On the inductive biases of deep domain adaptation

Rodrigue Siry, Louis Hémadou, Loïc Simon, Frédéric Jurie

▶ **To cite this version:**

Rodrigue Siry, Louis Hémadou, Loïc Simon, Frédéric Jurie. On the inductive biases of deep domain adaptation. 2023. hal-03290701v2

HAL Id: hal-03290701
https://hal.science/hal-03290701v2

Preprint submitted on 25 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Inductive Biases of Deep Domain Adaptation*

Rodrigue Siry Louis Hémadou, Loïc Simon and Frédéric Jurie

April 25, 2023

Abstract

Even if it is not always stated explicitly, the majority of recent approaches to domain adaptation are based on theoretically unjustified assumptions, on the one hand, and (often hidden) inductive biases on the other. This paper first point out that *feature alignment* which is often misrepresented as a minimizer of some theoretical upper-bounds on risk in the target domain is neither necessary nor sufficient to obtain low target risk. This paper also demonstrates, through numerous experiments, that deep domain adaptation methods, in fact, rely heavily on some hidden inductive biases found in common practices, such as model pretraining or encoder architecture design. In a third step, the paper argues that using handcrafted priors might not be sufficient to bridge distant domains: powerful parametric priors can be instead learned from data, leading to a large improvement in target accuracy. A meta-learning strategy allowing to find inductive biases that solve specific transfers is proposed. It shows superior performance to that of handcrafted priors on several tasks.

1 Introduction

Deep learning models achieve impressive performance on image classification tasks, when a large number of training samples is provided. However, in many situations, there is not enough labeled data available for the task of interest (the target domain), while there is some in related domains (source domains) but with a statistical bias which makes them impossible to use as is to learn a model. The objective of *domain adaptation* is to exploit the data in the source dataset to obtain a model that performs well in the target domain. In this article, we focus on *unsupervised domain adaptation*, a task for which only unlabeled examples of the target domain are available to train the model.

This work focuses on domain adaptation under the covariate shift assumption, which assumes that the same input data is associated with a single label, regardless of the domain to which it belongs (see Section 3 for a formal definition of the covariate shift). Indeed, transfer learning usually assumes that the labeling function is consistent across all domains and only accounts for qualitative shifts in the distribution of input data.

Following Bouvier et al. (2020), we analyze domain adaptation through the scope of *inductive biases*. An inductive bias is a set of hidden assumptions that condition the behavior of the model on unseen data (in that case, the target domain). Indeed, the vast majority of domain adaptation

*Preprint accepted for publication in Computer Vision and Image Understanding

algorithms work by enforcing some regularization criterion that is expected to enhance the transferability of representations to the target domain, such regularization being a contribution to the inductive bias.

Among them, domain alignment (e.g., Ganin et al. (2016); Häusser et al. (2017); Xu et al. (2019); Zhang et al. (2019)) is the dominant approach for solving unsupervised domain adaptation problems that fall under the covariate shift assumption, i.e., when the two domains contain the same classes and the same labeling rule. Their goal is to enforce domain-independence (in the distributional sense) of the hidden features produced by the encoder model. These methods are based on a series of theoretical results related to unsupervised domain adaptation: such as Ben-David et al. (2010); Mansour et al. (2009); Redko et al. (2020); Zhang et al. (2019) which proposed various upper bounds on target risk. Indeed, in almost all these contributions, the bound includes - among other things - the source risk and some measure of deviation between source and target input data distributions.

The first thing to notice is that this theoretical justification is flawed: further analyses of Zhao et al. (2019); Johansson et al. (2019) and Bouvier et al. (2020) have shown that practical domain-alignment algorithms are in fact largely inconsistent with the theory from which they claim to be derived and may even be counterproductive in some cases. We summarize all the important facts in section 3.

Our second remark is that the empirical success of such methods on popular benchmark datasets can be better explained thanks to implicit inductive biases found in the standard practice of domain alignment literature. For example use of pre-trained models or augmentation. In other words, such biases help to ensure that domain alignment behaves well despite the lack of conclusive theoretical guarantees. We give a comprehensive presentation of existing biases and illustrate their effects through various experiments in section 4.

This analytical work shows that human-designed priors may be sub-optimal and cannot be safely combined to solve any domain adaptation scenario. This introduces the second contribution of our paper, namely the introduction of meta-learning principles to produce more adaptable models. We propose a strategy to *learn* a flexible and powerful parametric inductive bias to enhance the transferability of representations for a single transfer. In contrast with Li et al. (2018) and Balaji et al. (2018) which employ meta-learning in the multi-source domain generalization setting (test on same classes but on a new, unseen domain), we rather consider the ability of models to perform the same transfer with tasks involving new test classes.

2 Related work

In this section, we review the works and concepts present in the literature and related to the question of domain adaptation, in order to be able to position our work within this context.

2.1 Learning bounds

The goal of domain adaptation theory is to find informative upper bounds on target risk to estimate how well a task can be adapted to the target domain. Most of them involve a divergence between the source and target distributions. They vary according to the chosen divergence and the statistical framework they are based on: (VC, Rademacher, PAC-Bayes).

Ben-David et al. (2010) first proposed a bound that uses the $H\Delta H$ divergence, which accounts for how much two hypotheses can deviate from one another and in a different way in the two

domains. Zhang et al. (2019) further refined the bound by relaxing the divergence to a single adversarial hypothesis. Mansour et al. (2009) provides another classifier-based bound that does not require the covariate shift assumption. Shen et al. (2017) proposes a bound based on the Wasserstein distance that is robust to disjoint supports. Several bounds were also proposed for the PAC-bayesian setting (Li and Bilmes, 2007; Germain et al., 2015, 2016). Simon et al. (2020) present a divergence based on PR curves. More details about the bounds will be provided in the next section, but we also advise the reader to refer to Kouw (2018), Redko et al. (2020) for an exhaustive list of existing domain adaptation bounds.

2.2 Domain-Invariant feature learning

The large majority of domain adaptation methods aim at producing a domain-invariant feature representation at the output of an encoder model by aligning source and target feature distributions. To perform this feat, early works minimize some closed-form measure of discrepancy between distributions; DeepCORAL (Sun and Saenko, 2016) trains the encoder model to align the distribution means and variance-covariance matrices. Tzeng et al. (2014) uses the more powerful Maximum Mean Discrepancy (MMD) metric, wich was then extended by Long et al. (2015) with multiple kernels. With the advent of GANs (Goodfellow et al., 2014), adversarial training was quickly used for domain-alignment, leading to the DANN algorithm (Ganin et al., 2016). A wide set of variants of DANN followed: ADDA (Tzeng et al., 2017) uses a separate feature encoder for source and target, Shen et al. (2017) uses a Wasserstein critic instead of the standard Jensen-Shannon. PixelDA (Bousmalis et al., 2017) aligns distributions in image-space by using an image translation model. As domain-alignment was proven to be insufficient for good target performance, new improvements were found by the community to better condition the process of distribution alignment: (Shu et al., 2018; Häusser et al., 2017; Xu et al., 2019) exploit a cluster assumption in feature space to stabilize alignment. (Cicek and Soatto, 2019; Chen et al., 2019; Lv et al., 2021; Zhang et al., 2019; Saito et al., 2018; Kang et al., 2019; des Combes et al., 2020) make use of pseudo-labels to condition alignment to class information. Kumar et al. (2018) performs alignment on an ensemble of feature-spaces and forces a single classifier to agree on all of them.

Inspired by the representation learning literature, other methods add auxiliary modeling objectives to the features to improve their robustness, interpretability and transferability: (Peng et al., 2019; Bousmalis et al., 2016) learn a feature space where class and style information are disentangled. Sun et al. (2019) completes alignment with a self-supervision objective.

2.3 Controversies

Alignment methods seek theoretical justification in the learning bounds derived from Ben-David et al. (2010) by applying them in feature space. However, further analyses from (Shu et al., 2018; Zhao et al., 2019; Johansson et al., 2019; Bouvier et al., 2020; Siry et al., 2020) proved that domain-alignment only partially minimizes those bounds and is not sufficient for provable good target accuracy. We will adopt the same critical approach in the next section and give additional examples.

2.4 Inductive bias

We refer as an *inductive bias* any set of assumptions we can use to help a model generalize to unseen data. The importance of inductive bias in domain adaptation has been first mentioned in (Bouvier et al., 2020). In this paper, we consolidate this work by unveiling various instances of inductive

bias that were already used in the literature and crucially underpin good target performance. A fact that has not been highlighted much by the domain adaptation community.

2.5 Meta-learning

Meta-learning, or learning to learn, is a recent paradigm in deep learning which can be viewed as a data-centric way to learn a parametric inductive bias that works for some distribution of tasks. First introduced to address few-shot generalization (Finn et al., 2017), meta learning has then been used in domain adaptation, especially in the multi-source domain generalization setting: Li et al. (2018) meta-learns an encoder representation that generalizes to test domains, this was then improved by Balaji et al. (2018) which meta-learns regularization weights optimized for transferable fine-tuning. Wei et al. (2021) uses meta-learning to synchronize the learning dynamics of domain alignment and supervision on source.

3 Two common assumptions that are nevertheless flawed

In this section, we attempt to highlight two commonly held assumptions in the domain adaptation literature and show that these assumptions are in some cases unfounded, thus leaving most of the practical algorithms presented in the literature without any proper theoretical guarantees.

We refer as *source* and *target* domains two distributions S and T over the space of labeled data $X \times Y$. We define S_X and T_X the marginals of S and T over X and S_Y, T_Y the marginals over Y. Finally, we call *domain shift* the discrepancy existing between S and T.

Provided labeled samples from S and unlabeled samples from T, we would like to exploit labeled knowledge from S to obtain a model that minimizes the target risk.

Obviously, this can only be possible if a relationship between S and T exists.

The following general cases are often described in the literature:

1. Covariate shift / inductive transfer: $S_X \neq T_X$ and $P_S(Y|X) = P_T(Y|X)$,
2. Concept shift / transductive transfer: $S_X = T_X$ and $P_S(Y|X) \neq P_T(Y|X)$,
3. Unsupervised transfer: $S_X \neq T_X$ and $P_S(Y|X) \neq P_T(Y|X)$.

Most deep learning domain adaptation contributions indeed invoke the covariate shift assumption, and, among them, several works build upon theoretical upper bounds on the target error. For example, this is the case of DANN (Ganin et al., 2016) which built upon the upper bounds developed in Ben-David et al. (2010).

The most commonly used upper bounds usually take the following form (see e.g., Ben-David et al. (2010); Mansour et al. (2009); Shen et al. (2017) or Redko et al. (2020) for a more exhaustive list).

$$\epsilon_T(h) \leq \epsilon_S(h) + \delta(S_X, T_X) + \lambda \tag{1}$$

where $h \in \mathcal{H}$ is a hypothesis, $\epsilon_S(h)$ and $\epsilon_T(h)$ represent the error associated with h in the two domains.

The second term of the right hand side of the bound, $\delta(S_X, T_X)$ represents a divergence between the marginal distribution of X under both domains, and this divergence involves the class of hypotheses $\mathcal{H}$.

The last term of Eq.(1), λ, underpins the adaptability in the sense that λ is small only if a common hypothesis h^* can perform well on both domains. This term is not amenable to be computed without the labels in the target, but it is expected to be small in well-posed problems.

The first term, $\epsilon_S(h)$, can be easily minimized to zero by training h with a standard cross-entropy cost as the source domain is assumed to be fully labeled. However, for a given hypothesis space, $\delta(S_X, T_X)$ cannot be brought close to zero, because the input distributions are fixed. Furthermore, in the majority of domain changes that are considered in the computer vision literature, the domains actually have supports that are disjoint, i.e. $S_X \perp T_X$.

For instance, pictures of an object taken during the day (the source domain) will have zero probability under the distribution of images taken at night (the target domain). Another example is the MNIST / SVHN pair of domains. Both are well-known datasets of digit recognition: while MNIST displays white-on-black handwritten digits with a moderate variety of shapes and sizes, the richer SVHN is a dataset of house numbers, hence displaying a wide variety of colors, fonts and sizes. Also in that case, it is straightforward to determine with 100% accuracy from which dataset a digit image was drawn.

In such cases, the notion of conditional distribution, say $P_S(Y|X)$, is undetermined outside of the support of S_X. As a result, the constraint $P_S(Y|X) = P_T(Y|X)$ is nonbinding, and **the notion of covariate shift no longer makes sense**. This statement is the first untruth we would like to point out.

Although this fact is often disregarded in domain adaptation theoretical works, which still advocate for the covariate shift assumption, the usual practice is to introduce a feature extractor $\Psi : \mathcal{X} \to \mathcal{Z}$ hence embedding both distributions in latent space $\mathcal{Z}$ where the supports may overlap (e.g. Sun and Saenko (2016); Long et al. (2015); Ganin et al. (2016); Häusser et al. (2017); Zhang et al. (2019)). When the bound is applied on feature-space, $\delta(S_X, T_X)$ can be trivially minimized: to bring the divergence term close to zero, it is sufficient to output a domain-invariant representation; indeed, a feature representation is domain-invariant if the features from the source domain follow the same distribution than features from the target domain. Most methods hence train the encoder to satisfy a dual objective: i) give sufficiently informative features to get a good classification accuracy on source and ii) minimize a measure of discrepancy between source and target feature distributions. The most prominent domain-alignment technique is DANN, which makes use of adversarial learning: it makes the feature extractor compete with a domain classifier in a minimax game; the domain classifier trains itself to recognize the domain from which feature samples come from by minimizing a classification error on domain labels, then the feature extractor trains itself to maximize this error, hence fooling the classifier and bringing source and target distributions closer to each other. After several steps of alternate optimization, the algorithm eventually reaches equilibrium and source and target distributions are aligned. At this point, the $H\Delta H$-divergence of the Ben-David bound from which DANN draws its inspiration is close to zero.

Since the last term λ cannot be evaluated without knowing the target labels, only the first two terms of the RHS are considered in the minimization, and when reasoning in feature space, the prior assumption that λ is small does not hold anymore.

This fact has been highlighted by at least three recent papers, namely (Zhao et al., 2019; Johansson et al., 2019; Bouvier et al., 2020) in a form of a *no-free-lunch* theorem. The gist of this result relies on the observation that **while the first two terms of the upper-bound are minimized, the third one can go out of control**. In particular, the feature extractor can map different regions of T_X of different classes to the same region of $\mathcal{Z}$ leading to a larger target Bayes risk. This label mixing will be further strengthened by minimizing the divergence in scenarios where

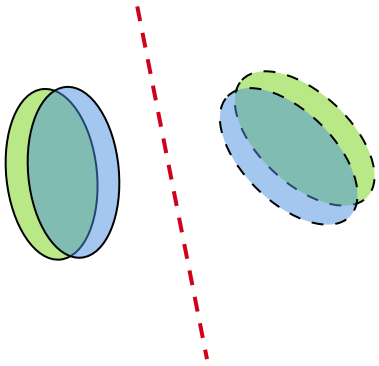

Figure 1: Well-behaved domain-alignment: both divergence between feature marginals and joint error λ are low. Colors denotes domain. Continuous boundaries denotes class 1, dashed boundaries denotes class 2.

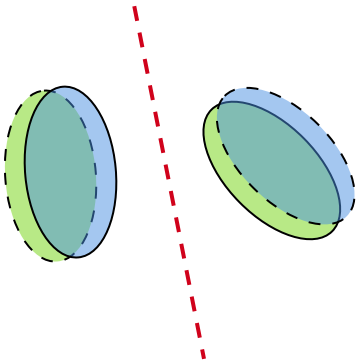

Figure 2: Degenerate domain-alignment: divergence between feature marginals is low but joint error λ is high. Colors denotes domains. Continuous boundaries denotes class 1, dashed boundaries denotes class 2.

classes are balanced differently in the source and target domains, a situation known as *a prior shift* which is not ruled out by the covariate shift assumption). Worst, nothing prevents regions of T_X of a given class to be embedded in regions corresponding to different classes of S_X. We illustrate this case in Figure 2.

Both phenomenons are due to the non-invertibility of Ψ (Johansson et al., 2019), which is necessary for the support of S_Z and T_Z to overlap.

Despite the previous caveat, domain alignment approaches based on upper-bounds are still often successful in practice. On the one hand, this success can be explained by the way in which published results are selected. For instance, while the transfer from SVHN to MNIST is often reported, it is almost never the case of the inverse (MNIST to SVHN – see Fig 3 for a visual comparison of DANN latent representation in both cases). Indeed, the SVHN to MNIST should be considered the "easy" way: as SVHN is richer than MNIST in terms of degrees of freedom (fonts, scales, colors, background textures), the features learned thanks to supervision on SVHN are already robust

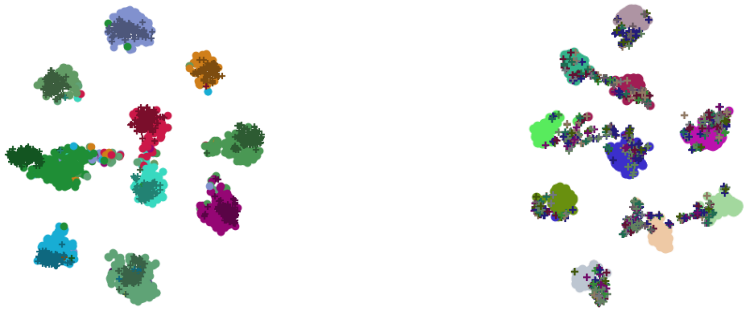

Figure 3: t-SNE plot of feature space after DANN adaptation (labels are color-coded, plain dots are source samples, crosses are target samples). Left: SVHN→MNIST, Right: MNIST→ SVHN. SVHN has many more degrees of freedom (digit fonts, scales, colors) than MNIST, it is therefore not surprising that the features learnt from raw SVHN supervision transfer consistently on MNIST, making the joint feature distributions close. Left: one can see that the alignment obtained with DANN is non-degenerate. Right: The inverse transfer MNIST → SVHN is much harder. Features learnt with MNIST supervision do not extrapolate consistently to the richer SVHN. In that case, applying the DANN algorithm leads to a degenerate alignment.

enough to generalize slightly on MNIST, and thanks to the adjustment brought by the domain alignment, we can see in the left plot of Fig 3 that not only the marginal feature distributions are aligned, but so is the joint feature-label distribution; on each source class cluster is a target cluster made of samples from the same class. The MNIST to SVHN case is much harder: supervision on MNIST does not naturally produce features that are transferable to SVHN, and aligning domains leads to a degenerate case in which marginals are aligned but not the joint distribution. In the right plot of Fig 3, on each source class cluster is a target cluster made of target samples from arbitrary classes, indicating that the encoder used its large capacity to exploit arbitrary sources of information found in the target region of image-space to match the source distribution. The few works reporting results on the MNIST to SVHN transfer use ad-hoc tricks such as augmenting MNIST with random colors Häusser et al. (2017), which qualitatively brings it closer to SVHN, or use of InstanceNorm Kumar et al. (2018), which helps the encoder model to ignore colors and contrasts, thus introducing a good handcrafted invariance that helps conditional alignment. In fact, these "tricks" are part of the inductive bias which will be the subject of the next section.

On the other hand, the analysis of their success is restricted to reasoning on bounds such as Eq 1, while the no-free-lunch theorem should raise concerns on these justifications. It is therefore clear that other explanations must be put forward to understand what makes a transfer susceptible to favorable alignment. In this article, we elaborate on the role of (hidden) inductive biases, as advocated by Bouvier et al. (2020).

4 Inductive (implicit) biases found in domain adaptation practice

Since theoretical bounds alone cannot fully enlighten what makes domain adaptation work, we speculate that its relative success is essentially conditioned by the existence of some inductive bias. For

reference, in machine learning, inductive bias is the set of assumptions (implicit or explicit) used to improve a model in addition to the training data. For example, one of such biases arises from use of pre-trained models, a design choice that is ubiquitous in the common practice of domain adaptation Ganin et al. (2016), Zhang et al. (2019), Shen et al. (2017), especially when dealing with difficult, low-shot transfers such as Office-31 (Gong et al., 2012). In domain-adaptation-related publications, such choices are often pushed into the background in the "experiment" section and presented as subsidiary information (Ganin et al., 2016; Lv et al., 2021). Their overall contribution on transfer success with respect to some domain adaptation method is rarely mentioned nor discussed. This lack of clarity introduces many uncertainties when comparing methods or reproducing results. We therefore propose here a set of experiments to complete the understanding of several cases of inductive bias. We organize the different biases in four main categories namely, i) biases involved in the domain alignment *per se*, ii) biases in the data augmentation process, iii) implicit biases of the feature extractor and iv) biases related to the network architectures. All the experiments involve training a classifier of the form $f = g \circ \Psi$, where Ψ is a feature extractor and g is a classifier.

The experiments presented in this section are based on different datasets commonly used in domain adaptation, *i.e.* MNIST (LeCun et al., 1998), MNIST-M (Ganin et al., 2016), SVHN (Netzer et al., 2011) and PACS (Li et al., 2017). Moreover, we use the default ResNet-18 encoder for Ψ in all experiments except for the BatchNorm ablation one (in that case, a simpler VGG-16 convolutional architecture is used). When applicable, we use the default ImageNet pre-trained weights provided by PyTorch. The classifier g is a simple feedforward network with one hidden layer of width 1024.

4.1 Inductive biases in the alignment method

Our first experiment is designed to evaluate the response of several domain alignment strategies with respect to prior shift. Indeed, we proved in the previous section that perfect marginal feature alignment is not sufficient to align domains; we will now further demonstrate that it is not necessary as well. Worse, it can have a negative impact on target accuracy on some cases when compared to the source-only baseline. To introduce such a shift while keeping the experiment compatible with the covariate shift assumption, we consider two versions of MNIST for both domains. In the source domain the classes are represented in a balanced ratio (i.e. 10%), while in the target we purposely make the ratios uneven, with class partitions ranging from 5 to 20%. Therefore, the transfer problem falls strictly speaking within the covariate shift assumption, *i.e.* $P_S(Y|X) = P_T(Y|X)$.

In this experiment, we consider three forms of domain adaptation. The first one corresponds to training $g \circ \Psi$ without any explicit alignment. In this case, referred to as *source only* (SO), the cost function is computed solely from the cross entropy on source samples. The second and third approaches correspond to DANN Ganin et al. (2016) and associative DA (Häusser et al., 2017). Associative-DA is an alignment technique which only partially aligns source and target distributions to avoid some limitations of DANN. Indeed, if the distribution of classes is not the same in source and target domains, the perfect feature alignment obtained through adversarial techniques is likely to degrade the performance: in such a case, to satisfy a perfect distribution matching criterion, samples from one mode would be moved to another mode to restore the balance and hence take the risk of being misclassified. To avoid this, Associative-DA relies on the assumption that, in the source and target distributions, samples of same class will cluster together, and proceeds to align clusters centroïds without aligning cluster population. We therefore expect Associative-DA to be more robust to prior shifts. Indeed, in the case of prior shift, aligning marginal distributions in the latent representation $Z = \Psi(X)$ is detrimental. This can be seen in Table 1: on the one

	SO	DANN	ASSO-DA
M→ MI	0.99	0.67	0.97

Table 1: Effect of alignment in a prior-shifted transfer; M = balanced MNIST, MI = MNIST with class imbalance

hand, the source-only training performs almost perfectly (0.99 target accuracy). On the other hand, the DANN approach tries to strictly impose the alignment of marginals and deteriorates the performance by a large margin (0.67 target accuracy). Last, the associative DA approaches do the alignment in a looser manner, by relying on the aforementioned *cluster assumption* and reaches a target accuracy of 0.97. In fine, even though the alignment remains slightly harmful to the performance, this experiment exhibits a form of inductive bias that renders alignment more robust to prior shift.

4.2 Data Augmentation

A straightforward and well-known way to enhance feature generalization is to explicitly augment input images with a family of class-preserving image transformations. For example, it can be constructed by a composition of random flips, scaling, rotation, translation, blur or changes in color statistics. Such transformations represent a subset of all the possible class-preserving transformations and can hence partly explain the domain shifts found in real life. Therefore, augmentation of the source images is another inductive bias and its effect on transferability must be measured in the domain adaptation setting. On Figure 2, we show that a simple color randomization of source MNIST digits and backgrounds helps solving the hard MNIST→SVHN transfer, usually out of reach even for pre-trained models, with a drastic 45% increase of the target performance. This shows how a simple but ad-hoc transform can help bridging the complex domain shift that exists between MNIST and 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 2: Effects of data augmentation; M=MNIST, S=SVHN, P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart; Model is pre-trained

4.3 Pretraining and optimizer implicit biases

The next question is probably the most subtle one. It has been noted by Siry et al. (2020) that the target accuracy of a source-only trained model is predictive of a high chance of success in the alignment itself. This observation can lead to several interpretations in terms of inductive biases.

Indeed, most image classification tasks share some common priors: sample images contain a single (or a few very salient) object to classify. Similarly, domain shifts found in real life consist in

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 3: Target accuracies of the KNN classifier on static pre-trained features (PT) and features obtained after source-only fine-tuning (PT+FT). We also give non-pretrained source-only features as a reference (NoPT+FT); P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart, M=MNIST, MM=MNIST-M, S=SVHN

class-preserving transformations, which are a narrow subset of all possible image transforms (see previous section). A realistic domain shift can hence be modeled as a random combination of such transforms. Taking those priors into account defines a narrower family of possible domains and transfers, for which it is easier to design domain-invariant encoding functions.

ImageNet pre-training is widely used in the Deep Learning community. It provides a good, general and robust feature space that considerably fastens convergence to almost all downstream classification tasks, and helps generalizing when training data is scarce. This tendency to generalize easily extends naturally to domain adaptation: in practice, pre-trained features are crucial for successful adaptation of low-shot domains of the Office-31 benchmark and make simpler transfers such as SVHN→ MNIST easier. Even the simplest "source-only" baseline, (fine-tune on source, test on target) already gives convincing results when pre-training is employed. If the same model started from a random initialization scheme, its target accuracy would drop dramatically.

In our experiments, pre-training is performed on the widely known ImageNet dataset, containing more that 1000 categories and a million samples. Solving this large-scale classification problem encourages the discovery of features that are invariant to all class-preserving transforms. It is therefore not surprising to see that the so obtained models can more easily bridge domains.

To characterize the role of pre-training in domain adaptation, we envision the following two hypotheses (the latter being a refined version of the former): i) pretraining the feature extractor leads to an operating point where source and target samples are grouped together in a consistent way in the latent embedding ii) pretraining the feature extractor and the optimization bias of the source cross-entropy leads to such a consistently clustered embedding. To assess the likelihood of these two potential explanations, we conducted two experiments as described below.

Assessing feature transferability with KNN: in order to validate the first hypothesis, we design a simple KNN classifier to predict the label of the feature samples from the target distribution. This classifier works as follow: it finds the top-K (K=50) labeled source nearest feature vectors of the unlabeled target feature embedding, according to a base metric in feature-space (e.g., the L2-norm), then returns the majority vote of their K labels. Good accuracy of this classifier would demonstrate that the representations of the classes in the target domain are close to the ones of the same classes in the source domain. To say it differently, good KNN performance implies natural class equivariance and domain invariance of the considered feature space. However, as mentioned earlier,

expecting the property to be verified for the initial ImageNet features might be too restrictive since the source-only training baseline also involves end-to-end tuning of the encoder model on source samples, which subsequently modifies the feature space. To account for this alternate explanation, the effects of the source-only fine-tuning have to be reported as well. Therefore, we evaluate the KNN classifier on ImageNet pre-trained ResNet-18 features, before (PT) and after (PT+FT) source-only fine-tuning. We also evaluate non-pretrained features obtained after source-only (NoPT+FT) as a baseline. We do so for several transfers and report results in Table 3. On digit classification transfers, we observe better-than-random but low KNN accuracy with not finetuned features, for example 19% for MNIST→ MNIST-M and 25% for SVHN→ MNIST. However, the source and target feature distributions become significantly closer as the encoder is tuned on source, with KNN accuracies rising to 56% and 84% respectively on those two transfers. The exact same trend can be observed on PACS transfers. The benefits of pre-training hence cannot be fully explained by the geometry of pre-trained output features: pre-trained hidden ResNet-18 layers also condition the dynamics of source-only fine tuning that eventually leads to good source and target representations thanks to an implicit optimization bias.

Dead Pixel on CIFAR dataset: To complement this first experiment on the influence of pre-training and source-only finetuning, we have designed an experiment to show that the presence of confusing factors in the source domain can make pretraining detrimental. Note that this experiment is deliberately extreme: it does not follow the prior assumptions of natural classification problems, to a point of completely defeating the commonly useful bias of ImageNet pretraining. For this experiment, we build a synthetic transfer tailored to make pre-training less efficient than random initialization. In both domains, we construct an image sample by taking a random CIFAR-10 image and set its i^{th} upper-left pixel to a fixed color value (grey), the pixel index i is chosen between 1 and 10. In the source domain, i matches the original CIFAR-10 image class. On the contrary, in the target domain, the dead pixel index and the original image class are completely decorrelated, and the image label is given by the dead pixel and not by the object contained in the image. The task is to evaluate to which extend the classifier is tied to the so-called Dead Pixel.

This experiment exposes one of the main obstacles in domain adaptation: the presence of confounding factors that are present in the source domain but absent from the target domain. When supervised on source, we expect pretraining to introduce a bias associating the content of image with the label. It is therefore highly probable that the pre-trained model will ignore the dead pixel and exploit features from the objects, as the initial parameters already make this information salient and filter out pixel-level details. In the target domain, where the CIFAR-10 object is not useful anymore to predict the image label, the pre-trained model should fail even if source-only fine-tuning is performed. On the contrary, a randomly initialized classifier should quickly identify the dead pixel as the safest, easiest way to predict the image label and should completely ignore the CIFAR-10 object. Consequently, the classifier trained from scratch on source should achieve a good accuracy on target. Results for the dead pixel CIFAR transfer (D-Pix) can be found on Table 4: as expected, pre-training performs worse, but still manages to exploit some Dead-Pixel information, leading to reasonable performance on target. The model trained from scratch performs consistently in both domains and reaches maximum accuracy on target. Also, the explicit alignment with DANN hardly improves over source only in this scenario.

We conducted additional experiments in Table 4 to sum-up the influence of pre-training on transfers that satisfy the properties of realistic domain shifts. We observe a consistent performance increase when pre-training is used, in both source-only and DANN adaptation. On most digits

	no pretraining	
	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

	with pretraining	
	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 4: Target accuracies for the source-only baseline and the DANN alignment algorithm; **Left**: No pretraining, **Right**: pretraining; P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart, M=MNIST, MM=MNIST-M, S=SVHN, D-Pix=Dead Pixel CIFAR transfer

transfers, DANN provides an additional increase in performance even in the non-pretrained case (+46% on MNIST → MNIST-M, +6% on SVHN→MNIST without pre-training). Those transfers fall in the "easy" category, in which supervision on source naturally builds features that are transferable to target without pre-training. The only exception is the very hard MNIST → SVHN in which DANN fails both in non-pre-trained and pre-trained regimes. This degenerate case has already been illustrated in Figure 3. On PACS transfers, we observe that using DANN is very beneficial in the pre-trained case, with large increases in target performance. However DANN still improves performance on average in the non-pre-trained regime.

In this section, we have confirmed that pre-trained representations usually allows to obtain features displaying a high degree of transferability. We have shown that pretraining alone is not always sufficient and is mainly useful if combined with source-only fine-tuning. Our synthetic experiment corroborates that pre-training should be considered as an inductive bias that helps with domain shifts similar to those found in real life, and fails in cases where confounding factors make this bias counter-effective.

4.4 Network architecture components

The update of features during source-only tuning is a complex, non-linear process that does not depend solely on the initialization of the feature encoder. To have a finer understanding of how transferable task-relevant features emerge, one must also take into account the architecture of the encoder. In this section, we conduct additional experiments to show how transfer learning can be sensitive to these architecture components.

The BatchNorm is mainly used to accelerate training of deep models: it rescales all activations in the effective range of the subsequent non-linearity, avoiding flat regions in the loss landscape on which gradient descent makes little progress. Furthermore, this multiplicative operation might give rise to descriptors that display higher degrees of invariance to certain transformations, for example variability in colorization or contrast. To illustrate this tendency to give rise to such invariant representations, we perform a very simple transfer, the purpose of which is to adapt MNIST to

its color-inverted counterpart, Inv-MNIST, and show in Table 5 that in the non-pretrained case, adding BatchNorm entirely conditions the success of domain adaptation for both source-only and DANN.

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

Table 5: Source-only accuracy for two different encoder architectures (with and without batch normalization); M=MNIST, InvM=MNIST with inverted colors

Global Pooling is an operation that collapses a whole feature map into a single vector by averaging across all spatial dimensions. The mean operator does not retain the original location of features and is therefore translation-invariant. Popular backbone architectures VGG-16, ResNet-18 or DenseNet have a 7x7 AvgPool layer. We conduct another transfer to evaluate the capacity to learn translation-invariance without being explicitly supervised to do so ; the source domain is MNIST, while the target domain is MNIST-T. MNIST-T is generated from MNIST samples, augmented by a random translation/scale/rotation. We report the performance of this transfer in Table 6. Results show, again, how a simple but appropriate architectural inductive bias such as Average Pooling can modify the extrapolation behavior of a model on an unseen target domain. Interestingly, when this positive bias is used, DANN brings further improvement over SO, while without global pooling, DANN performs even worth than SO.

Dropout Srivastava et al. (2014) is a simple yet very effective technique to regularize the capacity of neural networks by setting activations to zero with some probability p during training. This avoids co-adaptation of neurons and performs implicit model averaging. As dropout is ubiquitous in standard classification to increase generalization, we would like to measure its contribution to domain-invariance. To do so, we run all experiments of Table 4 again with and without dropout in the penultimate layer. We noticed that the impact of dropout was not the same depending whether pre-training is applied as well or not. We therefore gather in Table 7 a detailed sets of results (with/without dropout and with/without pretraining). Comparison with the non-dropout baseline shows a negligible impact in both non-pretrained and pre-trained cases.

M→ MT	SO	DANN
NoGlobalAvgPool	0.51	0.48
GlobalAvgPool	0.8	0.95

Table 6: Source only accuracies for two different encoder architectures; M=MNIST, MT=MNIST with random translation/scale/rotation augmentation

	no pretraining	
	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

	with pretraining	
	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 7: Target accuracy without / with dropout; **Left table:** Non-pre-trained **Right table:** Pre-trained; P=PACS-Photo, A=PACS-Art-Painting, Sk=PACS-Sketch, C=PACS-Clipart, M=MNIST, MM=MNIST-M, S=SVHN

5 Optimization of Inductive Biases

So far, we have looked for inductive biases controlled by a fixed set of hyperparameters. Although some of them, like pre-training, seem to provide a steady increase in target accuracy, there is no principled way to determine how such design choices should be chosen, adjusted and combined to address a given transfer in the general case. Moreover, there is no guarantee that fixed priors and assumptions produce an optimal solution.

In this section, we will explore several methods that add a degree of flexibility to the model by enabling optimization of parametric inductive biases. This optimization can be either heuristic or designed to minimize the actual objective (target risk). In the latter case, we fall in the meta-learning case.

5.1 Choosing Augmentation Parameters by Optimizing a Proxy Objective

In the usual domain adaptation experimental setting, the true objective cannot be computed nor optimized directly, as target labels are not available. Therefore, we must choose a proxy objective according to which our hyperparameters controlling the inductive bias can be optimized. The SDA method (Ilse et al., 2020) falls within this category: it aims at learning the augmentation parameters leading to the best adaptation between a pair of domains. Given a set of augmentation functions (e.g., random rotation, gaussian blur, color jitter, etc.), SDA greedily chooses the augmentation function that maximizes domain confusion. It does so by training a classifier to distinguish between the two augmented domains and select the augmentation for which the accuracy is minimal (i.e. domain confusion is highest). The process is then repeated with all the remaining augmentations, composed with the previous one and so on until a stopping criterion is met. At test time, we train the model on the source domain augmented with the final chain of chosen augmentations and evaluate the performance on the target domain.

To study the soundness of this proxy objective, we evaluate in Figure 4 the domain confusion induced by 4 augmentation types (random rotation, random color jitter, gaussian noise and random crop) on 5 transfers. For each experimental configuration (augmentation function, transfer), we gradually increase the parameter controlling the augmentation intensity from its minimal value

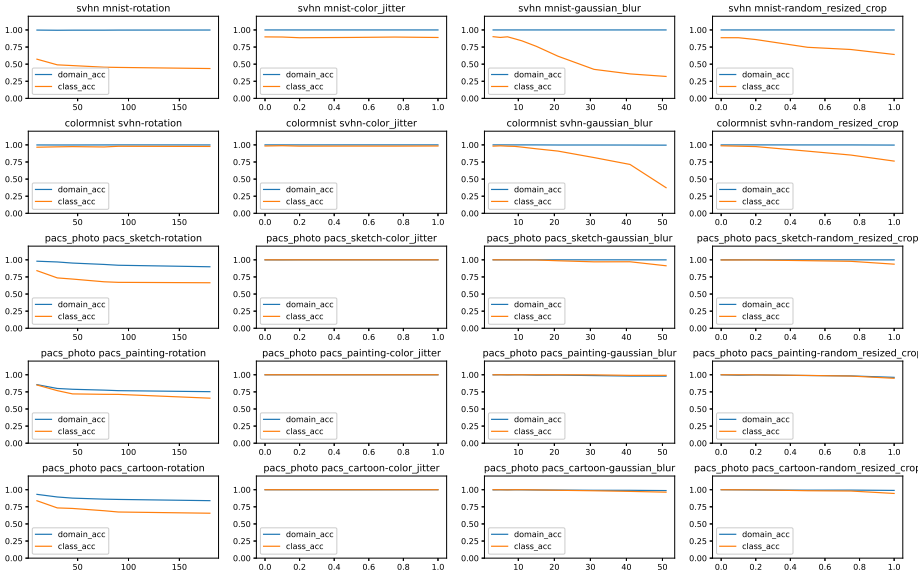

Figure 4: Increase in domain-confusion and class-confusion for a set of transfers and augmentations. Despite the increase of the augmentation parameter, a simple ResNet-18 model easily manages to distinguish the domains perfectly. As a sanity check, we also provide the source class as a measurement of image corruption.

(corresponding to the application of no augmentation) towards its maximum value. For example, in the case of rotations, we span from $\pm 0°$ to $\pm 180°$. For each degree of intensity, we report domain classification accuracy as a measure of confusion, and class accuracy on source as a measure of how the augmentation removes the useful class information.

Except for the pacs-photo to pacs-painting transfer, in which we observe a slight decrease in domain classification accuracy as the augmentation becomes too severe, the domains can be perfectly distinguished with a simple pre-trained ResNet-18. However, we observe a consistent drop in classification accuracy in most transfers.

To facilitate the emergence of domain confusion, we tried to weaken the domain classifier by using the older AlexNet architecture and SGD instead of Adam and report the results in Figures 5 and 6. We observed that for those regularized classifiers, domain classification remains trivial even on heavily corrupted images. However, the class accuracy is significantly lower than with ResNet-18.

Our conclusions on SDA is twofold: 1) the domain classification accuracy given by a deep model is not a reliable criterion for measuring confusion, as augmented source and target distributions do not overlap. 2) the heuristic chosen by SDA to choose the right set of augmentations do not have any consideration for the corruption of relevant class information, and hence does not lead to a dependable increase in target accuracy.

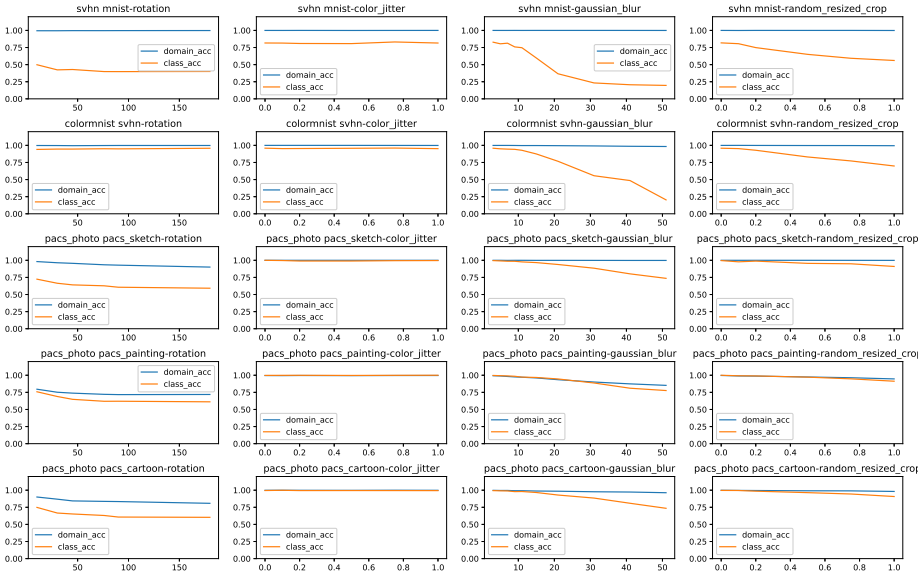

Figure 5: Same experiment than in figure 4, but with an AlexNet trained by Adam

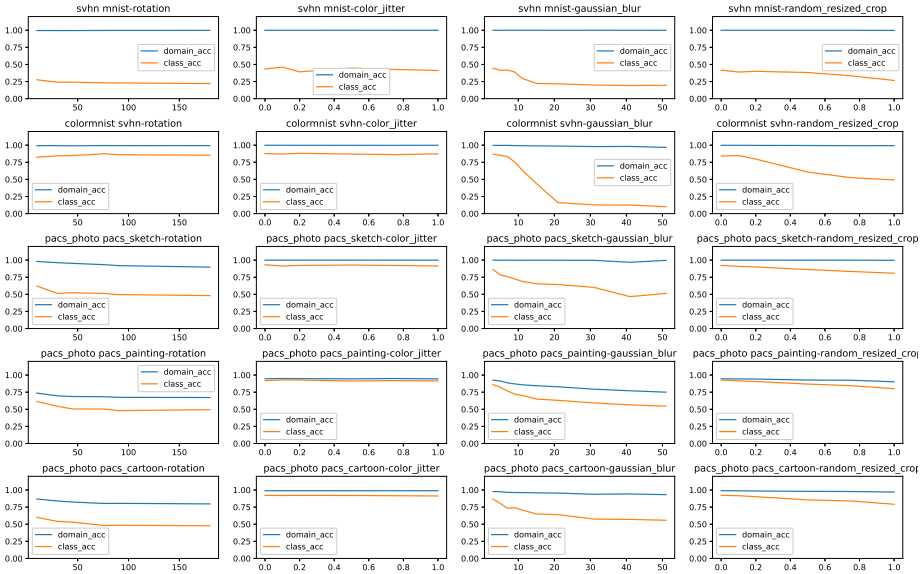

Figure 6: Same experiment than in figure 4, but with an AlexNet trained by SGD

5.2 Meta-Learning

5.2.1 Formalism and definition

Meta-learning, or "learning to learn", is a recent trend in machine learning that aims at learning the inductive bias from data itself, by directly optimizing the purpose it should serve. In this section, we propose to meta-learn a parametric inductive bias that helps better transfer some given domain shift, namely the initial weights of the network.

We define a task $\mathcal{T}$ as a N-way classification problem with its corresponding training set $\mathcal{T}^{train}$ and test set $\mathcal{T}^{test}$. A meta learning model defines an inner learning procedure that aims at solving any task $\mathcal{T}$ sampled from a task distribution $p(\mathcal{T})$, which means that, on average, for any $\mathcal{T} \sim p(\mathcal{T})$, it should display a good performance on the test set after training itself on the training set. Indeed, tasks from $p(\mathcal{T})$ are assumed to share some common information to justify the learning of a common inductive bias. A meta-learning model is defined by two sets of parameters θ and Φ:

- θ, also called "fast weight" contains the parameters that are specific to the learning of a task $\mathcal{T}_i$ provided on-the-fly, That means that it is authorized to be trained into θ_i after observing the task training data $\mathcal{T}_i^{train}$. θ_i is obtained during an imposed heuristic called "inner-loop training". In practice, we may initialize θ from only Φ and/or a random source of information and then train θ into θ_i.
- Φ is the task-agnostic parameter, also called "meta parameter" or "slow weight" : it embeds the common information shared among all $\mathcal{T} \sim p(\mathcal{T})$. It is not allowed to change when a specific $\mathcal{T}_i$ is provided, but it is meta-optimized to a fixed value that maximizes expected performance of (θ_i, Φ) on every $\mathcal{T}_i$. This process is called "meta-optimization" or "outer-loop training".

We can sum-up the key principles of meta-learning by the following two equations :

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

Any prototype of meta-learning model can hence be defined by a parametrization of θ, Φ, some functional *innerloop* and a loss function $\mathcal{L}$. We expect a clever meta-learning design to beat on average any heuristic optimization/regularization on $\mathcal{T}_i$ once meta-trained. However, to demonstrate its usefulness, a meta-learning method must be trained on tasks $\mathcal{T}_i$ that are different from test tasks. We therefore define two sets of tasks that are mutually exclusive, called "meta-training" and "meta-test" sets. We hope that some $\Phi^\star$ that is optimal for the meta-train set will also optimal on the meta-test set. Similarly to standard machine learning, there is a risk of "meta-overfitting" (Rajendran et al., 2020) on meta-training tasks that we must mitigate, for example by increasing the diversity of meta-training tasks, or by choosing a clever parametrization of θ and Φ.

5.2.2 Application of Mela-Learning to Domain Adaptation

Early works on meta-learning focused on few-shot learning problems, which is an extreme case of standard generalization. However, meta-learning is an extremely flexible framework that we can apply to any scenario, including domain adaptation.

We propose to use meta-learning to maximize the target-domain performance of a "Source-Only" training. In other words, for any task $\mathcal{T}_i$, we set $\mathcal{T}_i^{train} = S$ and $\mathcal{T}_i^{test} = T$. During its

inner loop, the model will hence optimize θ_i by exploiting samples from the source domain, and will be meta-optimized so that the tuple (Φ, θ_i) performs well on the same classification task, but on the target domain. We favor the Source-Only setting for its simplicity and flexible experimental conditions. Indeed, it does not require unlabeled target samples during inner-loop training. Note that several other meta-learning methods for domain adaptation do not use the Source-only setting for their inner loop (Wei et al., 2021). We now describe in detail the two parts of our meta-learning method.

Inner loop : We use the same parametrization than MAML (Finn et al., 2017) : the goal is to meta-train the initialisation θ^0 of a chosen neural network architecture, so that K gradient descent steps from this initialization computed on $\mathcal{T}_i^{train}$ lead to a final parameter $\theta^K = \theta_i$ that performs well on the target domain.

To catch up with the formalism presented in part 5.2.1, we can consider the equivalence $\Phi = \theta^0$ and $\theta_i = \theta^K$.

Outer loop optimization : To optimize θ^0 into $\theta^{0\star}$, we unroll the full graph of derivatives of *innerloop*, evaluate the meta-objective, compute the gradient of this objective w.r.t. θ^0 and finally perform a gradient descent step on θ^0. We repeat this process until convergence to $\theta^{0\star}$.

Algorithm 1 MAML-2DOM

Require: γ: inner learning rate, η: outer learning rate, S source domain, T target domain, Y_{MTrain} set of all meta-training classes
  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**(S, Y_{T_i})
    $T_i^{test} \sim$ **get_samples_of_said_classes**(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

Definition of meta-train and meta-test sets : Until now, our training loop is identical to the one proposed by Li et al. (2018). However, we still have to define exactly what will be our meta-train set and, most importantly, to which meta-test set of tasks we expect our learned inductive bias to generalize. There are several possibilities :

- The algorithm of Li et al. (2018) solves a problem of multi-source domain generalization with meta-learning. In this setting, we have $P-1$ source domains and only 1 target domain. In this case, to build a meta-train task, we select 2 source domains among the $P-1$, and use them as $\mathcal{T}^{train}$ and $\mathcal{T}^{test}$. Only one task is evaluated in meta test : it uses the union of the $P-1$ source domains as $\mathcal{T}^{train}$ and the unique target domain as $\mathcal{T}^{test}$. Note that the same classes are used during meta-training and meta-testing. In this case, meta-training is exploited as a means of cleverly merging the information from different source domains to better solve the same task on an unseen target domain.

- In our contribution, we address a totally different use-case. We have only two domains from

which we can sample tasks. In these domains, we have P meta-training classes and Q meta-testing classes. To build a meta-training task, we choose N classes among the P, and define $\mathcal{T}^{train}, \mathcal{T}^{test}$ as the sets of samples from domain 1 (resp. from domain 2) belonging to these classes. To build a meta-test task, we proceed similarly, but by picking among the Q meta-test classes instead. Here, the goal is to learn an inductive bias that implements the ability to transfer from one fixed domain to another fixed domain regardless of the classes encountered in the task. This has a strong connection with the discussions of section 4. We define this training and evaluation setting by the acronym **MAML-2DOM**.

The two nested training loops of MAML-2DOM are summarized in pseudo-code in Algorithm 1.

5.2.3 Experiments

Dataset: To satisfy the experimental setting described above, we use images from the VisDA dataset. VisDA is a dataset that includes 6 domains and subsequent domain shifts of varying difficulty : "Real", "Painting", "Sketch", "Quickdraw", "Clipart" and "Infograph". All domains contain the same 345 classes. Hence, making this dataset ideal to synthesize various transfer tasks. In our experiments, we define the first 200 classes as meta-training classes and the remaining 145 classes as meta-test classes. All tasks are 10-way classification problems.

Training MAML-2DOM: Given a pair of domains S and T chosen in VisDA, we meta-train an instance of MAML-2DOM to solve this transfer, and this transfer only. To do that, we generate tasks by sampling 10 classes among the 200 meta-train tasks, simulate inner source-only trainings from the source samples, measure the loss on target samples and minimize it w.r.t. the meta-parameter until convergence.

Implementation details: Training a meta-learning model can be costly in terms of memory and compute. Indeed, we must unroll and store the whole graph of derivatives at each realization of the inner loop. To save compute and memory, we propose to reduce considerably model size by avoiding working directly on image space. Indeed, doing so requires the use of a deep convolutional architecture, involving many operations and storing high-dimensional feature maps during the inner loop. We rather define the meta-model on top of a feature space produced by the pre-trained ResNet-18, that are assumed to be informative enough to solve our downstream tasks. We can therefore 1) pre-encode the whole VisDA database into features and 2) use a smaller model architecture : in our case a perceptron with two hidden layers.

Baselines: To demonstrate the superiority of our approach based on learned inductive bias, we must compare it to several reference baselines. Unless otherwise specified, these baselines also use the same architecture and work on the same pre-trained feature space.

- Random initialization (Random): A classifier initialized with the standard random initialization, that is then trained on source-domain samples of the 10-way task.
- Pre-training followed by fine-tuning (PT+SO): Random initialization might not be a fair baseline, as it could not exploit labeled information from the 200 meta-train classes from both domains. We hence propose to pre-train the classifier model to solve the 200-way classification problem for both meta-train domains simultaneously, then replace the last layer and fine-tune on the downstream 10-way test task.
- DANN: We perform the 10-way transfer with both labeled source images and unlabeled target images, we add the DANN adversarial constraint in the classifier to enhance transferability.

	Random+SO	Pretrain+SO	DANN	Pretrain+DANN	CAN	ASAN	MAML-2DOM
real→quickdraw	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
real→painting	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
real→sketch	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
real→clipart	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
real→infograph	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

Table 8: Average target accuracies and standard deviations for 10-way domain adaptation test tasks, computed over 10 runs; we outline best performance in bold

- Pre-training followed by DANN (PT+DANN): We combine pre-trained initialization and DANN training. Note that none of the transfers we build are prior-shifted as we use artificially balanced source and target batches, and that in all cases, we favor the baselines by choosing on it the best optimizer, learning rate and number of fine-tune iterations.
- CAN and ASAN (Kang et al., 2019; Raab et al., 2020) are state-of-the-art single-source domain adaptation methods. We use the default implementation provided by the authors : those methods hence work directly on image space. Note that when using their respective codebases, we nonetheless choose the pre-trained ResNet extractor and perform hyperparameter tuning to improve on our VisDA experiments.

We favor the baselines as much as possible by doing hyperparameter search on optimizer, learning rate and number of training iterations during fine-tuning.

Results : We study the transfers from "Real" towards any of the 5 other domains and display our results in table 8. Except for the transfer real→infograph, the corresponding MAML-2DOM beats all baselines by a large margin, especially on real→quickdraw, that is peculiarly difficult and on which the descriptors from ResNet-18 does not seem to transfer well, necessitating a correction from the subsequent classifier through a learned inductive bias. Concerning baselines, DANN generally provides an improvement of several percents compared to Source-Only both in the pre-trained and non-pre-trained cases. Those gains, however, are not comparable to those brought by MAML-2DOM. Last, pre-training the classifier on the full 200-way meta-train task does not seem to improve over random initialization.

6 Conclusion

In this paper, we pursued the analysis from Zhao et al. (2019); Johansson et al. (2019) to defend a meta-learning approach for unsupervised domain-adaptation. As a starting point, we consider the following question. How much of the success of deep domain alignment approaches can be unraveled through theoretical upper-bounds from domain adaptation theory. Despite their appealing prospects and their prescriptive significance in terms of modern approaches, a no-free-lunch theorem can be stated which invalidates a universal benefit of the domain alignment strategy. This surprising fact calls for a refined analysis of the key ingredients of successful domain alignment transfers.

We therefore investigate the role of various inductive biases to give a more appropriate account of the situation. We illustrate four kinds of inductive biases ranging from those inherent to the alignment approach itself to more generic ones such as the network architecture or data augmentation. In particular, the role of pretraining on a general purpose database such as imagenet is insidious

for several reasons. First, without this step, current alignment methods often fail which suggests that pre-training helps moving the supports of source and target distributions to a close match from which alignment methods can catch up. It appears nonetheless that the role of pretraining is sometimes more tangled as was evidenced by our kNN classifier experiment. In a nutshell, in such situations it is the combined effect of pretraining and source-driven optimization biases that are responsible for successful alignment.

The latter evidence motivated our search for efficient ways to design good inductive biases in a principled way. We have conducted an illustrative experiment in which we meta-learned parametric inductive biases that perform better than usual domain-adaptation heuristics on a given transfer. Given the impact of pre-training indicated by our former analysis, we proposed to meta-learn the initialization of the network. Although this choice revealed very effective in our set of experiments, it is not the only meaningful avenue. For instance, it is also possible to approach meta-learning on a regularization perspective or an optimization one. We hope to see more future work exploring this direction. On a closing note, we would like to point out that if our analysis sheds some light on the inner working of domain alignment, it is only in an empirical way: much remains to be done on the theoretical counterpart.

Acknowledgments

Research reported in this publication was supported by the Agence Nationale pour la Recherche (ANR) under award number ANR-19-CHIA-0017.

References

Y. Balaji, S. Sankaranarayanan, and R. 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.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016.

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

V. Bouvier, P. Very, C. Chastagnol, M. Tami, and C. 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.

C. Chen, W. Xie, T. Xu, W. bing Huang, Y. Rong, X. Ding, Y. Huang, and J. Huang. Progressive feature alignment for unsupervised domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 627–636, 2019.

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

R. T. des Combes, H. Zhao, Y. Wang, and G. J. Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *CoRR*, abs/2003.04475, 2020. URL `https://arxiv.org/abs/2003.04475`.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *ArXiv*, abs/1505.07818, 2016.

P. Germain, A. Habrard, F. Laviolette, and E. Morvant. Pac-bayesian theorems for domain adaptation with specialization to linear classifiers. *ArXiv*, abs/1503.06944, 2015.

P. Germain, A. Habrard, F. Laviolette, and E. Morvant. A new pac-bayesian perspective on domain adaptation. In *International conference on machine learning*, pages 859–868. PMLR, 2016.

B. Gong, Y. Shi, F. Sha, and K. 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.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

P. Häusser, T. Frerix, A. Mordvintsev, and D. Cremers. Associative domain adaptation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2784–2792, 2017.

M. Ilse, J. M. Tomczak, and P. Forr'e. Designing data augmentation for simulating interventions. *ArXiv*, abs/2005.01856, 2020.

F. D. Johansson, D. Sontag, and R. Ranganath. Support and invertibility in domain-invariant representations. *ArXiv*, abs/1903.03448, 2019.

G. Kang, L. Jiang, Y. Yang, and A. 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.

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

A. Kumar, P. Sattigeri, K. Wadhawan, L. Karlinsky, R. Feris, B. Freeman, and G. Wornell. Co-regularized alignment for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 31, 2018.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

X. Li and J. Bilmes. A bayesian divergence prior for classiffier adaptation. In M. Meila and X. 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.

M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

F. Lv, J. Liang, K. Gong, S. Li, C. H. Liu, H. Li, D. Liu, and G. Wang. Pareto domain adaptation. *arXiv preprint arXiv:2112.04137*, 2021.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

X. Peng, Z. Huang, X. Sun, and K. Saenko. Domain agnostic learning with disentangled representations. *ArXiv*, abs/1904.12347, 2019.

C. Raab, P. Vath, P. Meier, and F.-M. Schleif. Bridging adversarial and statistical domain transfer via spectral adaptation networks. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

J. Rajendran, A. Irpan, and E. Jang. Meta-learning requires meta-augmentation. *CoRR*, abs/2007.05549, 2020.

I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv: Learning*, 2020.

K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

J. Shen, Y. Qu, W. Zhang, and Y. Yu. Adversarial representation learning for domain adaptation. *ArXiv*, abs/1707.01217, 2017.

R. Shu, H. H. Bui, H. Narui, and S. Ermon. A dirt-t approach to unsupervised domain adaptation. *ArXiv*, abs/1802.08735, 2018.

L. Simon, J. Rabin, and R. Webster. Equivalence of several curves assessing the similarity between probability distributions. *ArXiv*, abs/2006.11809, 2020.

R. Siry, L. Simon, and F. Jurie. A study of alignment mechanisms in adversarial domain adaptation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1816–1820. IEEE, 2020.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958, 2014.

B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros. Unsupervised domain adaptation through self-supervision. *ArXiv*, abs/1909.11825, 2019.

E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

G. Wei, C. Lan, W. Zeng, and Z. 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.

X. Xu, X. Zhou, R. Venkatesan, G. Swaminathan, and O. 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.

Y. Zhang, T. Liu, M. Long, and M. Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413, 2019.

H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.