu l'ensemble d'entraînement d'une tâche T donnée. Pour justifier l'entraînement d'un biais inductif utile, on suppose que les tâches de $p(T)$ possèdent un certain nombre de caractéristiques communes.

Un modèle de méta-entraînement est défini par deux ensembles de paramètres θ et Φ :

— θ, également appelé « fast weight » contient les paramètres spécifiques à l'apprentissage d'une tâche T_i présentée à la volée, cela signifie qu'il est autorisé à être entraîné en θ_i après avoir observé les données relatives à cette tâche. θ_i est obtenu selon un processus appelé « boucle

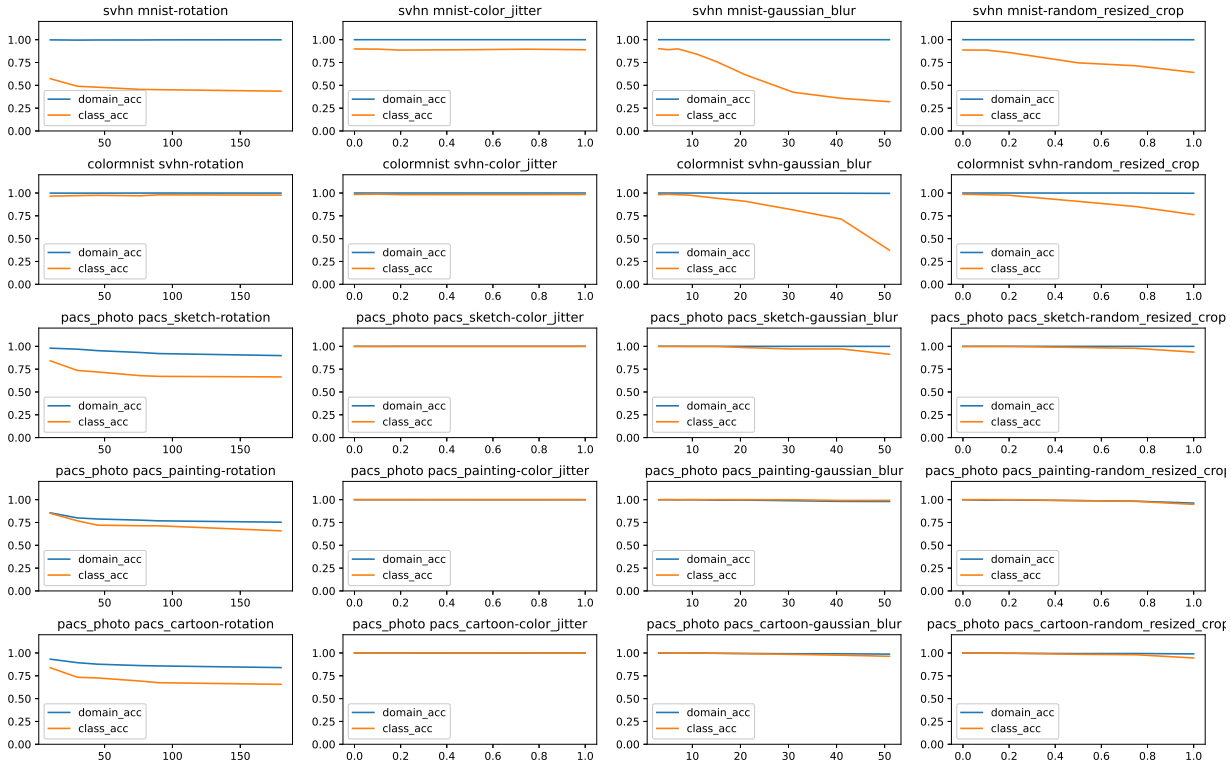

FIGURE 6.1 – Augmentation de la confusion entre les domaines pour un ensemble de transferts et d'augmentations. Malgré la hausse du paramètre d'augmentation, un simple ResNet-18 parvient facilement à distinguer les domaines parfaitement. On fournit également la performance de classification sur le domaine source comme mesure de la corruption de l'image.

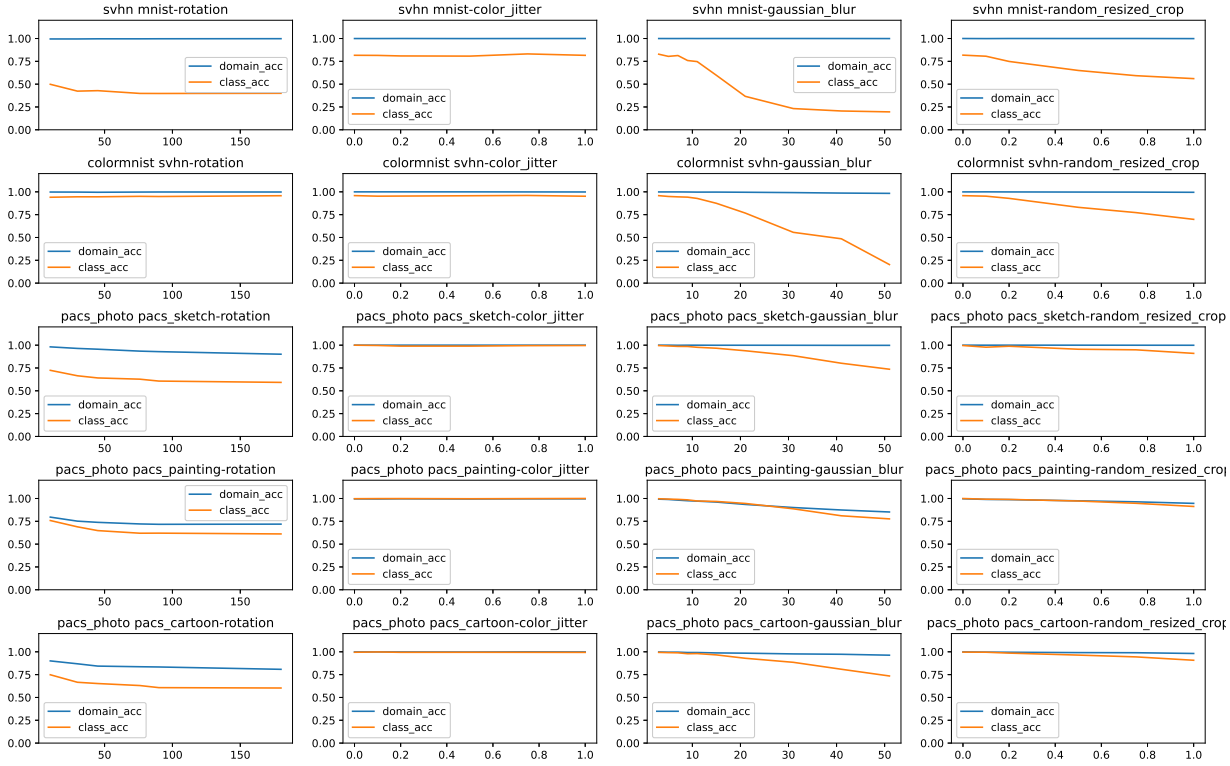

FIGURE 6.2 – Même expérience que dans la Figure 6.1, mais avec un AlexNet entraîné avec Adam

interne ». En pratique, on initialise d'abord θ à partir uniquement de Φ et/ou d'une source aléatoire, puis on entraîne ce θ en θ_i.

— Φ est le paramètre agnostique à la tâche, il est aussi appelé « méta-paramètre » ou « slow weight ». Il n'est pas autorisé à changer lorsqu'une tâche T_i est présentée, mais est méta-optimisé à une valeur fixe qui maximise la performance moyenne de (θ_i, Φ) sur chaque T_i. Ce procédé est appelé « méta-optimisation » ou « boucle externe »

On peut résumer le principe du méta-apprentissage par les deux équations suivantes :

$$\Phi^\star = \underset{\Phi}{\operatorname{argmin}} \underset{T_i \sim p(T)}{\mathbf{E}} [\mathcal{L}(\theta_i, \Phi, T_i^{test})]$$
$$\theta_i = innerloop(\Phi, T_i^{train})$$

N'importe quel prototype de modèle de méta-apprentissage peut donc être défini par une paramétrisation de θ, Φ et une définition des fonctions *innerloop* et $\mathcal{L}$. On s'attend à ce qu'un modèle de méta-apprentissage bien pensé puisse battre n'importe quel biais inductif défini heuristiquement une fois entraîné. Cependant, pour démontrer son utilité, le méta-apprentissage doit être entraîné sur des tâches T_i différentes des tâches de test. On appelle ces ensembles de tâches ensemble de « méta-entraînement » et ensemble de « méta-test ». On espère donc qu'un $\Phi^\star$ optimisé sur l'ensemble de méta-entraînement soit également optimal sur l'ensemble méta-test. Comme en apprentissage classique, il existe donc un risque de « méta-surapprentissage » [Rajendran et al., 2020] sur les tâches de méta-entraînement qu'il faut pouvoir contrôler, par exemple en augmentant la diversité des tâches de méta-entraînement, ou en choisissant intelligemment la paramétrisation de θ et de Φ.

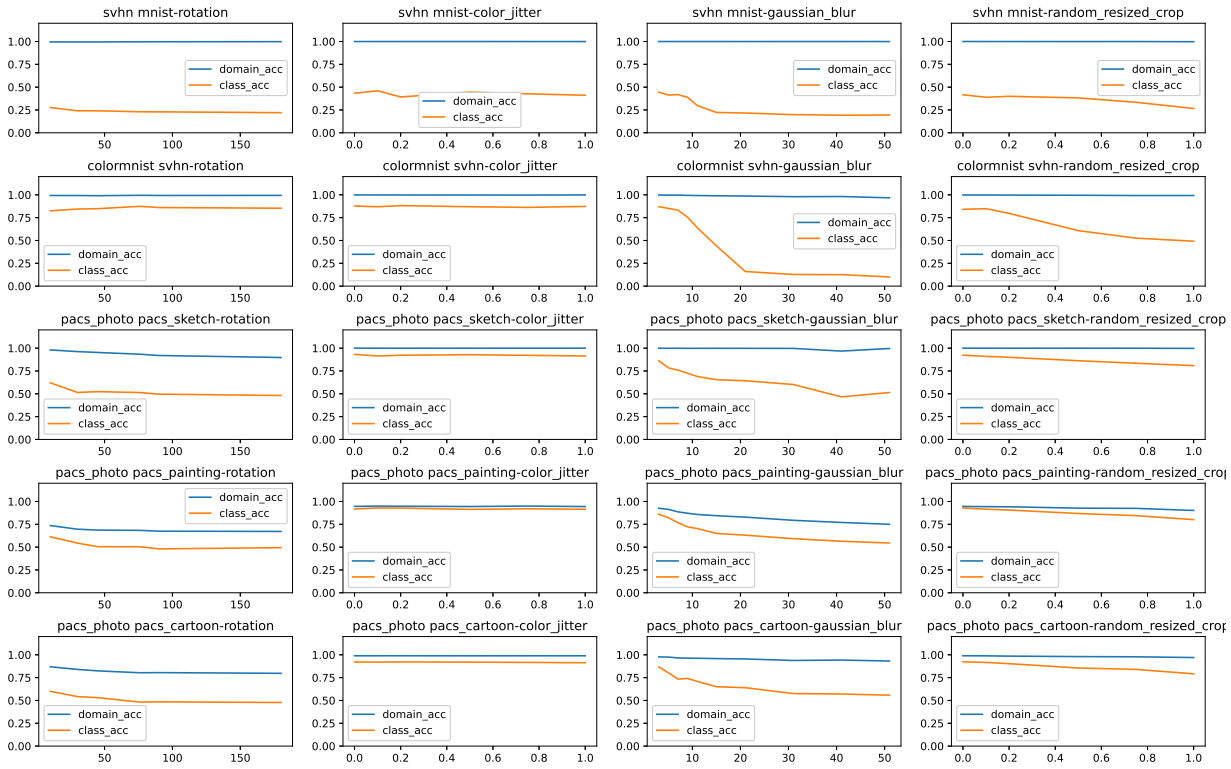

FIGURE 6.3 – Même expérience que dans la Figure 6.1, mais avec un AlexNet entraîné avec SGD

6.2.2 Utilisation du méta-entraînement pour l'adaptation de domaine

Historiquement, le méta-apprentissage a surtout été proposé pour résoudre les problèmes de few-shot learning, c'est-à-dire des tâches de généralisation classique. Le méta-apprentissage est un framework extrêmement flexible que l'on peut appliquer à n'importe quel scénario, en particulier l'adaptation de domaine.

Pour nos besoins, on propose d'utiliser le méta-apprentissage pour maximiser la performance en cible d'un entraînement « Source-Only », cela signifie que pour toute tâche T_i, $T_i^{train} = \mathcal{S}$ et $T_i^{test} = \mathcal{T}$. Autrement dit, le modèle pourra optimiser θ_i en boucle interne en exploitant les échantillons du domaine source, et sera méta-optimisé pour que le Φ, θ_i soit performant sur la même tâche de classification, mais dans le domaine cible. Il faut noter que d'autres méthodes de méta-apprentissage appliquées à l'adaptation de domaine n'utilisent pas le scénario d'entraînement Source-Only pour leur boucle interne, par exemple l'algorithme de Wei et al. [2021] méta-apprend à conditionner l'alignement des domaines, et se base davantage sur les conditions expérimentales de l'UDA. Pour notre part, nous favorisons le scénario Source-Only pour sa simplicité et ses conditions expérimentales peu contraignantes : pas besoin d'échantillons cible non-annotés pour l'entraînement en boucle interne. Nous allons maintenant définir en détail les deux composantes de notre méthode de méta-apprentissage.

Définition de la boucle interne : Nous empruntons dans notre algorithme la même paramétrisation que MAML [Finn et al., 2017] : il s'agit de méta-entraîner l'initialisation θ^0 d'un réseau de neurones choisi, afin que K étapes de descente du gradient à partir de cette initialisation sur T_i^{train} permettent d'obtenir un paramètre final $\theta^K = \theta_i$ qui soit performant dans le domaine cible.

Pour raccorder la définition de MAML au formalisme du méta-apprentissage établi plus haut, on peut considérer que $\Phi = \theta^0$ et $\theta_i = \theta^K$.

Optimisation de la boucle externe : Pour optimiser θ^0 en $\theta^{0\star}$, on déroule le graphe différentiable des opérations successives de *boucleinterne*, on évalue le méta-objectif à minimiser par-rapport à θ^0, enfin, on réalise une itération de descente du gradient de cet objectif par-rapport à θ^0, jusqu'à convergence vers $\theta^{0\star}$.

Nous résumons notre boucle de méta-entraînement en pseudo-code dans la figure 1 :

Algorithm 1 MAML-2DOM

Require: γ : learning rate interne, η : learning rate externe, $\mathcal{S}$ domaine source, $\mathcal{T}$ domaine cible, Y_{MTrain} ensemble de toutes les classes de méta-apprentissage

for $0 \leq i < n_{iters}$ **do**
  $Y_{T_i} \leftarrow$ **sample_10_classes**(Y_{MTrain})
  $T_i^{train} \sim$ **get_samples_of_said_classes**$(\mathcal{S}, Y_{T_i})$
  $T_i^{test} \sim$ **get_samples_of_said_classes**$(\mathcal{T}, Y_{T_i})$
  $\theta \leftarrow \theta^0$
  for $0 \leq j < K$ **do**
    $\theta \leftarrow \theta - \gamma \frac{\partial \mathcal{L}(\theta, T_i^{train})}{\partial \theta}$
  end for
  $\theta^0 \leftarrow \theta^0 - \eta \frac{\partial \mathcal{L}(\theta, T_i^{test})}{\partial \theta^0}$
end for

Définition des ensembles méta-entraînement et de méta-test : Jusqu'à maintenant, notre boucle d'apprentissage est en tous points identique à celle proposée par Li et al. [2018]. Il reste cependant à définir exactement quelles tâches que nous souhaitons résoudre, et surtout, vers quelles tâches de test nous souhaitons que le biais inductif généralise. Plusieurs possibilités se présentent à nous :

— Li et al. [2018] essaient de résoudre un problème de multi-source domain generalization. Dans ces conditions, on dispose de $P-1$ domaines source et 1 seul domaine cible. Dans ce cas, pour synthétiser une tâche d'entraînement, on sélectionne 2 domaines parmi les $P-1$ domaines sources, que l'on utilise comme T^{train} et T^{test}. Une seule tâche est évaluée en méta-test : elle utilise l'union des $P-1$ domaines source comme T^{train} et l'unique domaine cible comme T^{test}. Les mêmes classes sont utilisées en méta-entraînement et en méta-test. Dans ce cas de figure, le méta-apprentissage est considéré comme un moyen de croiser intelligemment les différents domaines source pour résoudre un unique problème de classification vers un domaine cible inconnu.

— Dans notre contribution, nous définissons un cas d'usage très différent. On ne dispose que de deux domaines à partir desquels on peut échantillonner des tâches. Dans chacun des deux domaines, on dispose de P classes de méta-entraînement et Q classes de méta-test. Pour synthétiser une tâche de méta-entraînement, on choisit N classes parmi les P, puis on définit T^{train}, T^{test} comme l'ensemble des échantillons du domaine 1 (resp. du domaine 2) appartenant à l'une de ces classes. Pour synthétiser une tâche de méta-test, on procède de la même façon, mais en piochant parmi les Q classes de méta-test. Notre but ici est d'apprendre un biais inductif implémentant la capacité d'adapter d'un domaine choisi vers un autre domaine choisi, et ce, indépendamment des types d'objets à détecter, ce qui est en lien direct avec le chapitre précédent. Nous définissons ce cadre d'entraînement et d'évaluation par l'acronyme **MAML-2DOM**.

6.2.3 Expériences

Jeu de données utilisé pour le méta-entraînement : Pour satisfaire les conditions expérimentales énoncées dans la section précédente, nous utilisons les images du dataset VisDA. VisDA est un dataset comportant 6 domaines plus ou moins différents entre eux : « Real », des photographies réelles, « Painting », des peintures assez réalistes, « Sketch », des crayonnés assez

réalistes, « quickdraw », des dessins très sommaires, « clipart » des dessins cartoonesques et « infograph », des posters explicatifs incluant l'objet d'intérêt, voire l'annexe A pour en visualiser quelques échantillons. Chacun de ces domaines contient les mêmes 345 classes. Ce dataset est donc idéal pour synthétiser des tâches de transfert variées. Dans nos expériences, nous désignons les 200 premières classes comme étant les classes de méta-entraînement et les 145 classes restantes comme classes de méta-test. Toutes les tâches que nous synthétiserons seront des problèmes de classification à 10 classes.

Entraînement de MAML-2DOM : Etant donné une paire de domaines $\mathcal{S}$ et $\mathcal{T}$ choisie dans VisDA, nous méta-entraînons une instance de MAML-2DOM pour résoudre ce transfert et ce transfert uniquement. Pour ce faire, on génère une tâche en échantillonnant 10 classes parmi les 200 de méta-entraînement, on simule une boucle fermée d'entraînement source-only à partir d'échantillons du domaine source, on optimise ensuite la performance du modèle obtenu après boucle fermée avec des échantillons du domaine cible, puis on optimise cet objectif par-rapport au méta-paramètre en rétropropageant à travers tout le graphe des opérations de la boucle fermée.

Variante MAML-ALL : Nous avons défini MAML-2DOM comme un biais inductif spécialisé dans la résolution d'un seul transfert. En réalité, rien ne nous empêche d'entraîner un seul méta-paramètre qui serait capable de résoudre n'importe quel transfert $\mathcal{D}_i \to \mathcal{D}_j$, avec $\mathcal{D}_i, \mathcal{D}_j$ deux domaines quelconques piochés parmi les 6 de VisDA. On pourrait même faire l'hypothèse qu'un méta-paramètre spécialisé dans la résolution d'un plus grand nombre de transferts généralise mieux vers les tâches de méta-test, puisque la résolution d'une tâche plus difficile ou un plus grand nombre de tâches pourrait agir comme une forme de régularisation. Nous baptisons cette méthode à MAML-ALL et comparerons sa performance pour chaque transfert à celle du MAML-2DOM spécialisé.

Détails d'implémentation : Entraîner un modèle de méta-apprentissage est extrêmement coûteux en mémoire et en temps de calcul. En effet, pour chaque pas de méta-optimisation, il faut pouvoir simuler et maintenir le graphe de toutes les opérations de l'apprentissage en boucle fermée. Le coût en mémoire et en temps d'un pas de méta-optimisation augmente donc linéairement par-rapport à la taille du modèle et le nombre de pas de la boucle interne. Pour gagner du temps de calcul, nous proposons de réduire considérablement la taille du méta-modèle en évitant de travailler directement sur l'espace des images. En effet, cela supposerait d'utiliser un réseau convolutif profond, nécessitant beaucoup d'opérations et de mémoire. On choisit donc plutôt de faire travailler le méta-modèle sur les descripteurs produits par un ResNet-18 pré-entraîné, en ayant mesuré au préalable que ces descripteurs soient suffisamment informatifs pour résoudre les tâches en question. Cela permet 1) de gagner du temps en pré-encodant tout VisDA en un jeu de données de descripteurs et 2) d'utiliser un méta-modèle beaucoup plus petit : dans notre cas, un perceptron à deux couches cachées.

Baselines : Il convient de comparer nos modèles à plusieurs baselines de référence afin de démontrer la supériorité des biais inductifs paramétriques. Sauf indication contraire, les baselines utiliseront la même architecture que le méta-modèle et fonctionneront également sur les mêmes descripteurs issus du modèle pré-entraîné.

— Initialisation aléatoire (Random) : Un classifieur initialisé aléatoirement, puis entraîné en source-only sur les échantillons du domaine source de la tâche à 10 classes.

— Pré-entraînement suivi de fine-tuning (PT+SO) : L'initialisation aléatoire n'est peut-être pas une baseline suffisamment avantagée, car elle n'a pas pu exploiter l'information provenant des 200 classes de méta-entraînement. Nous proposons donc de pré-entraîner le classifieur sur un unique problème consistant à distinguer les 200 classes de méta-entraînement sur l'union des deux domaines du transfert considéré. On remplace ensuite la dernière couche puis on fine-tune sur la tâche de test à 10 classes.

— DANN : On mène le transfert à 10 classes en utilisant à la fois les données annotées du domaine source, ainsi que les données non-annotées du domaine cible pour satisfaire une contrainte d'alignement sur une couche intermédiaire du classifieur dans le but d'améliorer sa transférabilité.

	Random+SO	PT+SO	DANN	PT+DANN	CAN	ASAN	MAML-2DOM	MAML-ALL
R→Q	0.185 ± 0.036	0.197 ± 0.053	0.191 ± 0.033	0.300 ± 0.082	0.28 ± 0.033	0.26 ± 0.023	**0.542** ± 0.010	0.353 ± 0.005
R→P	0.708 ± 0.065	0.641 ± 0.065	0.731 ± 0.067	0.728 ± 0.050	0.50 ± 0.051	0.61 ± 0.043	**0.767** ± 0.009	0.714 ± 0.014
R→S	0.501 ± 0.053	0.447 ± 0.043	0.536 ± 0.054	0.576 ± 0.082	0.58 ± 0.063	0.68 ± 0.016	**0.681** ± 0.015	0.505 ± 0.011
R→C	0.620 ± 0.059	0.543 ± 0.061	0.656 ± 0.082	0.640 ± 0.071	0.62 ± 0.054	0.70 ± 0.033	**0.746** ± 0.008	0.645 ± 0.007
R→I	0.359 ± 0.086	0.313 ± 0.032	0.446 ± 0.067	0.415 ± 0.062	0.34 ± 0.027	**0.71** ± 0.017	0.502 ± 0.005	0.431 ± 0.008

TABLE 6.1 – Performances moyenne dans le domaine cible sur les problèmes de transfert à 10 classes de l'ensemble de méta-test (nouvelles classes, même paire de domaines), les statistiques sont moyennées sur 10 tâches de test (mais à partir d'une seule réalisation du méta-entraînement/pré-entraînement pour les méthodes concernées). La meilleure performance est surlignée en gras ; R=Real, Q=Quickdraw, S=Sketch, C=Clipart, I=Infograph, P=Painting

— Pré-entraînement suivi de DANN (PT+DANN) : Nous combinons l'initialisation pré-entraînée décrite plus haut, et l'alignement avec DANN sur la tâche d'intérêt.

— CAN et ASAN [Kang et al., 2019], [Raab et al., 2020] sont des méthodes état-de-l'art d'adaptation de domaine single-source. Nous utilisons l'implémentation par défaut fournie par les auteurs.

Dans tous les cas, nous favorisons les baselines autant que possible en choisissant le meilleur optimiseur, pas d'apprentissage et nombre d'itérations pour le fine-tuning. Pour renforcer davantage notre résultat, nous comparons notre méta-modèle à deux méthodes état-de-l'art issues de la littérature.

Résultats : Nous exposons les résultats sur la table 6.1. Exception faite du transfert real→infograph, le MAML-2DOM spécialisé pour chaque transfert bat largement toutes les baselines, en particulier sur le transfert real→quickdraw, qui est particulièrement difficile et pour lequel les descripteurs de ResNet-18 ne semblent pas transférables, nécessitant une correction de la part du classifieur subséquent par l'intermédiaire du biais inductif. En ce qui concerne les baselines, que ce soit dans les cas de l'initialisation aléatoire ou de l'initialisation pré-entraînée, DANN permet généralement de gagner quelques pourcents de précision par-rapport à Source-Only, mais ces gains demeurent incomparables avec ceux de MAML-2DOM. Du reste, le fait de pré-entraîner le classifieur sur l'ensemble de meta-train ne semble pas apporter de gain significatif par-rapport à l'initialisation aléatoire. De façon surprenante, on constate que MAML-ALL est globalement moins bon que MAML-2DOM, on n'observe donc pas de synergies positives dans le fait de savoir résoudre plusieurs transferts à la fois. Autre résultat important, en testant MAML-ALL sur des transferts impliquant de nouveaux domaines (et pas seulement de nouvelles classes), le bénéfice du méta-apprentissage n'est plus observé, voire dégrade les performances par-rapport à la baseline Random+SO.

6.3 Conclusion

Dans ce chapitre, nous nous sommes intéressés à une famille d'algorithmes construits autour de la notion de biais inductif optimisable. Pouvoir optimiser un prototype de biais inductif paramétrique permet d'obtenir un degré de flexibilité supplémentaire, puisque cela permet d'obtenir des instances de biais inductif composites, non triviales, qui faisaient défaut à notre étude du chapitre 5. Nous avons dégagé deux types d'objectifs qu'il est possible d'optimiser pour obtenir de tels biais inductifs.

D'une part, on peut optimiser des objectifs heuristiques (donc différents de la performance dans le domaine cible), dont on imagine qu'ils aideront tout de même à trouver un biais inductif utile pour adapter vers le domaine cible. Parmi les méthodes se référant à cette catégorie, nous nous sommes intéressés à l'algorithme SDA, qui consiste à construire une chaîne de fonctions d'augmentation destinée à augmenter le domaine source de façon optimale pour généraliser vers le domaine cible.

Il nous est apparu que le mécanisme d'optimisation de SDA était inopérant lorsque les domaines avaient un support disjoint, ce qui rend la méthode caduque.

D'autre part, on peut optimiser directement l'objectif d'intérêt (c'est-à-dire la performance dans le domaine cible). C'est évidemment impossible si on ne dispose que des données de la tâche d'intérêt, puisque les annotations du domaine cible ne sont par définition pas accessibles. Toutefois, en supposant qu'on ait accès à grand nombre de tâches de méta-entraînement entièrement annotées, on peut se placer dans les conditions expérimentales des méthodes de meta-learning qui ouvrent un nouvel éventail de possibilités. Nous proposons de reprendre la boucle d'entraînement proposée par Li et al. [2018], en détournant son utilisation pour d'autres cas d'usage. Après avoir méta-entraîné ce modèle sur les tâches de méta-entraînement, nous avons constaté que 1) le biais inductif appris était très utile pour apprendre des tâches de test impliquant la même paire de domaines, mais avec de nouvelles classes, dépassant de loin les méthodes d'alignement traditionnelles, en particulier sur les transferts difficiles tels que Real $\rightarrow$ Quickdraw. 2) le biais inductif appris n'était en revanche pas apte à généraliser vers l'adaptation de tâches impliquant de nouveaux domaines. En définitive, le méta-learning apparaît comme une piste prometteuse en termes de gains en performance, mais avec deux défauts notoires : il requiert des données tierces difficiles à obtenir pour construire un ensemble de méta-entraînement (beaucoup de classes et beaucoup de domaines) et ses propriétés de généralisation vers de nouvelles tâches sont difficiles à anticiper.

Dans la prochaine partie, nous allons plutôt nous concentrer sur la construction du biais inductif par l'exploitation de données tierces en grande quantité, mais faciles à obtenir (par exemple des images non-annotées).

Chapitre 7

Représentations désentrelacées pour l'adaptation de domaine

Sommaire

Dans ce chapitre, nous poursuivons la recherche d'un mécanisme de biais inductif favorable à la transférabilité. Nous nous intéressons cette fois à un autre pan important de la littérature de l'apprentissage non supervisé qui peut facilement s'adapter au cas de l'adaptation de domaine : l'apprentissage de représentations désentrelacées. Apprendre une représentation de l'image qui démêle l'information de classe et de domaine est une idée qui a déjà pu être exploitée dans la littérature, avec quelques variations dans la construction du modèle et le choix des fonctions de coût à minimiser [Peng et al., 2019, Bousmalis et al., 2016, Cai et al., 2019, Gonzalez-Garcia et al., 2018, Fu et al., 2017, Lee et al., 2021]. L'intérêt de telle ou telle variation n'est pas toujours discuté par les auteurs : dans ce chapitre, nous proposons une version simplifiée à l'extrême de l'algorithme, puis nous l'étudierons en détail pour identifier les facteurs qui contribuent à l'adaptation.

7.1 Introduction

On dit qu'un descripteur (ou une représentation) est désentrelacé lorsque les facteurs latents « interprétables » sont séparés dans des sous-espaces orthogonaux au sein de l'espace latent Z. Etant donné une distribution d'images non-annotées $\mathcal{D}_X$ pouvant être expliquée par quelques facteurs latents interprétables, le but de ces méthodes est d'entraîner un encodeur $\Psi : X \to Z$ qui puisse prédire la valeur de ces facteurs latents en les présentant de manière désentrelacée.

Prenons par exemple le jeu de données MNIST : le premier facteur latent serait par exemple la classe (il serait donc à valeurs discrètes), le second pourrait contrôler l'épaisseur du trait, le troisième l'inclinaison de la police d'écriture et ainsi de suite.

La découverte non-supervisée des facteurs latents interprétables est un des problèmes fondamentaux en vision artificielle. Ce principe de désentrelacement trouve une application immédiate pour l'adaptation de domaine non supervisée en considérant le mélange des deux domaines $\mathcal{D}_X = \frac{\mathcal{S}_X + \mathcal{T}_X}{2}$. En effet, pour décrire cette distribution d'images, le premier facteur latent serait la classe, tandis que les autres facteurs latents contrôleraient le style de l'échantillon, en particulier le domaine auquel il appartient. Si une telle factorisation peut-être trouvée par une méthode de désentrelacement, alors le problème de l'UDA est *de facto* résolu. En effet, puisque l'information de la classe est séparée de celle du domaine, un classifieur entraîné sur cet espace latent devrait en principe avoir une bonne performance sur le domaine cible.

Définir mathématiquement la notion de facteur interprétable n'est pas chose aisée. C'est pourquoi les méthodes existantes essaient d'optimiser des critères heuristiques, dont on imagine qu'ils favoriseront l'émergence de tels facteurs latents dans les dimensions de Z.

D'abord, une telle représentation latente se doit d'être exhaustive, ce qui peut être vérifié en reconstruisant l'image $x \in \mathcal{D}_X$ à partir de sa représentation latente avec un décodeur Φ, soit $\Phi \circ \Psi(x) \approx x$. Pour structurer l'espace latent en facteurs interprétables, il est très courant de vouloir forcer l'indépendance statistique des dimensions de l'espace latent Z en complément du critère de reconstruction. En d'autres termes, $p(z) = \Psi(\mathcal{S}_X) = \prod_1^N p(z_i)$ avec N le nombre de dimensions de l'espace latent. Dans la plupart des méthodes, on impose simplement un à-priori gaussien : $p(z) = \mathcal{N}(0, \mathbf{I}^N)$.

Les premières méthodes à avoir appliqué le critère d'indépendance des dimensions pour la modélisation de facteurs latents sont l'autoencodeur adversaire [Makhzani et al., 2015] et le β-VAE [Higgins et al., 2017]. Notons que la plupart de ces méthodes ont également été pensées pour la génération d'image par échantillonnage ancestral à partir d'un à-priori connu sur Z i.e. $\mathcal{D}_X = \Phi(p(z))$.

Remarque 7.1. *Rappelons que l'indépendance des dimensions n'est qu'un objectif heuristique, on peut facilement se convaincre que les facteurs parfaitement indépendants ne sont en réalité pas nécessairement interprétables et vice-versa, divers exemples réfutant cette assertion ont pu être proposés par Locatello et al. [2018], Mathieu et al. [2018]. Cependant, les méthodes imposant ce*

critère connaissent des succès pratiques avec des jeux de données où les facteurs sont suffisamment évidents.

7.2 Approche proposée

L'algorithme proposé dans cette section est très proche de travaux antérieurs : Bousmalis et al. [2016] propose d'utiliser d'une part des encodeurs séparés pour le domaine source et le domaine cible produisant respectivement l'information spécifique à chaque domaine (le style) et d'autre part un encodeur partagé extrayant l'information commune (la classe). Leur méthode impose la reconstruction, minimise pour chaque domaine l'information mutuelle entre descripteurs communs et spécifiques, mais n'impose pas de prior $p(z)$ sur l'ensemble de l'espace latent. Faisons enfin remarquer que cette méthode est relativement ancienne et que pour cette raison, sa manière d'imposer un critère d'indépendance entre les différentes sources d'information pourrait être améliorée. Peng et al. [2019] utilise un encodeur commun. Enfin, Cai et al. [2019] propose une architecture plus simple, appelée DSR, avec un encodeur et un espace latent partagé par les deux domaines et séparé en deux sous-espaces, le premier contenant l'information de classe et le second l'information de domaine. La méthode impose la reconstruction et l'indépendance statistique entre ces deux sources d'information. Elle impose également $p(z) = \mathcal{N}(0, \mathbf{I}^N)$ avec l'ELBO classique du VAE Kingma and Welling [2014], mais dans une des couches précédentes.

Pour cette étude, nous proposons un algorithme simplifié au maximum qui puisse satisfaire les trois objectifs suivants : 1) imposer la reconstruction, 2) séparer l'information de classe et de domaine et 3) imposer $p(z) = \mathcal{N}(0, \mathbf{I}^N)$ sur l'espace latent. Nous choisissons donc d'écarter tous les choix d'implémentation vus dans l'état-de-l'art ajoutant de la complexité sans justification particulière. Nous retenons finalement les éléments suivants : un encodeur commun aux deux domaines produisant l'espace latent, lequel est séparé en deux sous-espaces que nous forçons à être indépendants : un pour l'information de classe et un pour l'information de domaine. Contrairement à la méthode DSR, c'est sur ce même espace latent que nous choisissons d'imposer $p(z) = \mathcal{N}(0, \mathbf{I}^N)$. Enfin, bien évidemment, un décodeur commun pour la reconstruction. L'algorithme, que nous appellerons DISTGL, contraction de « disentangled » est résumé dans la figure 7.1.

À l'instar de DSR, nous utilisons 4 classifieurs $dom1, dom2, class1, class2$ pour désentrelacer l'information de classe et de domaine : $class1$ et $dom2$ sont entraînés conjointement avec Ψ pour prédire respectivement la classe (resp. le domaine) de l'échantillon à partir de f1 (resp. de f2). De cette façon, nous nous assurons que le vecteur $f1$ contient l'information de classe et $f2$ contient l'information de domaine. $class2$ et $dom1$ sont également entraînés à prédire la classe (resp. le domaine) à partir de f2 (resp. de f1), mais cette fois comme adversaires de Ψ. Nous nous assurons de cette façon que $f1$ ne contient pas l'information de domaine et $f2$ ne contient pas l'information de classe.

Contrairement à DSR, nous choisissons d'imposer $p(z) = \mathcal{N}(0, \mathbf{I}^N)$ à l'aide d'un troisième adversaire *dis* qui s'entraîne à distinguer les codes latents de vrais échantillons de $\mathcal{N}(0, \mathbf{I}^N)$. En cela, nous réalisons un AAE [Makhzani et al., 2015] plutôt qu'un VAE. Ce choix fut motivé par une plus grande stabilité d'optimisation par-rapport aux autres algorithmes que nous avons pu tester (VAE et BiGAN [Donahue et al., 2016] notamment).

Remarque 7.2. *La répartition de l'information de domaine entre $f1$ et $f2$ peut être entièrement contrôlée, puisque le label de domaine est disponible pour tous les échantillons. En revanche, la répartition de l'information de classe ne peut être supervisée que pour le domaine source. On espère du reste que l'ensemble des régularisations imposées par l'algorithme conditionneront le domaine cible à factoriser son information de classe de la même façon (ce qui est indispensable pour l'UDA).*

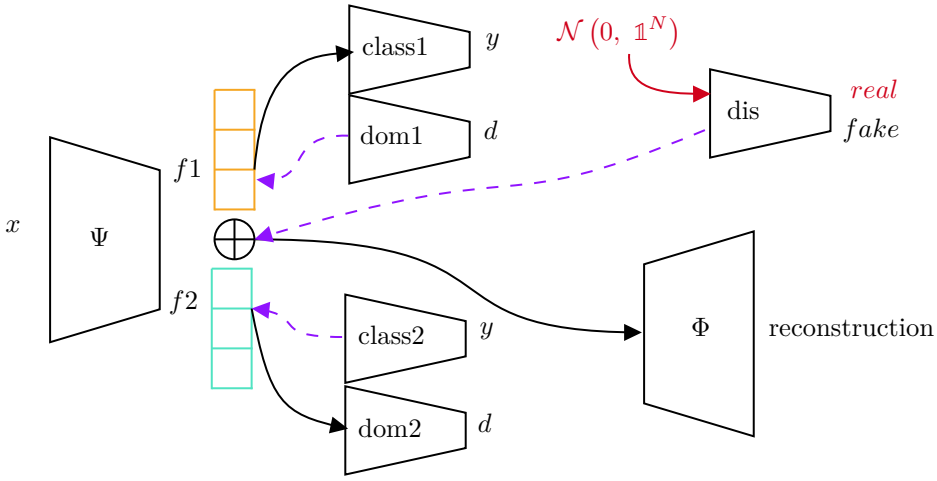

FIGURE 7.1 – Schéma de la méthode proposée ; Flèches noires = entraînement conjoint ; Flèches violettes = entraînement adversaire

7.3 Expériences

Nous évaluons l'algorithme DISTGL dans le cadre de l'UDA sur différents types de transferts. Dans un premier temps, nous le comparerons à Source-Only et DANN, puis dans un second temps, nous réaliserons diverses études ablatives pour évaluer l'importance de chaque élément le constituant.

Jeux de données : Nous utilisons pour cette étude les jeux de données de chiffres manuscrits : MNIST, COLOR-MNIST, MNIST-T et SVHN (voir annexe A pour une présentation détaillée de ces jeux de données). Nous nous restreignons à ce type d'images pour trois raisons : 1) les algorithmes de modélisation latente nécessitent un grand nombre d'images, 2) ils ont du mal à modéliser les jeux de données trop complexes visuellement (reconstruction floue) et 3) Ces images sont en 32×32 ce qui accélère grandement la vitesse des entraînements. Même avec ces jeux de données, il est possible de proposer des transferts relativement difficiles pour des modèles non pré-entraînés.

Détails d'implémentation Pour l'encodeur Ψ, nous utilisons un petit ResNet à 4 couches résiduelles dimensionné pour les images au format 32×32. Sauf indication contraire, nous utilisons pour le décodeur Φ un autre ResNet, qui est le symétrique de Ψ avec des couches déconvolutives. Les classifieurs sont tous les réseaux entièrement connectés avec le même nombre de couches et le même nombre de neurones dans chaque couche cachée.

Concernant l'entraînement adversaire, nous conservons les choix d'implémentation habituels : tous les classifieurs adversaires $dom1$, $class2$, dis utilisent la normalisation spectrale et sont entraînés avec 5 itérations pour chaque itération de Φ, Ψ, $class1$, $dom2$. Ψ optimise la loss non-saturante pour combattre ses adversaires à 2 classes, $dom1$ et dis et maximise l'entropie de classification de $class2$, qui a plus de deux classes en sortie.

Pour satisfaire au mieux l'ensemble des critères, nous choisissons de minimiser le compromis suivant, on fixe λ à 0.01 après plusieurs tentatives, cette valeur étant celle qui satisfaisait le mieux les différents objectifs.

$$\min_{\Psi,\Phi,class1,dom2} \mathcal{L}_{rec} + \lambda(\mathcal{L}_{dis} + \mathcal{L}_{class1} + \mathcal{L}_{class2} + \mathcal{L}_{dom1} + \mathcal{L}_{dom2}) \tag{7.1}$$

Avec ce réglage, l'optimisation de la plupart des critères ne pose pas de problème. Les reconstructions sont de qualité satisfaisante, $class1$ et $dom2$ atteignent rapidement une performance de classification maximale et $dom1$ et $class2$ sont quant à eux maintenus proches du niveau de confusion maximum : autour de 55% d'accuracy pour $dom1$ (la confusion totale étant atteinte

à 50%) et 20% d'accuracy pour $class2$ (la confusion totale étant atteinte à 10%). Cependant, l'entraînement adversaire relatif à dis est en pratique beaucoup plus difficile à faire converger : en effet, nous avons initialement choisi de fixer le nombre de dimensions de Z à 256 : dans ce cas dis se maintient à une bonne accuracy de discrimination (80%). En réduisant ce nombre à 32, la performance de dis peut-être ramenée à 56%, c'est-à-dire plus près du seuil de confusion. Nous constatons par-ailleurs cette amélioration dans la qualité des echantillons générés. Enfin, nous avons essayé de remplacer l'entraînement adversaire classique par la variante plus stable de l'algorithme WGAN-GP [Gulrajani et al., 2017], mais nous ne l'avons pas retenue faute de gain en performance significatif.

Nous retrouvons les baselines Source-Only et DANN comme l'optimisation d'un sous-ensemble de ces fonctions de perte. Ainsi, pour implémenter SO, nous n'optimiserons que $\mathcal{L}_{class1}$ et pour implémenter DANN, nous n'optimiserons que $\mathcal{L}_{class1}$ et $\mathcal{L}_{dom1}$. Cela permet de comparer les méthodes toutes choses égales par-ailleurs.

7.3.1 DISTGL comparée aux méthodes classiques

Nous commençons par comparer la méthode DISTGL aux baselines habituelles de l'adaptation de domaine, à savoir Source-Only et DANN, sur un ensemble de transferts. Nous choisissons la plupart du temps MNIST comme domaine source et un dataset plus riche comme domaine cible afin d'obtenir des transferts qui ne soient pas triviaux en Source-Only. L'ensemble des résultats sont récapitulés dans la table 7.1.

Nous remarquons dans un premier temps que DANN bat systématiquement Source-Only, ce qui confirme l'efficacité des mécanismes adversariaux de l'implémentation sur ce type de transfert. Il est important de remarquer que sur MNIST→MNIST-M, cette implémentation de DANN bat largement l'originale [Ganin et al., 2016] et bat également une partie des méthodes de désentrelacement classe-domaine existantes [Bousmalis et al., 2016, Lee et al., 2021], ce qui montre une fois de plus l'extrême sensibilité des résultats aux facteurs extérieurs à la méthode (type d'encodeur, implémentation du mécanisme adversaire...). Dans un second temps, nous constatons une très mauvaise performance de DISTGL par-rapport à DANN. Ce résultat ne peut être imputé à un problème d'optimisation, puisque au terme de l'entraînement de DISTGL, $class1$ converge vers une bonne performance en source et $dom1$ vers une confusion maximale de l'information de domaine. Nous en déduisons donc que les contraintes de modélisation que DISTGL ajoute à DANN gênent la transférabilité, du moins avec nos choix d'implémentation. La mauvaise performance de DISTGL incite donc à réaliser une série d'études au cours desquelles on interviendra sur un facteur de l'algorithme afin d'étudier son intérêt pour la transférabilité.

Un intérêt de la méthode DISTGL par-rapport aux méthodes sans décodeur est que l'on peut avoir une interprétation visuelle des représentations apprises, ce qui permet de vérifier si la méthode a bien appris à factoriser les concepts, et, dans le cas contraire, d'obtenir un diagnostic visuel de l'échec. Une manipulation intéressante consiste synthétiser des échantillons originaux en croisant le style et le contenu de deux véritables échantillons. Notons d'abord $\Psi_{f_1}(x)$ (resp. $\Psi_{f_2}(x)$) les fonctions d'encodage donnant l'embedding de contenu (resp. de style) d'une image x, voir figure 7.1. Alors, pour deux images x_1, x_2, $\hat{x} = \Phi(\Psi_{f_1}(x_1) \oplus \Psi_{f_2}(x_2))$ devrait être un échantillon synthétique original, ayant le contenu de x_1 et le style de x_2. Cette manipulation permet notamment de transférer le style d'un échantillon du domaine source vers le domaine cible et vice-versa. Dans la figure 7.2, on montre une telle manipulation latente, dans un transfert sur lequel DISTGL fonctionne et dans un cas sur lequel il échoue. Des images de ces manipulations sont disponibles en annexe C pour l'ensemble des transferts.

7.3.2 Intérêt d'un a priori global sur l'espace latent

Dans la méthode DISTGL, deux contraintes de modélisation latente se superposent : 1) nous imposons le désentrelacement classe-domaine sur les deux moitiés du descripteur $f1$ et $f2$ et 2) nous imposons une distribution marginale à-priori $p(z) = \mathcal{N}(0, \mathbf{I}^N)$ sur l'ensemble de l'espace latent

	SO	DANN	DISTGL
M → MT	0.51 ± 0.01	0.52 ± 0.01	0.38 ± 0.01
M → MM	0.67 ± 0.02	0.98 ± 0.01	0.62 ± 0.01
M → CM	0.51 ± 0.03	0.99 ± 0.01	0.96 ± 0.01
M → S	0.36 ± 0.05	0.43 ± 0.05	0.17 ± 0.01
S → M	0.73 ± 0.01	0.80± 0.02	0.32 ± 0.04

TABLE 7.1 – Performance dans le domaine cible de DISTGL par-rapport aux baselines classiques de l'adaptation de domaine ; M=MNIST, MM=MNIST-M, MT=MNIST-T, CM=Color-MNIST, S=SVHN ; Toutes les expériences ont été répétées 3 fois

FIGURE 7.2 – Manipulation du style ; image de gauche=MNIST→Color-MNIST, image de droite=MNIST→SVHN ; Colonne 1=échantillon source, Colonne 2=classe du source, style du cible, Colonne 3=classe du cible, style du source, Colonne 4=échantillon cible

FIGURE 7.3 – Génération non-conditionnelle d'échantillons sur le transfert MNIST→Color-MNIST ; gauche : $\mathcal{L}_{dis}$ imposée (DISTGL) ; droite : $\mathcal{L}_{dis}$ non-imposée (NOGAUSS)

	DISTGL	NOGAUSS
M → MT	0.38 ± 0.01	0.31 ± 0.02
M → MM	0.62 ± 0.01	0.64 ± 0.01
M → CM	0.96 ± 0.01	0.99 ± 0.01
M → S	0.17 ± 0.01	0.12 ± 0.01
S → M	0.38 ± 0.04	0.36 ± 0.01

TABLE 7.2 – Impact du prior gaussien sur la performance dans le domaine cible ; DISTGL=$\mathcal{L}_{dis}$ imposée, NOGAUSS=$\mathcal{L}_{dis}$ non-imposée ; M=MNIST, MM=MNIST-M, MT=MNIST-T, CM=Color-MNIST, S=SVHN

$\mathcal{Z}$. Si cette dernière contrainte est utile pour la génération non-conditionnelle d'échantillons, ce n'est ici pas le but recherché : nous cherchons avant tout à savoir si imposer un tel à-priori robustifie les représentations et donne une meilleure performance dans le domaine cible. Nous proposons donc de comparer DISTGL à une variante dans laquelle la fonction de perte $\mathcal{L}_{dis}$ n'est plus imposée. Nous appelons cette variante NOGAUSS.

Pour confirmer le respect de cette contrainte par l'algorithme DISTGL, nous proposons de générer des échantillons par échantillonnage ancestral : nous générons quelques vecteurs $z \sim \mathcal{N}(0, \mathbf{I}^N)$ que nous décodons à l'aide de Φ. La distribution d'images obtenue devrait alors être l'union des deux domaines $\mathcal{S}_X \cup \mathcal{T}_X$. Dans la figure C.2, nous constatons pour le transfert MNIST→Color-MNIST que les images générées sont bien plus proches des données d'entraînement lorsque la loss $\mathcal{L}_{dis}$ est imposée, ce qui confirme l'impact de ce critère sur la modélisation de l'espace latent. Noter que les échantillons générés par tous les modèles DISTGL sont visibles en annexe C.

Nous comparons enfin les performances de DISTGL et NOGAUSS dans la table 7.2. L'optimisation de $\mathcal{L}_{dis}$ a un impact variable sur la performance selon le transfert considéré : par exemple, elle l'améliore dans le cas MNIST→Color-MNIST mais la dégrade dans le cas MNIST→MNIST-T. Nous ne pouvons donc pas conclure sur l'utilité de ce critère.

Remarque 7.3. *Il est possible que le discriminateur dis n'arrive pas à imposer exactement* $p(z) = \mathcal{N}(0, \mathbf{I}^N)$. *En effet, les vecteurs latents ayant un nombre de dimensions important, il faudrait en théorie entraîner dis sur une quantité immense d'échantillons.*

7.3.3 Influence du décodeur

À l'instar du travail réalisé dans le chapitre 5 sur les méthodes d'alignement de domaine, nous pouvons nous demander si les méthodes de désentrelacement ne sont pas sensibles à des paramètres

	DECONV	SIREN
M → MT	0.38 ± 0.01	0.80 ± 0.03
M → MM	0.62 ± 0.01	0.60 ± 0.01
M → CM	0.96 ± 0.01	0.96 ± 0.01
M → S	0.17 ± 0.01	0.15 ± 0.01
S → M	0.38 ± 0.04	0.25 ± 0.05

TABLE 7.3 – Impact du type de décodeur sur la performance dans le domaine cible ; DECONV=décodeur décovolutif, SIREN=décodeur Siren ; M=MNIST, MM=MNIST-M, MT=MNIST-T, CM=Color-MNIST, S=SVHN

algorithmiques autres que leurs propres objectifs. Nous pouvons par exemple penser à l'architecture d'encodeur ou de décodeur utilisée. Le cas du décodeur est particulièrement intéressant, car ce composant n'est pas présent dans les méthodes d'alignement classiques. Puisque Ψ et Φ sont entraînés conjointement pour satisfaire la reconstruction, le choix de Ψ peut influencer la nature des représentations latentes apprises.

Les mauvais résultats de DISTGL sur MNIST→MNIST-T sont particulièrement intéressants à étudier. En effet, les réseaux convolutifs n'étant pas équivariants aux transformations géométriques telles que la translation, la mise à l'échelle ou la rotation, nous pouvons nous demander si des facteurs latents tels que le degré de translation ou le degré de mise à l'échelle d'un objet sur une image peuvent être découverts naturellement avec un objectif de désentrelacement. Pour répondre à cette question, nous proposons une expérience supplémentaire consistant à changer l'architecture du décodeur Φ en laissant les autres paramètres de DISTGL inchangés.

Récemment, les réseaux « Siren » [Sitzmann et al., 2020] se sont imposés comme une alternative intéressante aux réseaux déconvolutifs. Le principe de ces modèles est très simple : considérons la génération d'une image à partir de sa représentation latente $z \in \mathcal{Z}$: pour générer la valeur RGB du pixel aux coordonnées (i, j) de cette image, on entraîne une fonction de rendu Φ prenant en entrée z, i et j. Pour générer une image entière, il suffit donc d'appliquer la fonction de rendu uniformément sur toutes les valeurs de (i, j) en gardant z constant : on appelle cela une représentation implicite. Φ peut être implémentée par un simple perceptron. Toutefois, pour des raisons d'optimisation pures, il est important de remplacer les fonctions d'activations classiques (ex : ReLU) par des activations sinusoïdales que l'on initialise avec des fréquences variées.

Les Siren pourraient être particulièrement intéressants pour traiter le cas MNIST→MNIST-T, car ils ont la possibilité de modéliser les translations explicitement en modulant le signal d'entrée de i et j avec l'information de z. On peut donc s'attendre à ce que la découverte de facteurs latents liés aux transformations géométriques soit facilitée par le choix d'un tel modèle. Pour confirmer cette hypothèse, on évalue DISTGL avec le décodeur déconvolutif (DECONV) et avec le décodeur Siren (SIREN). Les résultats sont reportés sur la table 7.3.

Sur la plupart des transferts, on constate que l'algorithme entraîné avec un décodeur Siren obtient en moyenne des performances légèrement inférieures qu'avec le décodeur déconvolutif, avec une exception notoire : nous observons un gain de performance significatif dans le transfert MNIST→MNIST-T : le modèle entraîné avec le décodeur Siren dépassant largement DISTGL déconvolutif ainsi que DANN. Ce que nous pouvons confirmer en visualisant les manipulations latentes sur la figure 7.4.

7.4 Conclusion

Les expériences menées au cours de ce chapitre n'ont pas permis de conclure que le désentrelacement classe / domaine apportait un avantage par rapport à une simple contrainte d'alignement. De plus, nous pouvons citer un certain nombre de limites relatives aux méthodes de désentrelacement :

FIGURE 7.4 – Manipulation du style entre MNIST et MNIST-T ; image de gauche=décodeur déconvolutif, image de droite=décodeur Siren ; Colonne 1=échantillon source, Colonne 2=classe du source, style du cible, Colonne 3=classe du cible, style du source, Colonne 4=échantillon cible

l'entraînement est relativement difficile à équilibrer et nécessite un grand nombre d'échantillons pour chaque domaine, de plus, il est connu que les autoencodeurs régularisés tels que le VAE ou l'AAE échouent à modéliser les jeux de données trop compliqués (par exemple ImageNet), ce qui limite ces méthodes à des domaines relativement simples en termes de degrés de liberté.

Cependant, nous avons également pu montrer que dans certains cas, la contrainte de reconstruction pouvait induire un biais favorable à la transférabilité pour peu qu'on utilise le bon modèle de décodeur, ce qui peut constituer une piste d'étude pour de futures recherches. Ce mécanisme de reconstruction permet également de visualiser l'effet concret des manipulations de l'information latente, ce qui est utile pour le diagnostic.

Chapitre 8

Représentations pré-entraînées pour l'adaptation de domaine

Sommaire

Dans ce chapitre, nous poursuivons le travail de recherche d'un biais inductif utile à la transférabilité. Nous choisissons dans ce chapitre de nous concentrer sur la notion de pré-entraînement. En effet, cette approche semble intéressante pour plusieurs raisons que nous allons exposer dans les paragraphes qui suivent.

Dans les chapitres 3, 9 et 6, nous avons pu évaluer une grande variété d'algorithmes d'adaptation de domaine requérant des conditions expérimentales différentes (pour en citer quelques-unes : adaptation de domaine non supervisée, adaptation de domaine semi-supervisée, généralisation de domaine, méta-entraînement). Dans la plupart des cas, ces conditions expérimentales sont assez difficiles à réunir dans un cas pratique réaliste. Pour commencer, les méthodes d'adaptation de domaine non supervisée ont besoin d'une certaine quantité de données du domaine cible, qui, même si elles ne sont pas annotées, peuvent être difficiles à récolter. On peut par ailleurs faire remarquer que si les données cibles sont difficiles à récolter, on en aura qu'un faible nombre et on pourra par conséquent les annoter en peu de temps à l'aide d'un expert humain. Le cas de figure dans lequel on dispose de beaucoup de données dans le domaine cible, mais non annotées, est donc peu fréquent. Mais même lorsque de telles conditions sont réunies, on constate une grande sensibilité des méthodes à certains aspects statistiques du domaine source et du domaine cible. Par exemple, si la proportion des classes n'est pas la même dans le domaine source et le domaine cible, certains algorithmes comme DANN peuvent avoir un comportement instable. Or, il ne faut pas attendre d'un problème réaliste qu'il soit parfaitement équilibré : dans l'industrie, il est courant de construire les jeux de données en rassemblant des données de provenance différente. Les aspects few-shots, déséquilibres et hétérogénéités y sont donc très fréquents. Une mauvaise représentativité de certaines modalités de la distribution d'entraînement peut également rendre impossible tout découpage entraînement/validation/test sophistiqué. Or, les méthodes d'adaptation de domaine étant très instables, il est en pratique indispensable de contrôler leur performance avec un ensemble de validation.

Les approches par méta-apprentissage étudiées chapitre 6 proposent un compromis différent en termes d'exigences expérimentales. En effet, entraîner un modèle de méta-apprentissage nécessite un ensemble de méta-entraînement très difficile à obtenir, car il doit contenir un grand nombre de classes et doit contenir des domaines similaires à l'ensemble de méta-test (nous avons pu étudier plus en détail la capacité de généralisation vers de nouvelles tâches du biais inductif méta-appris). En contrepartie, on obtient un biais inductif relativement puissant permettant de résoudre n'importe quelle tâche impliquant des classes jamais vues avec de bonnes garanties de fonctionnement, puisque la fiabilité du modèle est éprouvée sur un grand nombre de tâches de méta-validation plutôt que sur un seul problème.

On aimerait donc trouver une instance de biais inductif 1) qui soit robuste par-rapport au transfert considéré, 2) qui n'exige pas de données irréalistes pour résoudre ce transfert (par exemple des instances du domaine cible) et 3) qui n'exige pas de données tierces difficiles à obtenir, (par exemple un ensemble de méta-entraînement complètement annoté).

Parmi les biais inductifs déjà étudiés, celui qui se rapproche le plus de ces propriétés est le pré-entraînement des modèles sur la base ImageNet. D'abord, cette méthode apporte un gain certain, peu importe le transfert considéré, pour peu que celui-ci ne soit pas aberrant (c.f. section 5.1). Deuxièmement, on peut tirer un bénéfice du pré-entraînement avec un minimum de données liées au transfert d'intérêt : un simple fine-tuning Source-Only obtient souvent de bonnes performances et ne nécessite par définition que des échantillons source annotés. Enfin, contrairement à ce que l'on pourrait imaginer, le pré-entraînement ne requiert pas de données tierces difficiles à obtenir pour des raisons que l'on expliquera dans le paragraphe suivant.

Le pré-entraînement est désormais largement utilisé dans la littérature récente : en effet, des formes de pré-entraînement de plus en plus avancées ont permis d'apporter des gains considérables en termes de performance et de diminuer fortement le besoin des modèles d'apprentissage profond en annotations. Cependant, ces travaux sont la plupart du temps empiriques et les raisons théoriques expliquant un tel succès demeurent largement méconnues. Le premier pré-entraînement à avoir été utilisé est le pré-entraînement supervisé sur ImageNet. Il consiste simplement à entraîner le

modèle à classer les 1000 classes du jeu de données ImageNet, contenant 1,4 Million d'images environ. La difficulté de ce problème entièrement supervisé encourage l'émergence de descripteurs robustes, qui peuvent ensuite être ajustés pour une nouvelle tâche. Il a été montré par la suite qu'il n'était pas nécessaire d'avoir une tâche supervisée pour le pré-entraînement. Par exemple, un grand nombre de modèles génératifs génèrent implicitement de bons descripteurs après avoir été entraînés à générer ImageNet. La famille de méthodes qui s'est le plus illustrée récemment est la famille des méthodes dites d' « auto-supervision » : le pré-entraînement consiste dans ce cas à résoudre une tâche prétexte ne requérant aucun label pour être synthétisée, et dont on imagine qu'elle fera émerger de bons descripteurs. Par exemple, reconstruire une zone artificiellement masquée de l'image sachant l'information visible oblige le modèle à produire des représentations du contexte visible à tous les niveaux d'abstraction (couleurs, objets, disposition de la scène). Comparées aux méthodes génératives, les méthodes d'auto-supervision ont l'avantage d'être faciles à entraîner puisqu'elles se résument parfois à un simple problème de classification supervisée. La tendance actuelle est à l'exploitation de bases de pré-entraînement de plus en plus massives, mais de moins en moins nettoyées et structurées. Pour suivre ce passage à l'échelle, les modèles augmentent également en taille et en expressivité. Cette massification du pré-entraînement a pour effet de rendre les modèles plus universels et robustes, et donc plus à même de s'adapter facilement vers une nouvelle tâche d'intérêt. Ces bonnes propriétés ont déjà été largement étudiées et mesurées par les auteurs de ces méthodes dans le cadre de la généralisation classique.

Dans ce chapitre, nous proposons de mener la même étude, mais en choisissant comme tâches d'intérêt des scénarios d'adaptation de domaine. Partant d'un modèle pré-entraîné avec une représentation de départ très riche, nous choisissons un protocole de fine-tuning très simple, ayant les mêmes conditions expérimentales que Source-Only pour résoudre chacun de ces transferts.

Le duo pré-entraînement/ protocole de fine-tuning Source-Only définit donc un prototype de méthode qui comme demandé, n'est exigeant ni sur les données du problème d'intérêt, ni sur les données tierces. Dans les parties suivantes, nous chercherons à déterminer au moyen d'une série d'expériences quels choix algorithmiques relatifs à ce prototype de méthode bénéficient le plus à la généralisation inter-domaine.

8.1 Comparaison de différent types de pré-entraînement

Nous commençons l'étude de la chaîne algorithmique combinant successivement pré-entraînement et entraînement source-only par la comparaison de différentes techniques de pré-entraînement proposées dans la littérature. Dans cette série d'expériences, nous faisons varier la tâche de pré-entraînement en laissant les autres paramètres invariants, dans la mesure du possible. Dans toutes les expériences qui suivront, l'ensemble de pré-entraînement sera la base ImageNet-1k. Pour évaluer l'utilité des représentations pré-entraînées pour la généralisation inter-domaine, nous utiliserons le protocole de « sonde linéaire » (ou linear probe), très employé dans la littérature. Ce protocole consiste simplement à entraîner un modèle de classification linéaire sur l'espace des descripteurs à la sortie d'une couche du modèle pré-entraîné que l'on aura choisie au préalable, tout en maintenant les poids de ce dernier figés (autrement dit ils ne sont pas fine-tunés, même sur source). Pour chaque transfert, on entraîne le modèle linéaire sur le domaine source et on l'évalue sur le domaine cible. Ce protocole n'est pas nécessairement optimal, chose qui sera discutée en section 8.3, mais il a l'avantage d'être très rapide à entraîner. De plus, étant donné une architecture pré-entraînée comportant N couches, on peut entraîner simultanément N régressions linéaires sur chacune de ces couches en une seule passe. En effet, comme nous ne pouvons pas savoir à priori à quel endroit du réseau se trouvent les meilleures représentations, il convient de les évaluer toutes. Par conséquent, dans toutes les expériences de ce chapitre, nous fournirons une courbe caractérisant la performance sur le domaine cible en fonction du numéro de couche.

Pré-entraînement supervisé : Le premier type de pré-entraînement que l'on considère est le pré-entraînement supervisé sur les 1000 classes d'ImageNet-1k. On utilise pour cela le pré-entraînement de référence fourni par la bibliothèque PyTorch. Ce pré-entraînement consiste à

résoudre le problème de classification standard à 1000 classes d'ImageNet.

Pré-entraînement par prédiction de rotation : Le second pré-entraînement que l'on considère est l'auto-supervision par prédiction de rotation [Gidaris et al., 2018]. Il s'agit d'une des premières méthodes d'auto-supervision proposées et a l'avantage d'être très simple à implémenter. Cette méthode consiste à montrer au réseau une image d'ImageNet-1k augmentée par une de ces 4 opérations de rotation : 0, 90, 180 ou 270 degrés. Le but du modèle est ensuite de prédire la rotation qui a été appliquée parmi les quatre choix possibles, sous la forme d'un problème de classification supervisée à 4 sorties. Prédire cette rotation oblige le modèle à examiner un certain nombre d'indices visuels de haut niveau, ce qui fait émerger des descripteurs de relativement bonne qualité. Dans cette expérience, nous utilisons l'architecture ResNet-18, initialisée aléatoirement avant le pré-entraînement, que l'on entraîne jusqu'à convergence (on obtient alors une performance de classification de 80% (pour une distribution à priori uniforme sur les 4 rotations).

Pré-entraînement par complétion d'une zone manquante de l'image : Le troisième pré-entraînement que l'on considère est l'auto-supervision par complétion d'une partie manquante de l'image [Pathak et al., 2016]. Cette méthode consiste à montrer au modèle une image d'ImageNet-1k dont on a masqué une zone plus ou moins significative (avec un rectangle noir par exemple). Le modèle s'entraîne ensuite à reconstruire (en espérance) la zone masquée étant donné la zone visible. Intuitivement, pour réaliser une reconstruction correcte, le modèle doit apprendre à analyser les éléments visibles de l'image à tous les niveaux d'abstraction (colorimétrie, mais aussi objets, disposition de la scène) afin de produire la meilleure estimation possible de la partie manquante. Des méthodes de ce type ont déjà été présentées avec des réseaux convolutifs [Pathak et al., 2016] et des vision transformers [He et al., 2021, Bao et al., 2021] et montrent que cet objectif fait émerger des représentations de bonne qualité. En ce qui nous concerne, nous utilisons une architecture U-Net, peu coûteuse à entraîner et assez adaptée à ce genre de problème de prédiction d'image à image. Ce choix permet surtout de garder la comparaison des méthodes de pré-entraînement indépendante de l'architecture considérée : en effet, nous faisons en sorte que l'étage d'encodage du U-Net corresponde exactement à l'architecture ResNet-18. Pour construire le U-Net, nous ajoutons simplement des couches « raccourci » et des couches de décodage, qui ne sont pas utilisées pour produire les descripteurs que l'on étudie. De façon analogue à He et al. [2021], nous choisissons de corrompre image en la découpant en 8×8 secteurs, que l'on masque avec une probabilité $p = 0.75$. En effet, il est important de dégrader 1) une proportion importante des pixels de l'image et 2) de dégrader une zone cohérente spatialement. En effet, ne pas respecter ces règles pourrait rendre la tâche de complétion triviale, par exemple en interpolant les pixels voisins, ce qui n'encouragerait pas la découverte de descripteurs complexes. Nous illustrons les résultats du pré-entraînement dans la figure 8.1.

Pré-entraînement par objectif de contraste : Les méthodes dites de contraste (contrastive representation learning) [Chen et al., 2020b, Grill et al., 2020, Caron et al., 2021] ont récemment connu un grand succès auprès de la communauté de l'apprentissage de représentations, du fait de leur simplicité et de leurs bonnes performances, que ce soit sur ImageNet ou sur une variété de jeux de données de transfert. La première et la plus emblématique de cette famille de méthodes est SimCLR [Chen et al., 2020b], son principe est très simple et fait directement écho à notre discussion sur les invariants de classe de la section 5.1 : il s'agit d'entraîner un extracteur de caractéristiques Ψ à être, pour un jeu de données $\mathcal{D}$ et une famille de transformations aléatoires sur l'espace des images T, 1) équivariant à l'image d'entrée, c'est-à-dire que $\forall x_1, x_2 \sim \mathcal{D}_X$, $\Psi(x_1) \neq \Psi(x_2)$ et 2) invariant à n'importe quelle transformation de T, c'est-à-dire $\forall x \sim \mathcal{D}_X, \forall t_1, t_2 \sim T, \Psi(t_1(x)) = \Psi(t_2(x))$. Concrètement, on choisit T comme étant la famille résultant de la composition de fonctions d'augmentation aléatoires classiques en apprentissage profond (rognage aléatoire, flou, distorsion des couleurs, translation, etc). Pour obtenir une telle propriété d'invariance-équivariance sur les descripteurs, on entraîne le modèle à rapprocher les descripteurs de paires d'images « positives » $t_1(x), t_2(x)$, c'est-à-dire obtenues à partir d'une seule même image de $\mathcal{D}_X$, et en éloignant les descripteurs des paires « négatives » $t_1(x_1), t_2(x_2)$, c'est-à-dire obtenues à partir de deux images distinctes de $\mathcal{D}_X$, nous invitons le lecteur à se référer au papier original pour voir la fonction de

FIGURE 8.1 – Pré-entraînement par complétion ; De droite à gauche : image corrompue, image reconstruite par le U-Net, image d'origine

coût exacte.

Dans notre implémentation, nous utilisons une fois de plus ResNet-18 pour Ψ et suivons les recommandations du papier original en chaînant un rognage aléatoire et une distorsion des couleurs pour synthétiser $t \sim T$. On entraîne Ψ avec une taille de batch 256 jusqu'à convergence, et on obtient alors une performance de classification de 90% (pour 50% de paires positives et 50% de paires négatives à priori).

Initialisation Aléatoire : Afin de montrer l'intérêt du pré-entraînement, on évalue la transférabilité d'un modèle ResNet-18 dont les poids sont laissés à leur valeur initialisée aléatoirement.

Détails généraux d'implémentation : Nous entraînons chaque méthode de pré-entraînement considérée jusqu'à convergence, en veillant à ce qu'elle ne sur-apprenne pas la tâche prétexte par l'intermédiaire d'un découpage train-test sur ImageNet-1k. Nous utilisons par défaut l'architecture ResNet-18 lorsque la tâche de pré-entraînement le permet, avec le pré-traitement des images recommandé pour cette architecture (normalisation et mise à l'échelle).

Lors de l'évaluation des descripteurs, les modèles linéaires sont entraînés sur 3000 itérations de descente de gradient avec l'optimiseur Adam, un pas d'apprentissage de $1e^{-4}$ et une taille de batch 64. Ces hyperparamètres permettent dans tous les cas d'atteindre une performance limite sur source sans que nous observions de sur-apprentissage. Une approche alternative consisterait à utiliser un solveur de régression logistique, mais nous avons pu constater que cela limitait le budget en nombre d'échantillons d'entraînement : les descripteurs sondés sont parfois de très grande dimension, ce qui fait exploser l'empreinte mémoire de tels algorithmes. De plus, nous avons pu constater qu'à nombre d'échantillons égal, le modèle linéaire entraîné par SGD n'était pas nécessairement moins bon que le modèle trouvé par le solveur Scikit-Learn. Dans l'ensemble des expériences, nous centrons-réduisons les descripteurs sondés suivant les statistiques du batch, ce qui a pour effet d'une part d'accélérer l'entraînement des modèles linéaires et d'améliorer la performance dans le domaine cible (notons qu'il s'agit d'une forme d'alignement basique). Sauf mention contraire, dans l'ensemble du chapitre, nous reproduisons chaque expérience 3 fois pour obtenir des écarts-types.

Résultats : Nous présentons les résultats de l'étude comparative des différents pré-entraînements sur les transferts habituels dans la figure 8.2. On remarque tout d'abord que le ResNet-18 initialisé aléatoirement (random) est systématiquement moins bon que les autres, et ce, peu importe la couche considérée, ce qui était plutôt attendu. Il est intéressant de constater que le modèle pré-entraîné de manière supervisée (supervised) partage d'une part des performances similaires à un de ses homologues non supervisés au moins (rotation, complétion ou simclr) sur les jeux de données de reconnaissance de chiffres, mais qu'il les bat largement d'autre part sur les transferts PACS (voir les 3 derniers graphes). Il est possible que cela découle du fait que PACS contient des classes déjà présentes dans ImageNet, ce qui avantage largement le pré-entraînement supervisé. Nous ferons remarquer par-ailleurs que ce biais pose un problème supplémentaire dans l'évaluation des méthodes d'adaptation de domaine existantes. En ce qui concerne les trois modèles auto-supervisés rotation, complétion et simclr, on peut affirmer que SimCLR est dans l'ensemble moins bon que les deux autres, ce qui est surprenant dans la mesure où SimCLR est explicitement entraîné à être invariant à des transformations qui pourraient constituer des écarts de domaine. Du reste, entre rotation et completion, la hiérarchie est moins claire et dépend largement du transfert que l'on considère. De manière générale, on peut constater que la variance du sondage linéaire est relativement faible et que pour les modèles auto-supervisés, ce sont les couches du milieu qui apportent la meilleure performance, une tendance déjà observée dans Chen et al. [2020a] avec les transformers. Enfin, on peut remarquer que le transfert MNIST→SVHN, que nous avons présenté comme extrêmement difficile tout au long de ce manuscrit, est tout de même résolu avec une performance de 53% par une sonde linéaire sur le pré-entraînement supervisé, pour peu qu'on l'applique à l'avant-dernière couche (située avant le Global Average Pooling).

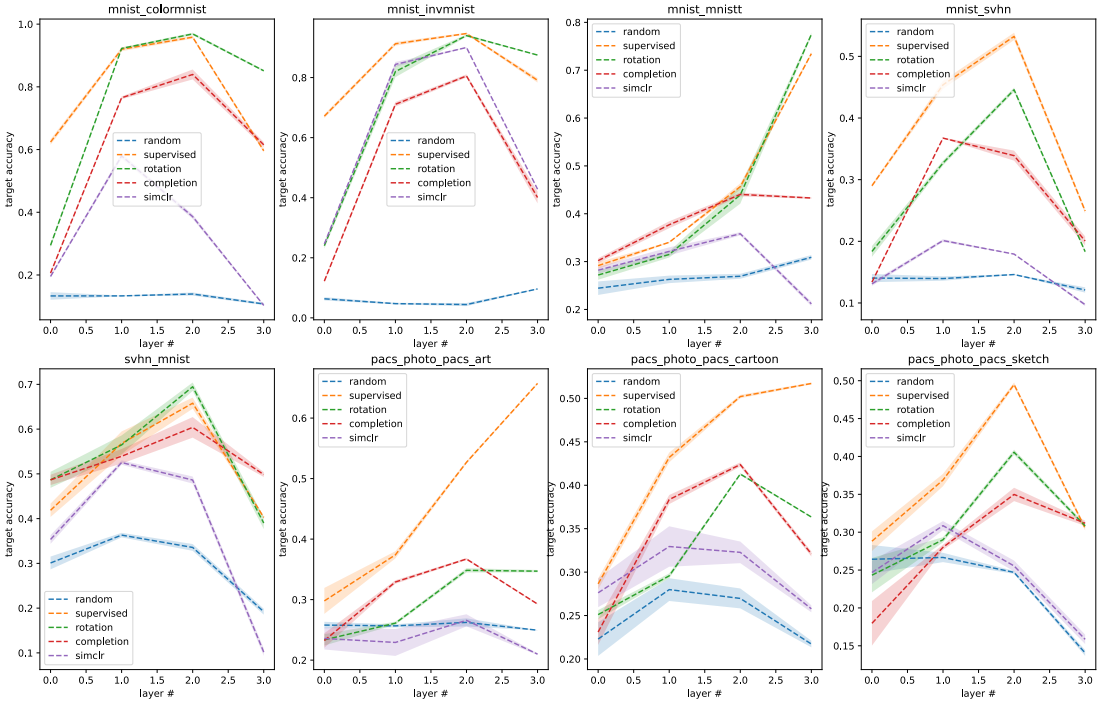

FIGURE 8.2 – Performance du linear probe en fonction du numéro de couche de l'architecture ResNet-18 pré-entraînée de différentes manières ; Pointillés=performance moyenne, polygones transparents=zone à l'intérieur des écarts-types

8.2 Comparaison de différentes architectures

Dans cette expérience, on compare trois architectures, avec des capacités différentes, entraînées avec une méthode de pré-entraînement similaire. On choisit comme référence le pré-entraînement supervisé sur ImageNet-1k, d'une part parce qu'on a pu voir dans la section 8.1 qu'il était compétitif lorsque confronté aux autres approches, et d'autre part parce que des poids pré-entraînés de bonne qualité sont fournis immédiatement par la bibliothèque PyTorch pour un grand nombre d'architectures.

On propose donc de comparer les poids pré-entraînés de la même manière d'architectures de taille variable : ResNet-18, ResNet-50 et ResNet-101 [He et al., 2015].

Résultats : On résume les résultats dans la figure 8.3. En moyenne, sur l'ensemble des transferts, on ne constate pas d'accroissement de la performance avec la capacité de l'architecture. Ce résultat est intéressant dans la mesure où, dans le cadre de la généralisation classique, les modèles de plus grande capacité ont généralement de meilleures performances [He et al., 2015].

8.3 Sonde linéaire vs fine-tuning

Dans cette série d'expériences, on évalue cette fois plusieurs protocoles permettant d'exploiter le modèle pré-entraîné au problème d'adaptation de domaine subséquent. Jusqu'à présent, on a uniquement exploité les descripteurs obtenus à la sortie de chaque couche en entraînant une couche linéaire sans ajuster les poids pré-entraînés des couches précédentes (linear-probe). Or, il est courant dans la littérature d'entraîner conjointement cette couche linéaire avec le reste des poids (fine-tuning) pour obtenir une meilleure performance, on a d'ailleurs pu observer section 5.2.3 que cela améliorait généralement l'utilité des descripteurs pour la tâche d'intérêt.

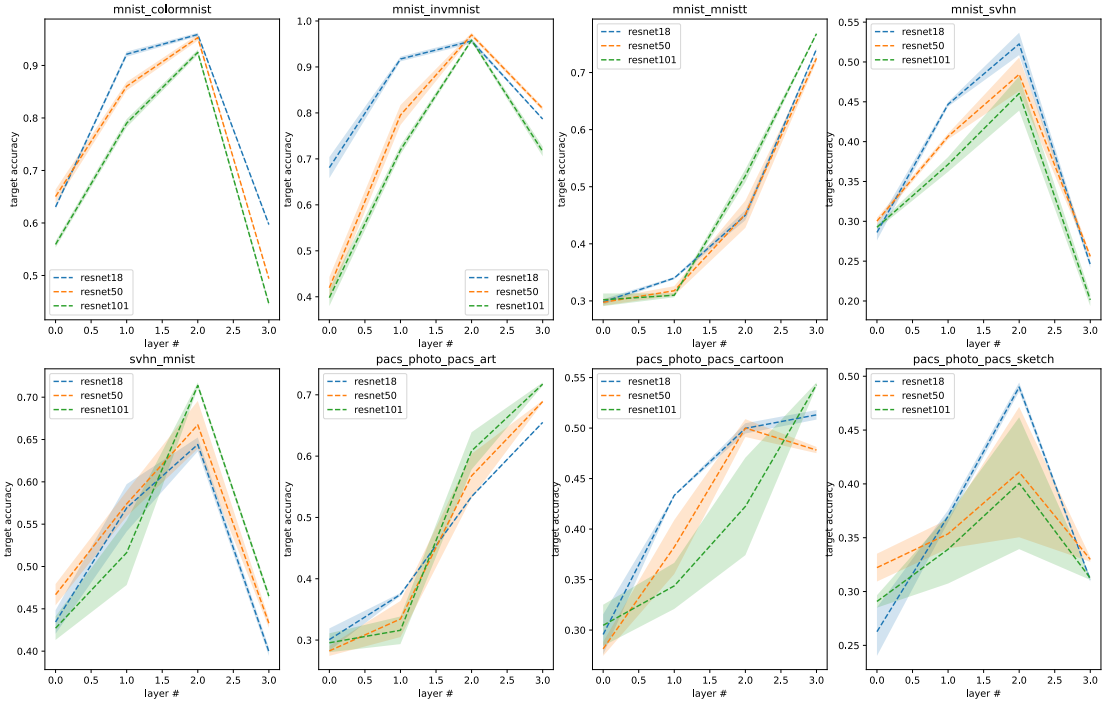

FIGURE 8.3 – Performance du linear probe en fonction du numéro de couche pour différents backbones pré-entraînés de la même façon ; Pointillés=performance moyenne, polygones transparents=zone à l'intérieur des écarts-types

On réalise cette comparaison sur la base du ResNet-18 pré-entraîné par classification supervisée sur ImageNet-1k. Dans le cas du linear probe, on conserve tous les choix d'implémentation des expériences précédentes. Dans le cas du fine-tuning, on maintient 3000 itérations avec l'optimiseur Adam, dans la mesure où ce choix ne semble pas sur-apprendre sur le domaine source, y compris sur les jeux de données low-shot tels que PACS. Contrairement aux sondes linéaires, que l'on peut entraîner sur toutes les couches simultanément en une seule passe à travers l'extracteur de caractéristiques, le fine-tuning nous oblige à réaliser un entraînement séparé pour chaque couche, en réinitialisant l'extracteur à son point de départ pré-entraîné entre chaque entraînement.

Résultats : On reporte les résultats sur la figure 8.4. Bien que l'on vérifie qu'il n'y ait pas de sur-apprentissage dans le domaine source, l'effet du fine-tuning est relativement instable en ce qui concerne la performance du domaine cible. D'abord, on constate systématiquement une variance accrue par-rapport au protocole de sonde linéaire. En ce qui concerne la performance moyenne, on observe systématiquement une amélioration sur les descripteurs de la couche 4 (dernier point de la courbe, descripteurs obtenus juste avant les logits de classification), ce qui confirme les observations faites dans le chapitre 5. Par contre, dans les couches intermédiaires du réseau, l'effet du fine-tuning est le plus souvent délétère pour la qualité des représentations.

8.4 Borne théorique appliquée à la sonde linéaire

Dans cette dernière section, on propose de réexaminer le cas de la classification linéaire sur des descripteurs pré-entraînés fixes à la lumière d'une borne de l'adaptation de domaine proposée récemment par Zhang et al. [2019]. Noter que cette borne a déjà été présentée en détail dans le chapitre 4. On la rappelle néanmoins ci-dessous.

$$\forall h \in \mathcal{H}, R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) + \gamma \tag{8.1}$$

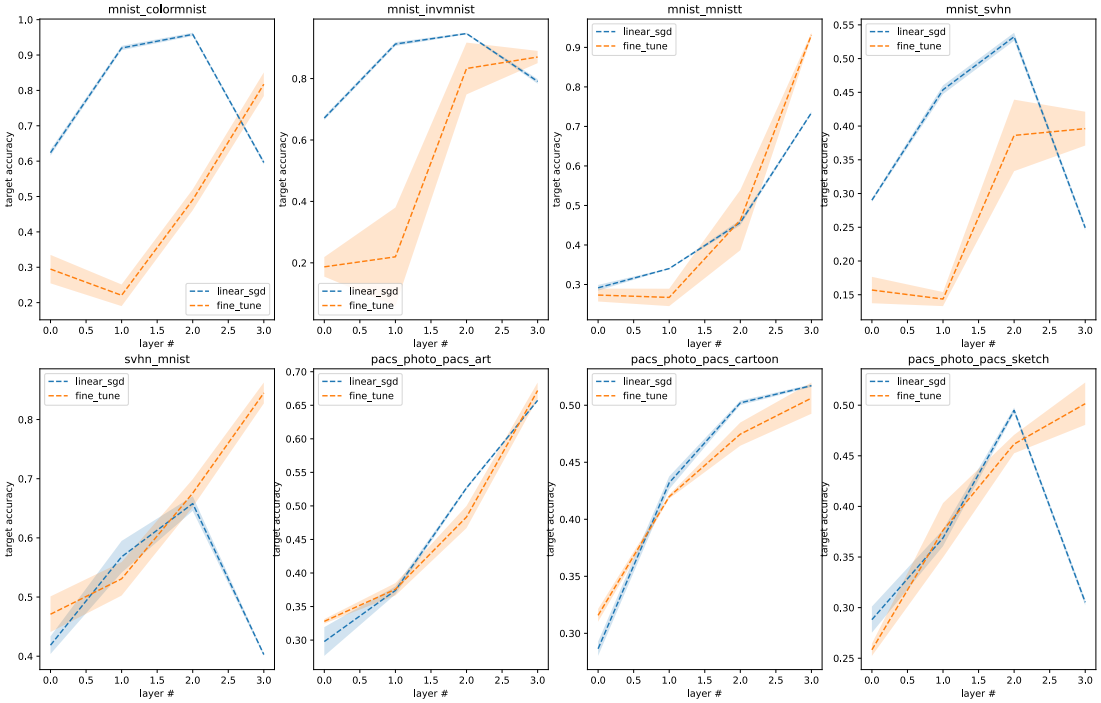

FIGURE 8.4 – Performance du linear probe et du fine-tuning en fonction du numéro de couche pour le ResNet-18 pré-entraîné sur ImageNet-1k ; Pointillés=performance moyenne, polygones transparents=zone à l'intérieur des écarts-types

Avec $\gamma = \min_{h^\star} R_{\mathcal{S}}(h^\star) + R_{\mathcal{T}}(h^\star)$ et $d_{h,\mathcal{H}}(\mathcal{S},\mathcal{T}) = \sup_{h' \in \mathcal{H}} \underset{\mathcal{T}}{\mathbf{Pr}}(h' \neq h) - \underset{\mathcal{S}}{\mathbf{Pr}}(h' \neq h)$

Dans les cas de figure que l'on a pu étudier précédemment, on a pu montrer que la borne ne pouvait pas être minimisée correctement : soit on raisonne sur l'espace d'entrée, c'est-à-dire $\mathcal{S}_X$ et $\mathcal{T}_X$ fixés et $\mathcal{H}$ expressive (pour garantir $R_{\mathcal{S}}(h)$ et γ faibles), mais dans ce cas $d_{h,\mathcal{H}}(\mathcal{S}_X,\mathcal{T}_X)$ ne peut pas être minimisé, soit on raisonne sur un espace de descripteurs produits par un encodeur Ψ, et dans ce cas $R_{\mathcal{S}}(h)$ et $d_{h,\mathcal{H}}(\mathcal{S}_X,\mathcal{T}_X)$ peuvent être contrôlés, mais pas γ. Rappelons que l'une des propriétés intéressantes de cette borne est sa dépendance à une famille d'hypothèses $\mathcal{H}$ plus ou moins riche. Or, on peut remarquer d'une part que la famille $\mathcal{H}$ des modèles de classification linéaire peut être considérée comme relativement faible, et d'autre part que les descripteurs pré-entraînés permettent à de tels modèles d'obtenir une performance raisonnable. On se trouve donc dans un cas de figure potentiellement idéal pour exploiter les propriétés de la borne, puisque $\mathcal{S}_X$ et $\mathcal{T}_X$ sont fixés, ce qui avec h linéaire garantit γ et $R_{\mathcal{S}}(h)$ raisonnablement bas. Enfin, puisque $\mathcal{H}$ est relativement peu expressive, on peut supposer que $d_{h,\mathcal{H}}(\mathcal{S}_X,\mathcal{T}_X)$ sera également plus faible.

On propose de vérifier cette hypothèse en évaluant les trois termes de la borne appliquée au cas d'un classifieur linéaire entraîné sur les espaces de descripteurs de ResNet-18 pré-entraîné. $R_{\mathcal{S}}(h)$ est donné immédiatement par la performance de h sur $\mathcal{S}$. γ est obtenu en entraînant $h^\star$ linéaire sur l'union des deux domaines, l'évaluation de ce terme nécessite bien évidemment les annotations du domaine cible, que l'on utilise ici à des fins de diagnostic.

L'estimation de $d_{h,\mathcal{H}}(\mathcal{S}_X,\mathcal{T}_X)$ est plus difficile : en supposant h fixé, il faut entraîner h' maximisant l'accord avec h sur $\mathcal{S}$ et maximisant le désaccord sur $\mathcal{T}$. Le principal problème est que cette quantité n'est pas dérivable par-rapport aux paramètres de h', puisqu'elle est basée sur la loss 0-1. On est donc obligés d'utiliser une loss « proxy » dérivable qui soit la plus proche possible du vrai critère.

Soit N le nombre total de classes, h une fonction d'hypothèse fixée et $\tilde{h}$ le modèle probabiliste

sous-jacent tel que $\tilde{h}(x)$ produit une distribution $\{p_i^h(x)\}_{1..N}$ avec $h(x) = \text{argmax}(\{p_i^h(x)\})$. On considère également deux échantillons $x_s \sim \mathcal{S}_X$ et $x_t \sim \mathcal{T}_X$. On propose l'algorithme suivant pour minimiser $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$. On commence par considérer $h(x_s)$ et $h(x_t)$ les labels catégoriques produits par h. La probabilité que le modèle probabiliste $\tilde{h}'$ soit d'accord avec le modèle catégorique h sur x_s est $p^{h'}_{h(x_s)}(x_s)$, et la probabilité qu'ils soient d'accord sur x_t est $p^{h'}_{h(x_t)}(x_t)$. En passant à l'espérance, estimer $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$ revient à maximiser :

$$\max_{h'} \mathop{\mathbb{E}}_{x_s \sim \mathcal{S}_X} \left[p^{h'}_{h(x_s)}(x_s) \right] - \mathop{\mathbb{E}}_{x_t \sim \mathcal{T}_X} \left[p^{h'}_{h(x_t)}(x_t) \right] \tag{8.2}$$

Zhang et al. [2019] proposent plutôt d'optimiser un critère basé sur les log-probabilités

$$\max_{h'} \mathop{\mathbb{E}}_{x_s \sim \mathcal{S}_X} \left[\log(p^{h'}_{h(x_s)}(x_s)) \right] + \mathop{\mathbb{E}}_{x_t \sim \mathcal{T}_X} \left[\log(1 - p^{h'}_{h(x_t)}(x_t)) \right] \tag{8.3}$$

Il peut sembler naturel de vouloir optimiser une quantité basée sur les log-probabilités, puisqu'il s'agit d'une pratique courante en machine learning connue pour stabiliser l'optimisation des modèles de classification. Cependant, dans ce cas précis, la présence ou non de log modifie le compromis devant être trouvé par h'. L'objectif 8.2 est plus proche de l'estimation de la partie optimisable de $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$, puisque dans ce cas, la seule approximation réside dans la nature probabiliste du $\tilde{h}'$ optimisé. Cet objectif devrait donc, en principe, trouver un meilleur optimum h' par-rapport au critère réel $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$. Pour valider cette hypothèse, on propose une première expérience consistant à comparer, pour h fixé, le h' obtenu en optimisant 8.2 ou 8.3 en évaluant la valeur réelle de $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$ à posteriori.

Remarque 8.1. *Aucune de ces deux fonctions de coût n'est à priori la fonction de coût optimale. On gardera donc à l'esprit que dans les deux cas les h' obtenus donnent une quantité qui sous-estime la vraie valeur de $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$.*

On propose de réaliser l'évaluation des termes de la borne sur quelques transferts habituels, pour un h entraîné sur chaque couche du ResNet-18 pré-entraîné. Pour chaque transfert, on comparera l'impact du choix de l'objectif heuristique sur l'estimation de $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$. Pour tracer ces caractéristiques, on procède en deux étapes : on entraîne d'abord en source-only sur le transfert considéré les h linéaires correspondant à chaque couche, exactement comme dans les expériences précédentes. Les h obtenus donnent directement $R_{\mathcal{S}}(h)$ pour chaque couche. On entraîne ensuite, pour ces h fixés, l'ensemble des h' permettant d'estimer le terme de divergence pour chaque couche, en choisissant un des deux objectifs heuristiques. Enfin, on entraîne l'ensemble des $h^\star$ sur l'union des deux domaines pour obtenir la valeur de γ à chaque couche. On trace les caractéristiques obtenues dans la figure 8.5.

Pour ce qui est de l'estimation du terme de divergence, le constat est sans appel : l'objectif 8.2 (sans les logs) permet d'optimiser beaucoup mieux la véritable quantité que l'objectif 8.3. En effet, lorsque ce dernier est utilisé, la borne ne majore même pas toujours $R_{\mathcal{T}}(h)$. On concentre donc notre commentaire sur les estimations faites avec le meilleur des deux critères. On remarque que lorsque h est entraîné en source-only, les bornes estimées majorent grossièrement $R_{\mathcal{T}}(h)$. On en déduit que si une borne faible implique $R_{\mathcal{T}}(h)$ faible, la réciproque, elle, est fausse.

On propose une seconde expérience qui consiste à entraîner h linéaire à minimiser $R_{\mathcal{S}}(h) + d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$ plutôt que $R_{\mathcal{S}}(h)$. Cela implique que contrairement à tout ce qui a été fait précédemment dans le chapitre, on quitte le cadre de la Domain Generalization pour revenir à celui de l'UDA. Pour mener à bien cette minimisation, au lieu d'entraîner successivement h puis h', on va les entraîner alternativement à la manière de l'apprentissage adversaire. En principe, on aimerait que h' et h maximisent et minimisent tour à tour l'objectif heuristique 8.2. Ce n'est pas faisable en pratique puisque cet objectif est dérivable par-rapport à h' mais pas par-rapport à h. Pour l'entraînement de h, on propose donc un objectif dual où h' produit les labels catégoriques et $\tilde{h}$ s'y adapte. Enfin, dans cette expérience, on ne minimise pas $R_{\mathcal{S}}(h)$ par l'intermédiaire d'un coût d'entropie croisée : on choisit plutôt de maximiser $\mathop{\mathbb{E}}_{x_s \sim \mathcal{S}_X} \left[p^h_{y_s}(x_s) \right]$. En effet, l'entropie croisée est

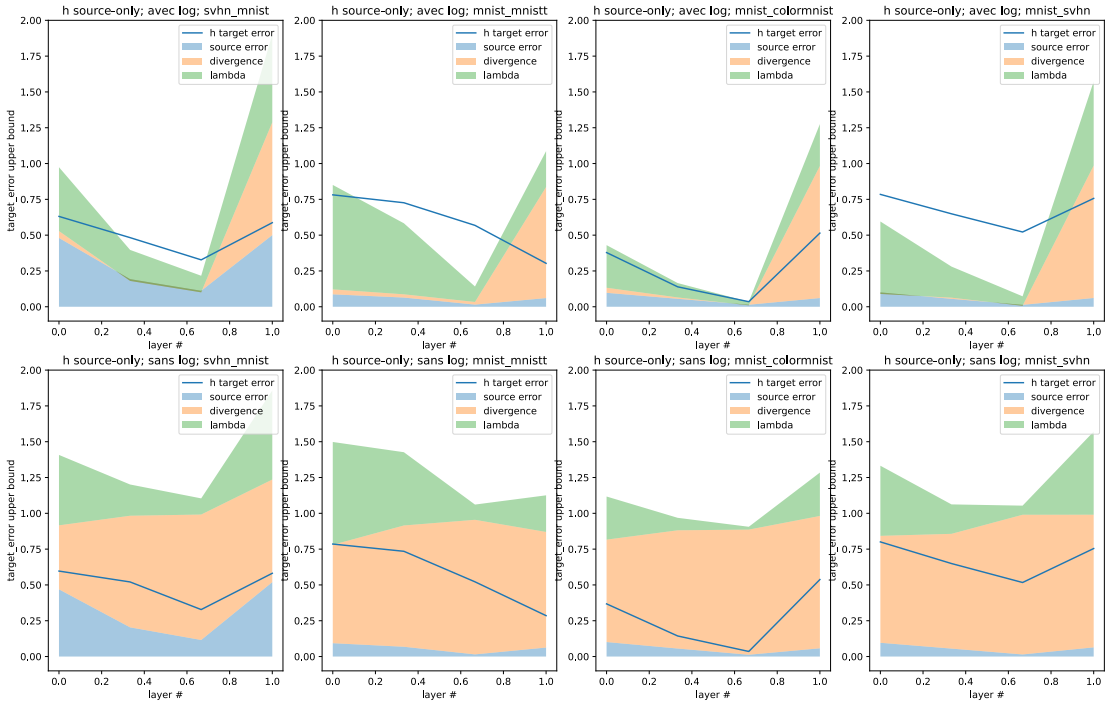

FIGURE 8.5 – Evaluation de la borne en fonction de la couche du modèle utilisée ; La valeur de la borne est le cumul des trois termes suivants : bleu=erreur source, orange=divergence, vert=erreur jointe ; La courbe bleue indique l'erreur de h dans le domaine cible, qui est la quantité majorée par la borne ; On compare par-ailleurs les deux objectifs heuristiques dans l'évaluation du terme de divergence ; $1^{ère}$ ligne=avec *log*, $2^{ème}$ ligne=sans *log*

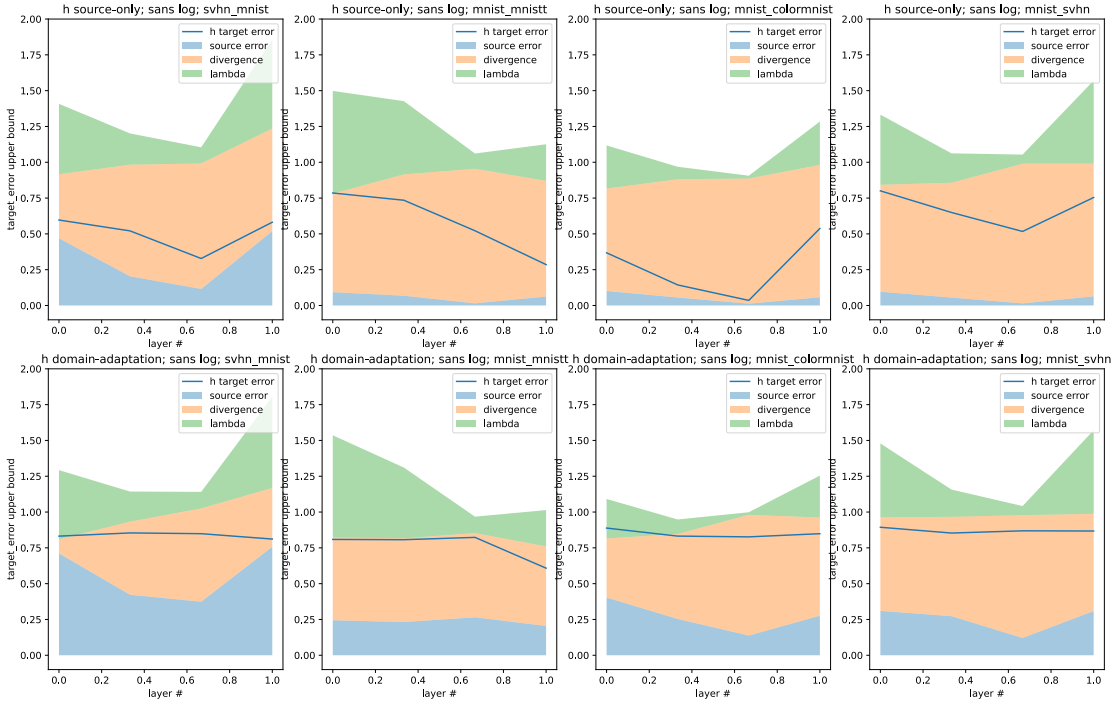

FIGURE 8.6 – Evaluation de la borne en fonction de la couche du modèle utilisée ; La valeur de la borne est le cumul des trois termes suivants : bleu=erreur source, orange=divergence, vert=erreur jointe ; La courbe bleue indique l'erreur de h dans le domaine cible, qui est la quantité majorée par la borne ; On compare par-ailleurs les deux objectifs heuristiques dans l'évaluation du terme de divergence ; $1^{ère}$ ligne=avec *log*, $2^{ème}$ ligne=sans *log*

basée sur les log-probabilités, ce qui pourrait modifier le compromis entre la minimisation de $R_{\mathcal{S}}(h)$ et celle de $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$. En prenant en compte ces remarques, l'algorithme se résume donc à :

$$\max_{h'} \mathop{\mathbb{E}}_{x_s \sim \mathcal{S}_X} \left[p^{h'}_{h(x_s)}(x_s) \right] - \mathop{\mathbb{E}}_{x_t \sim \mathcal{T}_X} \left[p^{h'}_{h(x_t)}(x_t) \right] \tag{8.4}$$

$$\max_{h} \mathop{\mathbb{E}}_{x_s \sim \mathcal{S}_X} \left[p^{h}_{y_s}(x_s) \right] - \mathop{\mathbb{E}}_{x_s \sim \mathcal{S}_X} \left[p^{h}_{h'(x_s)}(x_s) \right] + \mathop{\mathbb{E}}_{x_t \sim \mathcal{T}_X} \left[p^{h}_{h'(x_t)}(x_t) \right] \tag{8.5}$$

Afin de veiller à ce que h' soit à l'optimum à chaque pas de minimisation de h, on veille à entraîner h' 5 fois pour chaque minimisation de h. On réalise cette optimisation alternée pour chaque transfert et chaque couche du réseau pré-entraîné. Noter que l'estimation de γ ne change pas dans cette nouvelle expérience, puisqu'elle découle d'un entraînement indépendant. On appelle cet algorithme de minimisation. On affiche les nouvelles caractéristiques dans la figure 8.6.

Les résultats obtenus sont plutôt intrigants : on constate que par rapport à source-only, la méthode d'adaptation de domaine minimise davantage $d_{h,\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X)$, mais au détriment de $R_{\mathcal{S}}(h)$. La somme des deux reste dans l'ensemble totalement inchangée. Il est donc possible que la borne, une fois de plus, ne puisse être minimisée vers une valeur intéressante. On constate par ailleurs que le h obtenu par adaptation de domaine a une erreur bien plus élevée sur le domaine cible $R_{\mathcal{T}}(h)$ que le h entraîné en source-only.

8.5 Conclusion

Dans ce chapitre, nous avons pu étudier en profondeur l'importance du pré-entraînement pour l'adaptation de domaine, en particulier dans le cadre de la généralisation de domaine.

Nous avons commencé par comparer différents types de pré-entraînement à architecture égale : il en est ressorti que dans nos expériences, le pré-entraînement supervisé était celui qui produisait les meilleurs descripteurs pour la généralisation de domaine. Cependant, les pré-entraînements non-supervisés demeurent compétitifs dans de nombreux transferts et pourraient être désavantagés dans la mesure où il existe des classes similaires entre ImageNet et nos jeux de données de test. On notera enfin qu'un travail présentant de nombreuses similarités avec le nôtre a pu être présenté très récemment [Kim et al., 2022], les auteurs montrent notamment que pour un backbone et un pré-entraînement moderne, des méthodes d'adaptation de domaine relativement anciennes, par exemple DANN, peuvent battre des méthodes récentes.

Nous avons pu mesurer l'intérêt du fine-tuning par-rapport à la sonde linéaire : la principale conclusion de cette étude est que l'entraînement d'un modèle linéaire donne des résultats avec une variance plus faible.

Nous sommes conscients des limites de ces études, dans la mesure où elles pourraient être largement renforcées en utilisant une implémentation de référence pour chaque pré-entraînement (ce qui dans notre cas contrevenait au fait d'évaluer le pré-entraînement *toutes choses égales par-ailleurs*) et en multipliant le nombre de backbones évalués.

Enfin, nous avons pu évaluer une borne d'adaptation de domaine récente dans le cadre de la régression linéaire sur des descripteurs pré-entraînés. Un cas de figure qui aurait pu être intéressant dans la mesure où les hypothèses h considérées sont de faible capacité. Il en ressort que même dans ce cas, la borne étudiée majore grossièrement le risque du domaine cible et semble difficile à minimiser en pratique.

Chapitre 9

Adaptation de domaine semi-supervisée

Sommaire

Ce chapitre résume les conclusions d'une étude débutée, mais non-aboutie, portant sur l'adaptation de domaine semi-supervisée. En effet, nous n'avons pas confronté nos résultats à des méthodes état-de-l'art de la littérature [Saito et al., 2019, Yoon et al., 2022, Kim and Kim, 2020], car ces dernières comportent de nombreux mécanismes difficiles à ajuster correctement lorsque l'on souhaite les réimplémenter. Par conséquent, nous limiterons notre étude à une méthode d'adaptation semi-supervisée relativement simple.

Nous avons pu constater dans les chapitres 3 et 4 que le succès des méthodes d'alignement de domaine n'était pas garanti, que ce soit en théorie ou en pratique. D'une part, d'après les bornes théoriques proposées pour l'adaptation de domaine, minimiser l'erreur de classification dans le domaine source et aligner les descripteurs source et cible n'est pas suffisant pour assurer une bonne performance dans le domaine cible. En effet, nous avons expliqué dans la section 4.3 qu'il est possible de tomber dans le cas pathologique où les distributions marginales source et cible sur $\mathcal{Z}$ sont parfaitement alignées, mais où les distributions jointes source et cible sur $\mathcal{Z} \times \mathcal{Y}$ ne sont pas alignées (c.f. figure 4.7). D'autre part, cette insuffisance théorique a été confirmée empiriquement dans le chapitre 3, puisque nous constatons une mauvaise performance des méthodes d'alignement sur un grand nombre de transferts.

Nous pouvons nous demander s'il ne serait pas possible d'éviter ces cas d'alignement pathologique, moyennant un effort d'annotation minime dans le domaine cible, c'est-à-dire en prenant la peine d'annoter un faible nombre d'exemples du domaine cible de chaque classe. En effet, nous pouvons émettre l'hypothèse que ces quelques annotations permettront d'identifier implicitement les clusters d'échantillons de chaque classe au sein du domaine cible et guider l'alignement vers une solution non dégénérée. Tout au long de ce court chapitre, nous nous placerons donc dans le cadre de l'adaptation de domaine semi-supervisée (SSDA).

9.1 Influence de l'annotation dans le domaine cible

Nous définissons donc une méthode à trois objectifs : 1) objectif de classification supervisée sur source, 2) objectif de classification supervisée sur les quelques annotations du domaine cible et 3) objectif d'alignement des distributions latentes en tirant parti du reste du domaine cible non annoté.

Cependant, pour mesurer avec rigueur l'intérêt de ces conditions expérimentales légèrement plus avantageuses, la méthode d'adaptation de domaine semi-supervisée doit être confrontée à une grande variété de baselines supplémentaires, qui doivent, elles aussi, bénéficier de ces annotations supplémentaires pour une comparaison juste :

— **Source-Only :** Classification supervisée sur le domaine source uniquement

— **Target-Only :** Classification supervisée sur les quelques échantillons supervisés du domaine cible uniquement (ce qui en fait un entraînement few-shot/low-shot)

— **Source+Target :** Classification supervisée sur la concaténation du domaine source et des quelques exemples annotés du domaine cible

— **SSDA :** Classification supervisée sur la concaténation du domaine source et des quelques exemples annotés du domaine cible que l'on associe à un objectif d'alignement non supervisé (qui fait intervenir les exemples non annotés du domaine cible).

On résume la liste des méthodes à comparer dans la table 9.1 en fonction des objectifs minimisés.

Il convient également de mesurer le gain obtenu en fonction du nombre d'annotations fournies pour chaque classe dans le domaine cible. Dans nos expériences, nous tracerons donc une caractéristique de performance en fonction du nombre de shots pour les méthodes dépendant de ce paramètre.

Méthode	Sup. source	Sup. (low-shot) cible	Alignement
Source-Only (SO)	✓		
Target-Only (TO)		✓	
Source + Target (ST)	✓	✓	
Unsupervised Domain Adaptation (UDA)	✓		✓
Semi-Supervised Domain Adaptation (SSDA)	✓	✓	✓

TABLE 9.1 – Ensemble des baselines à comparer dans le cas semi-supervisé, ici résumées comme une combinaison particulière de trois fonctions de coût.

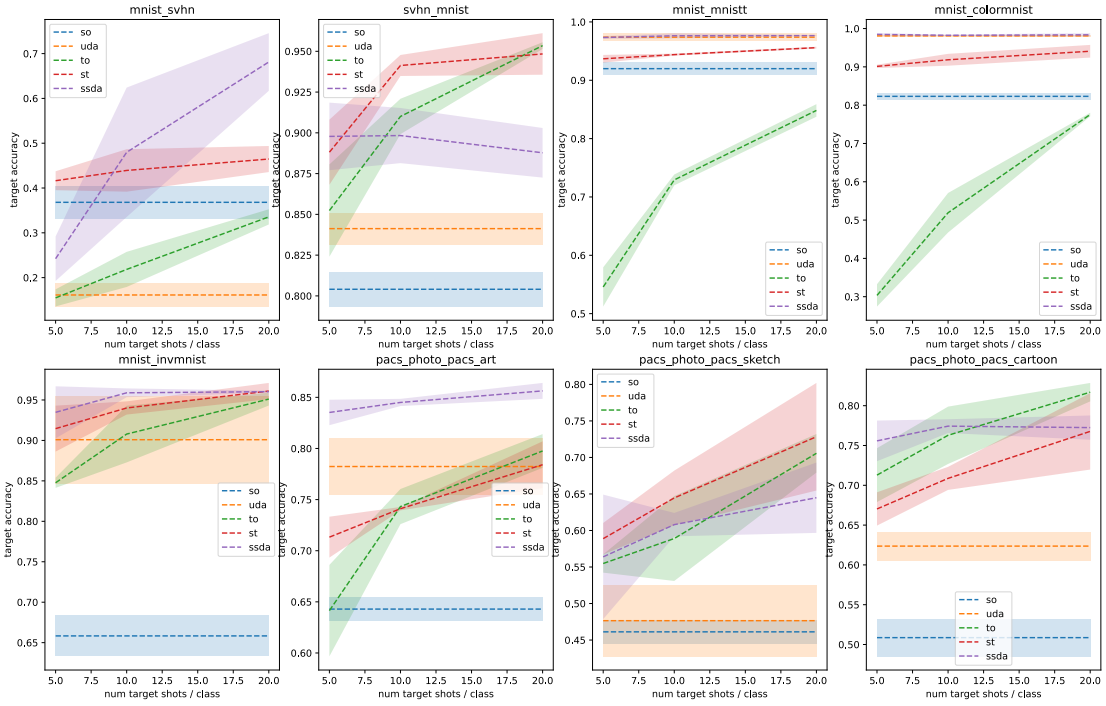

FIGURE 9.1 – Caractéristiques de performance de chaque méthode en fonction du nombre d'annotations par classe en cible ; SO=Source-Only, UDA=DANN, TO=Target-Only, ST=Source+Target, SSDA=Semi-Supervised Domain adaptation ; Pointillés=performance moyenne, polygones transparents=zone à l'intérieur des écarts-types

9.2 Expériences

Jeux de données : Nous évaluons l'ensemble des méthodes mentionnées dans la section précédente sur une batterie de transferts de domaine aux propriétés variées. On construit ces transferts à partir des jeux de données suivants. D'abord le jeu de données PACS, qui contient 7 classes d'objets et 4 domaines relativement différents (photo, sketch, cartoon et painting). Ensuite, les jeux de données usuels de chiffres MNIST et SVHN. Enfin, nous proposons également un ensemble de variantes de MNIST, mettant chacune l'accent sur une transformation difficile de l'espace des images : COLOR-MNIST présente des augmentations de colorisation, MNIST-T des transformations géométriques et MNIST-INV une inversion des couleurs. L'ensemble de ces jeux de données et variantes sont présentés en détail, avec des exemples visuels dans l'annexe A.

Nous veillons évidemment à ce que domaine cible soit bien découpé en sous-ensembles entraînement/test, avec l'ensemble des exemples annotés dans l'ensemble d'entraînement.

Détails d'implémentation : Nous utilisons comme extracteur Ψ le ResNet-18 pré-entraîné. Comme constaté dans le chapitre 5, le pré-entraînement apporte un gain significatif en termes de performance et est en pratique toujours utilisé. De plus, il permet d'avantager au maximum la baseline Target-Only, entraînée sur très peu d'exemples. Du reste, les classifieurs et discriminateurs ont la même architecture que dans le chapitre 3, intrègrent également la normalisation spectrale et sont entraînés avec la même cadence. Chaque expérience est cette fois entraînée pendant 2000 itérations.

Nous évaluons les méthodes dépendant de la supervision en cible selon trois cas de densité d'annotation : 5 shots/classe, 10 shots/classe et 20 shots/classe. Pour chaque cas de figure, nous relançons l'expérience 3 fois pour calculer les écarts-types. Pour chaque transfert et chaque méthode, on trace donc une caractéristique de performance en fonction du nombre de shots. Les résultats sont présentés dans la figure 9.1.

Résultats : À partir des courbes de la figure, nous pouvons établir un certain nombre de faits dont la véracité est robuste au transfert considéré. Sans surprise, les méthodes supervisées sur cible ont une performance qui va croissant avec le nombre d'annotations fournies. En moyenne, l'alignement non supervisé (UDA) semble obtenir de meilleures performances que l'entraînement source-only (SO), la seule exception étant MNIST→SVHN, qui est particulièrement difficile et mal conditionné. De plus, nous observons que l'alignement semi-supervisé (SSDA) bat l'alignement non supervisé de manière consistante.

Du reste, le comportement de l'adaptation de domaine semi-supervisée semble très dépendant du transfert considéré. Par exemple, il est difficile d'établir une hiérarchie entre les méthodes supervisées TO, ST et SSDA : SSDA, que l'on imagine être la plus sophistiquée des trois, se fait pourtant battre par TO et ST dans un nombre important de cas de figures. Or, TO et ST sont bien moins exigeantes en termes de conditions expérimentales que SSDA, puisqu'elles ne nécessitent pas un grand nombre d'exemples dans le domaine cible. Plus troublant, dans le transfert PACS_photo → PACS_cartoon, TO bat ST (voire SSDA avec 20 shots). Cela signifie que dans certains cas, le domaine source a une influence purement négative lorsqu'il est ajouté aux quelques annotations du domaine cible.

9.3 Conclusion

L'adaptation de domaine semi-supervisée, dont nous avons évalué uniquement une implémentation simple (SSDA) dans ce chapitre, semble avoir un impact globalement positif sur la performance lorsqu'elle est comparée à sa contrepartie non supervisée (UDA). Cependant, elle peut, dans certains cas, largement se faire battre par des baselines naïves exploitant le même supplément de supervision, mais dépourvues de mécanisme d'alignement (TO et ST). Cela pose la question de l'utilité réelle de l'adaptation de domaine semi-supervisée, du moins dans son implémentation la plus version naïve, étant donné que d'une part ces baselines sont plus simples et moins exigeantes en données que SSDA, et que d'autre part la performance relative entre TO, ST et SSDA semble extrêmement

dépendante du transfert considéré et ne peut pas être connue à l'avance. Enfin, rappelons que ces conclusions doivent du reste être renforcées par la confrontation à des méthodes semi-supervisées état-de-l'art.

Chapitre 10

Conclusions et perspectives

Sommaire

10.1 Rappel des principaux résultats

Le travail de cette thèse est d'abord analytique : il permet de comprendre plus en détail les difficultés inhérentes à l'adaptation de domaine, en relevant par exemple les inadéquations existant entre la théorie proposée et la réalité pratique des modèles d'apprentissage profond appliqués à l'imagerie. Il montre par ailleurs la fragilité des méthodes existantes, par exemple leur sensibilité exacerbée à un grand nombre de paramètres exogènes ainsi qu'au transfert considéré, ce qui remet en question la manière dont ces méthodes sont évaluées et comparées dans la littérature. Mais étudier ces facteurs exogènes, qui constituent le biais inductif favorisant ou non la transférabilité des représentations, est également bénéfique : approcher le problème de l'adaptation de domaine sous l'angle du biais inductif permet de s'inspirer de l'ensemble littératures voisines (meta-learning, pré-entraînement, représentations désentrelacées...) et donc d'explorer des axes de recherche bien plus variés que le simple alignement des distributions, exploration que l'on a pu entamer. Les contributions apportées sont, point par point, les suivantes :

10.1.1 Théorie

La première contribution théorique de cette thèse consiste à compléter le travail critique apporté par Zhao et al. [2019] et Johansson et al. [2019] sur plusieurs axes : nous relevons notamment dans le chapitre 5 que la théorie ne tire pas parti des spécificités des problèmes d'adaptation de domaine en imagerie. Nous montrons par-ailleurs expérimentalement que ces bornes donnent des résultats triviaux sur des problèmes d'imagerie avec des modèles profonds.

La seconde contribution théorique est un ensemble de nouvelles bornes pour l'adaptation de domaine basées sur les courbes PR / Lorenz, ainsi que sur les Phi-divergences. Ces bornes n'apportent pas de rupture en termes de performance pratique, mais sont plus fines que celles proposées par Ben-David. Un papier présentant ces nouvelles bornes a été soumis au journal JMLR et est actuellement en cours de revue.

10.1.2 Lien entre théorie et pratique

La première contribution réside dans une étude expérimentale qui exhibe les faiblesses des algorithmes d'alignement des domaines. Nous introduisons un nouveau dataset, MNIST-Algebra, qui permet d'expliquer en partie ce qui est réellement appris par ces algorithmes. L'ensemble de ces contributions ont été publiées à ICIP 2020 [Siry et al., 2020a].

La seconde contribution est une étude montrant l'importance des biais inductifs dans l'adaptation de domaine : on montre qu'un grand nombre de facteurs extérieurs à l'algorithme d'adaptation de domaine ont une influence conséquente sur la performance finale du modèle. On identifie ensuite une partie de ces facteurs, que nous interprétons comme autant de réalisations du biais inductif. Ces travaux ont également fait l'objet d'un papier, soumis au journal CVIU et actuellement en cours de revue.

La troisième contribution, moins aboutie, est une étude portant sur l'intérêt de l'adaptation de domaine dans le cas semi-supervisé, que l'on compare à des baselines plus simples et moins exigeantes en termes de conditions expérimentales. On montre que ces baselines peuvent être parfois plus performantes, ce qui pose la question de l'utilité de l'adaptation de domaine.

10.1.3 Ouverture sur d'autres méthodes

La première contribution est l'évaluation d'un algorithme de méta-apprentissage pour l'adaptation de domaine directement inspiré de MAML. Contrairement aux travaux similaires qui traitent principalement de domain generalization, on évalue la capacité de l'algorithme à apprendre un seul transfert, en généralisant vers des tâches impliquant de nouvelles classes.

La seconde contribution est l'évaluation d'une méthode impliquant des représentations désentrelacées dans le cadre de la résolution d'un problème d'UDA.

La troisième contribution est une étude approfondie des espaces de descripteurs appris par les modèles pré-entraînés, qui s'intéresse notamment à la meilleure façon de les mettre à contribution pour résoudre un problème d'adaptation de domaine.

10.2 Perspectives

Ce travail de thèse a permis de mettre en évidence que l'adaptation de domaine était un problème difficile et souvent mal défini, du moins dans le cas des problèmes de vision par ordinateur. Il révèle largement les insuffisances des méthodes basées uniquement sur un critère d'alignement. Par conséquent, pour les recherches futures, nous sommes encouragés à chercher du côté des littératures voisines, telles que celle de l'apprentissage de représentations ou du pré-entraînement, actuellement en pleine dynamique et proposant un panel varié de méthodes que l'on peut facilement étudier dans le cadre de l'adaptation de domaine.

La fragilité et les conditions expérimentales exigeantes propres aux méthodes d'adaptation de domaine posent la question de leur utilité réelle. Par exemple, le cas le plus étudié, à savoir celui de l'UDA, ne semble pas s'appliquer à beaucoup de cas concrets. En effet, dans un problème réel, soit on ne dispose que du domaine source (car il est physiquement impossible de récolter des échantillons du domaine cible). Soit on dispose également de données du domaine cible : dans ce cas, on a tout intérêt à annoter un faible nombre d'échantillons (ce qui est rarement coûteux) puisque cela permet de mettre l'adaptation de domaine en concurrence avec tout un panel d'autres méthodes, en particulier celles de few-shot. Notons qu'il est de plus très difficile de constituer un domaine source et un domaine cible possédant exactement les mêmes classes. Enfin, la performance de ces méthodes étant plus qu'imprévisible selon le transfert considéré, il apparaît de toute façon nécessaire de contrôler le modèle obtenu avec un ensemble de validation du domaine cible annoté, ce qui est paradoxal. Toutes ces observations sont autant d'arguments pour concentrer les futures recherches sur des cadres expérimentaux plus adaptés aux situations concrètes, par exemple celui de la « domain generalization ».

Pour terminer, une autre famille de méthodes prometteuses que l'on n'a pas étudiée au cours de cette thèse est celle des approches multi-modales, en particulier celles mélangeant langage et image. En effet, on a pu montrer qu'en pré-entraînant un modèle sur un très grand nombre d'images légendées par une phrase (ce qui est très facile à trouver sur internet), on pouvait obtenir un classifieur zero-shot pour n'importe quel nouveau problème de classification, en spécifiant simplement les classes à l'aide de phrases. Le modèle montre d'une part des capacités de généralisation impressionnantes, que l'on peut probablement retrouver dans un scénario d'adaptation de domaine. D'autre part, spécifier les classes à l'aide de phrases pourrait permettre d'éliminer les cas ambigus. Prenons par exemple le cas MNIST→SVHN : ce problème est ambigu, car un exemple de SVHN contient potentiellement plusieurs chiffres, et il faut savoir *à priori* que c'est celui du centre qui donne la classe. Avec un tel modèle, on pourrait par exemple représenter la classe 2 par la phrase « une image avec un 2 au centre »plutôt que « une image de 2 »pour résoudre rapidement l'ambiguïté. Cette méthode ne requérant pas de domaine source, elle pose la question de l'utilité de l'adaptation de domaine, dans la mesure où une phrase en langage naturel est peut-être plus efficace pour spécifier entièrement une classe qu'une image provenant d'un domaine différent.

10.3 Publications

Une partie des contributions apportées au cours de cette thèse a été publiée dans une conférence de rang international. Deux autres parties des contributions ont été soumises dans deux journaux internationaux et sont actuellement en cours d'évaluation. Les deux preprint sont toutefois disponibles en ligne. Ci-après un résumé des publications publiées ou en attente de publication :

— *A study of alignment mechanisms in Adversarial Domain Adaptation.*
[Siry et al., 2020a]. ICIP 2020

— *On the Theoretical Equivalence of Several Trade-Off Curves Assessing Statistical Proximity.*
[Siry et al., 2020b]. JMLR. Soumis, en cours d'évaluation

— *On the Inductive Biases of Deep Domain Adaptation.*
[Siry et al., 2021]. CVIU. Soumis, en cours d'évaluation

Bibliographie

David Acuna, Guojun Zhang, Marc T Law, and Sanja Fidler. f-domain adversarial learning : Theory and algorithms. In *International Conference on Machine Learning*, pages 66–75. PMLR, 2021. 42, 44, 45, 47

Rohit Agrawal and Thibaut Horel. Optimal bounds between f-divergences and integral probability metrics. In *International Conference on Machine Learning*, pages 115–124. PMLR, 2020. 43

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society : Series B (Methodological)*, 28(1) :131–142, 1966. 34

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv :1701.07875*, 2017. 25

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg : Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 17, 66

Hangbo Bao, Li Dong, and Furu Wei. Beit : BERT pre-training of image transformers. *CoRR*, abs/2106.08254, 2021. 19, 88

Shai Ben-David and Ruth Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. In *International Conference on Algorithmic Learning Theory*, pages 139–153. Springer, 2012. 14, 41

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2) :151–175, 2010a. 20

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2) :151–175, 2010b. 10, 14, 16, 22, 30, 31, 40, 46, 48

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 81–88, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273507. 14

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016. 18, 76, 77, 79

Konstantinos Bousmalis, N. Silberman, David Dohan, D. Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017. 16, 55

Victor Bouvier, Philippe Very, Céline Hudelot, and Clément Chastagnol. Hidden covariate shift : A minimal assumption for domain adaptation. *ArXiv*, abs/1907.12299, 2019. 17

Victor Bouvier, Philippe Very, Clément Chastagnol, Myriam Tami, and Céline Hudelot. Robust domain adaptation : Representations, weights and inductive bias. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 353–377. Springer, 2020. 16, 17, 49, 54

Ruichu Cai, Zijian Li, Pengfei Wei, Jie Qiao, Kun Zhang, and Zhifeng Hao. Learning disentangled semantic representation for domain adaptation. *IJCAI : proceedings of the conference*, 2019 : 2060–2066, 2019. 18, 19, 76, 77

Yue Cao, Mingsheng Long, and Jianmin Wang. Unsupervised domain adaptation with distribution matching machines. In *AAAI*, 2018. 17

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 19, 88

Chaoqi Chen, Weiping Xie, Tingyang Xu, Wen bing Huang, Yu Rong, Xinghao Ding, Yue Huang, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 627–636, 2019. 17

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020a. 90

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020b. 19, 88

Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1416–1425, 2019. 17

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. *CoRR*, abs/1602.07576, 2016. 55

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *ArXiv*, abs/1705.08848, 2017. 14

Imre Csiszár. An information-theoretic inequality and its application to proofs of the ergodicity of markoff chains. *Magyar Tud. Akad. Mat. Kutato Int. Koezl.*, 8 :85–108, 1964. 34

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006. 38

Morris H DeGroot. Uncertainty, information, and sequential experiments. *The Annals of Mathematical Statistics*, 33(2) :404–419, 1962. 34, 35

Josip Djolonga, Mario Lucic, Marco Cuturi, Olivier Bachem, Olivier Bousquet, and Sylvain Gelly. Precision-recall curves using information divergence frontiers. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research. PMLR, 2020. 34

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015. 19

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *CoRR*, abs/1605.09782, 2016. 18, 77

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017. 17, 70

Tzu-Chien Fu, Yen-Cheng Liu, Wei-Chen Chiu, Sheng-De Wang, and Yu-Chiang Frank Wang. Learning cross-domain disentangled deep representation with supervision from A single domain. *CoRR*, abs/1705.01314, 2017. 18, 76

Yaroslav Ganin, E. Ustinova, Hana Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *ArXiv*, abs/1505.07818, 2016. 10, 15, 16, 22, 24, 49, 54, 56, 79, 119

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. Pac-bayesian theorems for domain adaptation with specialization to linear classifiers. *ArXiv*, abs/1503.06944, 2015. 14, 30

Pascal Germain, Amaury Habrard, François Laviolette, and Emilie Morvant. A new pac-bayesian perspective on domain adaptation. In *International conference on machine learning*, pages 859–868. PMLR, 2016. 14, 30

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018. 19, 88

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012. doi: 10.1109/CVPR.2012.6247911. 54, 118

Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *NeurIPS*, 2018. 18, 76

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 15, 25

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent : A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020. 19, 88

Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017. 79

Peter Harremoës. A new look on majorization. *Proceedings ISITA 2004, Parma, Italy*, pages 1422–1425, 2004. 35

Philip Häusser, Thomas Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2784–2792, 2017. 17, 49, 51, 54, 56

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 91

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. *CoRR*, abs/2111.06377, 2021. 19, 88

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae : Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 18, 76

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. 14

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5) :550–554, 1994. doi: 10.1109/34.291440. 117

Maximilian Ilse, Jakub M. Tomczak, and Patrick Forr'e. Designing data augmentation for simulating interventions. *ArXiv*, abs/2005.01856, 2020. 66

Sergey Ioffe and Christian Szegedy. Batch normalization : Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 61

Fredrik D. Johansson, D. Sontag, and R. Ranganath. Support and invertibility in domain-invariant representations. *ArXiv*, abs/1903.03448, 2019. 16, 30, 49, 50, 106

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 17, 54, 73

Donghyun Kim, Kaihong Wang, Stan Sclaroff, and Kate Saenko. A broad study of pre-training for domain generalization and adaptation. *arXiv preprint arXiv :2203.11819*, 2022. 97

Taekyung Kim and Changick Kim. Attract, perturb, and explore : Learning a feature alignment network for semi-supervised domain adaptation. In *ECCV*, 2020. 20, 100

Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014. 77

Wouter M. Kouw. An introduction to domain adaptation and transfer learning. *ArXiv*, abs/1812.11806, 2018. 14

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 8

Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 31, 2018. 17, 51

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998. 117

Seunghun Lee, Sunghyun Cho, and Sunghoon Im. Dranet : Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. *CoRR*, abs/2103.13447, 2021. 18, 76, 79

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 54, 119

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize : Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 17, 66, 71, 74

Xiao Li and Jeff Bilmes. A bayesian divergence prior for classiffier adaptation. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 275–282, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. 14, 30

Friedrich Liese and Igor Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10) :4394–4412, 2006. 34, 46

Zinan Lin, Ashish Khetan, Giulia C. Fanti, and Sewoong Oh. Pacgan : The power of two samples in generative adversarial networks. *CoRR*, abs/1712.04086, 2017. 36

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan : The power of two samples in generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1498–1507, 2018. 34, 36

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *CoRR*, abs/1811.12359, 2018. 18, 76

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015. 10, 15, 49

Max O Lorenz. Methods of measuring the concentration of wealth. *Publications of the American statistical association*, 9(70) :209–219, 1905. 35

Fangrui Lv, Jian Liang, Kaixiong Gong, Shuang Li, Chi Harold Liu, Han Li, Di Liu, and Guoren Wang. Pareto domain adaptation. *arXiv preprint arXiv :2112.04137*, 2021. 17, 54

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. *ArXiv*, abs/1511.05644, 2015. 18, 76, 77

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation : Learning bounds and algorithms. *arXiv preprint arXiv :0902.3430*, 2009. 14, 30, 48

Emile Mathieu, Tom Rainforth, Siddharth Narayanaswamy, and Yee Whye Teh. Disentangling disentanglement. *ArXiv*, abs/1812.02833, 2018. 18, 76

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018. 25

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 117

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11) :5847–5861, 2010. 42, 43, 45

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016. 19

Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015. 8

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders : Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016. 19, 88

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda : The visual domain adaptation challenge, 2017. 118

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. *ArXiv*, abs/1904.12347, 2019. 18, 19, 76, 77

Benedetto Piccoli, Francesco Rossi, and Magali Tournus. A wasserstein norm for signed measures, with application to nonlocal transport equation with source term. *arXiv preprint arXiv :1910.05105*, 2019. 33

Christoph Raab, Philipp Vath, Peter Meier, and Frank-Michael Schleif. Bridging adversarial and statistical domain transfer via spectral adaptation networks. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020. 54, 73

Janarthanan Rajendran, Alex Irpan, and Eric Jang. Meta-learning requires meta-augmentation. *CoRR*, abs/2007.05549, 2020. 69

Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. A survey on domain adaptation theory. *arXiv preprint arXiv :2004.11829*, 2020. 8, 14, 30, 40, 42, 48

Avraham Ruderman, Mark Reid, Darío García-García, and James Petterson. Tighter variational representations of f-divergences via restriction to probability measures. *arXiv preprint arXiv :1206.4664*, 2012. 43

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 17

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8049–8057, 2019. 20, 100

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5234–5243. Curran Associates, Inc., 2018. 33, 34

Mateus Sangalli, Samy Blusseau, Santiago Velasco-Forero, and Jesús Angulo. Scale equivariant neural networks with morphological scale-spaces. In *DGMM*, 2021. 55

Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter V. Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. *ArXiv*, abs/2107.08221, 2022. 55

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise : Consistency and maximal denoising. In *Conference on learning theory*, pages 489–511. PMLR, 2013. 34

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Adversarial representation learning for domain adaptation. *ArXiv*, abs/1707.01217, 2017. 14, 16, 22, 30, 32, 48

Rui Shu, Hung Hai Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *ArXiv*, abs/1802.08735, 2018. 16, 17

Loic Simon, Ryan Webster, and Julien Rabin. Revisiting precision recall definition for generative modeling. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5799–5808, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 33, 34, 35, 121

Rodrigue Siry, Loic Simon, and Frederic Jurie. A study of alignment mechanisms in adversarial domain adaptation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1816–1820. IEEE, 2020a. 16, 22, 30, 57, 106, 108

Rodrigue Siry, Ryan Webster, Loïc Simon, and Julien Rabin. Equivalence of several curves assessing the similarity between probability distributions. *CoRR*, abs/2006.11809, 2020b. 30, 33, 41, 108

Rodrigue Siry, Louis Hémadou, Loïc Simon, and Frédéric Jurie. On the inductive biases of deep domain adaptation. *CoRR*, abs/2109.07920, 2021. 16, 66, 108, 119, 120

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *ArXiv*, abs/2006.09661, 2020. 82

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56) :1929–1958, 2014. 61

Ingo Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26(2) :225–287, 2007. 48

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(35) :985–1005, 2007. 14

Baochen Sun and Kate Saenko. Deep coral : Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 10, 15, 49

Y. Sun, Eric Tzeng, Trevor Darrell, and Alexei A. Efros. Unsupervised domain adaptation through self-supervision. *ArXiv*, abs/1909.11825, 2019. 19

Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(5), 2007. 48

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion : Maximizing for domain invariance. *arXiv preprint arXiv :1412.3474*, 2014. 15

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 16, 22, 25, 54

Tim van Erven and Peter Harremoës. Rényi divergence and majorization. In *2010 IEEE International Symposium on Information Theory*, pages 1335–1339, 2010. doi: 10.1109/ISIT.2010.5513784. 34, 35

C. Villani. *Optimal Transport : Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2016. ISBN 9783662501801. 32

Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign : Coordinating domain alignment and classification for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16643–16653, 2021. 17, 66, 70

Junfeng Wen, Chun-Nam Yu, and Russell Greiner. Robust learning under uncertain test distributions : Relating covariate shift to model misspecification. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 631–639, Bejing, China, 22–24 Jun 2014. PMLR. 14

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Chase Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *ICML*, 2019. 17

Xiang Xu, Xiong Zhou, Ragav Venkatesan, Gurumurthy Swaminathan, and Orchid Majumder. d-sne : Domain adaptation using stochastic neighborhood embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2497–2506, 2019. 17

Jeongbeen Yoon, Dahyun Kang, and Minsu Cho. Semi-supervised domain adaptation via sample-to-sample self-distillation. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1686–1695, 2022. 20, 100

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2715–2724, June 2019. doi: 10.1109/CVPR.2019.00283. 17

B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Third IEEE International Conference on Data Mining*, pages 435–442, 2003. doi: 10.1109/ICDM.2003.1250950. 14

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv :1611.03530*, 2016. 42

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019. 14, 16, 17, 22, 23, 29, 30, 32, 46, 48, 49, 92, 94

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019. 16, 30, 49, 106

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. 16

Annexe A

Jeux de données utilisés

A.0.1 Jeux de données de chiffres

MNIST LeCun et al. [1998] est une base de données de chiffres écrits à la main. Ce sont des images en noir et blanc, réparties en 10 classes, normalisées et centrées de 28 pixels de côté. Le jeu de données regroupe 60000 images d'apprentissage et 10000 images de test.

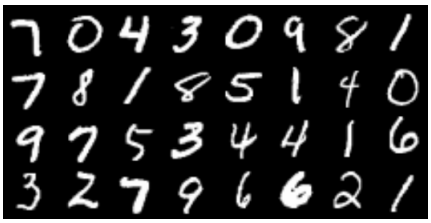

FIGURE A.1 – Échantillons de MNIST

USPS Hull [1994] Un jeu de données très similaire à MNIST. Les images sont de dimension 16×16. La base contient 7291 images d'entraînement et 2007 images de test.

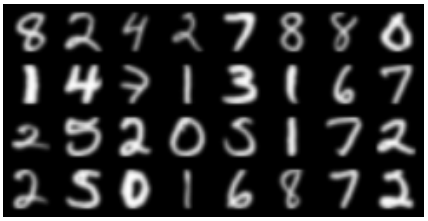

FIGURE A.2 – Échantillons de USPS

SVHN Netzer et al. [2011] est un autre jeu de données de reconnaissance de chiffres. Il est constitué de vignettes de taille 32×32 de numéros de maison. Les chiffres qui y figurent sont qualitativement beaucoup plus riches que ceux de MNIST : plus grande variété de polices, couleurs et tailles. Seul le chiffre du centre de la vignette est indicatif de la classe, des chiffre auxiliaires peuvent exister sur l'image et peuvent par-ailleurs être un important facteur de confusion dans un contexte d'adaptation de domaine. Le je de données contient 73257 images d'entraînement et 26032 images de test.

FIGURE A.3 – Échantillons de SVHN

A.0.2 Jeux de données d'images naturelles

Office-31 **Gong et al. [2012]** est un jeu de données très utilisé en adaptation de domaine. Il comporte 31 classes, généralement des objets domestiques, et 3 domaines : Amazon (photo commerciale de l'objet parfaitement détouré sur fond blanc), Webcam (photo prise par une webcam) et DSLR (photo prise par un appareil photo). Le jeu de données comporte en réalité très peu d'images (10 à 20) par classe

FIGURE A.4 – Échantillons de Office-31

VisDA **Peng et al. [2017]** Est un jeu de données de grande échelle conçu à la base pour l'adaptation de domaine multi-source. Il comporte 6 domaines plus ou moins différents entre eux : « Real », des photogra- phies réelles, « Painting », des peintures assez réalistes, « Sketch », des crayonnés assez réalistes, « quickdraw », des dessins très sommaires, « clipart » des dessins cartoonesques et « infograph », des posters explicatifs incluant l'objet d'intérêt. Chacun de ces domaines contient les mêmes 345 classes, avec environ 200 images pour une classe donnée dans un domaine donné. Nous utilisons la version 2019 de la base, qui est mise à jour tous les ans pour le challenge du même nom.

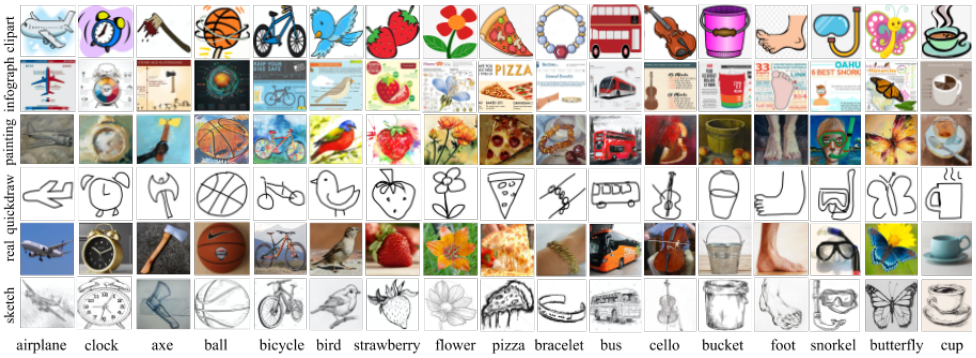

FIGURE A.5 – Échantillons de VisDA

PACS Li et al. [2017] PACS est un jeu de données très utilisé en adaptation et généralisation de domaine. Il a été proposé pour remplacé Office-31, jugé trop facile pour les modèles pré-entraînés. Il inclut 4 domaines : Photo (1670 images), Art Painting (2048 images), Cartoon (2344 images) et Sketch (3,929 images). Chaque domaine contient 7 classes.

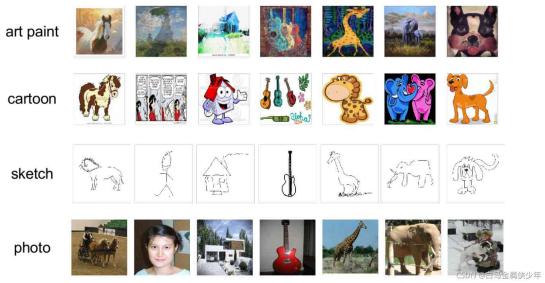

FIGURE A.6 – Échantillons de PACS

A.0.3 Variantes synthétiques de MNIST

MNIST-M Ganin et al. [2016] Est une variante de MNIST conçue en fusionnant le fond noir de chaque échantillon avec une image réelle et en fusionnant l'intérieur blanc du chiffre avec une autre image réelle, ce qui crée une variété de textures parasites.

FIGURE A.7 – Échantillons de MNIST-M

Color-MNIST Siry et al. [2021] est une variante augmentée synthétiquement de MNIST que nous proposons afin de diversifier les benchmarks d'adaptation de domaine. On le synthétise en colorant le fond de chaque échantillon avec une couleur aléatoire et en colorant le chiffre avec une autre couleur aléatoire.

FIGURE A.8 – Échantillons de Color-MNIST

MNIST-T Siry et al. [2021] est une variante augmentée synthétiquement de MNIST que nous proposons afin de diversifier les benchmarks d'adaptation de domaine. On l'obtient en faisant subir aux images de MNIST des rotations, translations et mises à l'échelle aléatoires.

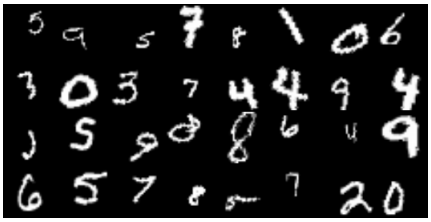

FIGURE A.9 – Échantillons de MNIST-T

MNIST-Inv **Siry et al. [2021]** est une variante augmentée synthétiquement de MNIST que nous proposons afin de diversifier les benchmarks d'adaptation de domaine. On l'obtient en inversant les couleurs de MNIST.

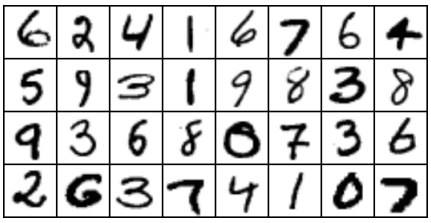

FIGURE A.10 – Échantillons de MNIST-Inv

Annexe B

Démonstrations

B.0.1 Démonstration de la borne de Ben-David et al.

En notant $h* = \underset{h\in\mathcal{H}}{\text{argmin}}(\epsilon_S(h) + \epsilon_T(h))$, $\lambda_S = \epsilon_S(h^*)$, $\lambda_T = \epsilon_T(h^*)$, $\lambda = \lambda_S + \lambda_T$

On démontre d'abord l'inégalité suivante :

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) &= 2 \sup_{h,h'\in\mathcal{H}} |\underset{\mathcal{S}_X}{\mathbf{Pr}}(h(x) \neq h'(x)) - \underset{\mathcal{T}_X}{\mathbf{Pr}}(h(x) \neq h'(x))| \\ &= 2 \sup_{h,h'\in\mathcal{H}} |R_{\mathcal{S}}(h, h') - R_{\mathcal{T}}(h, h')| \\ &\geq 2|R_{\mathcal{S}}(h, h') - R_{\mathcal{T}}(h, h')| \end{aligned} \tag{B.1}$$

On peut désormais démontrer la borne supérieure :

$$\begin{aligned} R_{\mathcal{T}}(h) &\leq R_{\mathcal{T}}(h^*) + R_{\mathcal{T}}(h, h^*) \\ &\leq R_{\mathcal{T}}(h^*) + R_{\mathcal{S}}(h, h^*) - R_{\mathcal{S}}(h, h^*) + R_{\mathcal{T}}(h, h^*) \text{ (inégalité triangulaire)} \\ &\leq R_{\mathcal{T}}(h^*) + R_{\mathcal{S}}(h, h^*) + |R_{\mathcal{S}}(h, h^*) - R_{\mathcal{T}}(h, h^*)| \\ &\leq R_{\mathcal{T}}(h^*) + R_{\mathcal{S}}(h, h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) \text{ (lemme)} \\ &\leq R_{\mathcal{T}}(h^*) + R_{\mathcal{S}}(h) + R_{\mathcal{S}}(h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) \text{ (inégalité triangulaire)} \\ &\leq \lambda + R_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) \\ &\leq \lambda + R_{\mathcal{S}}(h) + \frac{1}{2} \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}_X, \mathcal{T}_X) + 4\sqrt{\frac{2dlog(2m') + log(\frac{2}{\delta})}{m'}} \text{ (majoration VC)} \end{aligned} \tag{B.2}$$

B.0.2 Preuve du Lemme 4.1

Nous allons utiliser le résultat suivant tiré du Théorème 5 dans Simon et al. [2019] qui stipule que la précision α_λ est optimale au sens suivant :

Théorème B.1. *Soient P, Q deux distributions de $\mathcal{M}_p(\Omega)$. Alors $\forall\lambda \in \overline{\mathbb{R}^+}$,*

$$\alpha_\lambda = \min_{A\in\mathcal{A}} \lambda(1 - P(A)) + Q(A) \tag{B.3}$$

Le lemme est un corollaire simple de ce théorème :

Démonstration. Il suffit de montrer que pour toute fonction mesurable $0 \leq f \leq 1$,

$$\alpha_\lambda \leq \lambda(1 - \int f dP) + \int f dQ$$

D'après le théorème B.1, cette inégalité est vraie pour les fonctions indicatrices. Outre le fait qu'elle est stable par combinaison convexe et par limite L^∞. Par conséquent, on peut étendre l'inégalité d'abord aux combinaisons convexes de fonctions indicatrices, c'est-à-dire à toutes les fonctions simples à valeurs dans $[0, 1]$. Noter que si à première vue, de simples fonctions à valeurs dans $[0, 1]$ prennet la forme de combinaisons sous-convexes de fonctions indicatrices, on peut exploiter $\mathbf{1}_\emptyset = 0$ pour les exprimer en termes de combinaisons convexes. Ensuite, en passant à la limite par-rapport à la convergence L^∞ et en utilisant les résultats standards de densité, on peut étendre davantage l'inégalité aux fonctions L^∞ à valeurs dans $[0, 1]$. □

Annexe C

Manipulations d'images avec DISTGL

FIGURE C.1 – Génération non-conditionnelle d'échantillons sur tous les transferts considérés par DISTGL avec décodeur déconvolutif ; en haut à gauche = MNIST ∪ Color-MNIST ; en haut à droite MNIST ∪ MNIST-M ; en bas à gauche = MNIST ∪ MNIST-T ; en bas à droite = MNIST ∪ SVHN

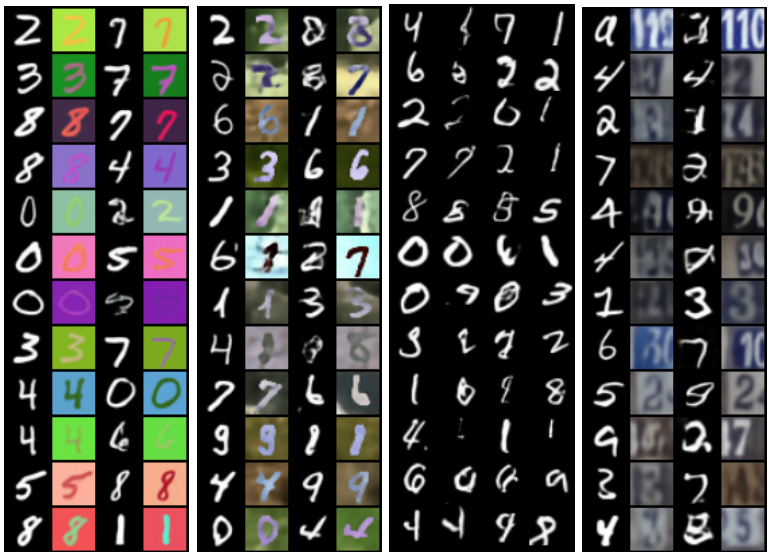

FIGURE C.2 – Manipulation du style pour chaque modèle DISTGL avec décodeur déconvolutif ; de gauche à droite : MNIST→Color-MNIST, MNIST→MNIST-M, MNIST→MNIST-T, MNIST→SVHN ; Colonne 1=échantillon source, Colonne 2=classe du source + style du cible, Colonne 3=classe du cible + style du source, Colonne 4=échantillon cible