



Domain adversarial neural networks for domain generalization: when it works and how to improve

Anthony Sicilia¹ · Xingchen Zhao² · Seong Jae Hwang³ 

Received: 18 April 2022 / Revised: 2 January 2023 / Accepted: 22 February 2023
© The Author(s) 2023

Abstract

Theoretically, domain adaptation is a well-researched problem. Further, this theory has been well-used in practice. In particular, we note the bound on target error given by Ben-David et al. (Mach Learn 79(1–2):151–175, 2010) and the well-known domain-aligning algorithm based on this work using Domain Adversarial Neural Networks (DANN) presented by Ganin and Lempitsky (in International conference on machine learning, pp 1180–1189). Recently, multiple variants of DANN have been proposed for the related problem of domain *generalization*, but without much discussion of the original motivating bound. In this paper, we investigate the validity of DANN in domain generalization from this perspective. We investigate conditions under which application of DANN makes sense and further consider DANN as a dynamic process during training. Our investigation suggests that the application of DANN to domain generalization may not be as straightforward as it seems. To address this, we design an algorithmic extension to DANN in the domain generalization case. Our experimentation validates both theory and algorithm.

Keywords Domain adaptation · Domain generalization · Domain adversarial neural networks · Domain alignment

Editor: Lijun Zhang.

✉ Seong Jae Hwang
seongjae@yonsei.ac.kr

Anthony Sicilia
anthonysicilia@pitt.edu

Xingchen Zhao
zhao.xingc@northeastern.edu

¹ Intelligent Systems Program, University of Pittsburgh, 210 S. Bouquet Street, Pittsburgh, PA 15260, USA

² Department of Electrical and Computer Engineering, Northeastern University, 360 Huntington Ave, Boston, MA 02115, USA

³ Department of Artificial Intelligence, Yonsei University, Seoul, Korea

1 Introduction

In general, in machine learning, we assume the training data for our learning algorithm is well representative of the testing data. That is, we assume our training data follows the same distribution as our testing data. Of primary interest to this paper is the case where this assumption fails to hold: we consider learning in the presence of multiple *domains*. We formalize the multiple domain problem of interest as the case where (at train-time) we observe k domains referred to as *sources* which have distributions $\mathbb{P}_1, \mathbb{P}_2, \dots, \mathbb{P}_k$ over some space \mathcal{X} . At test-time, we are evaluated on a distinct *target* domain which has distribution \mathbb{Q} over \mathcal{X} . All of these feature distributions have (potentially) distinct labelling functions and our goal is to learn the labeling function on the target. Typically, we assume some restriction on observation of the target domain at train-time. In the literature, a large amount of work is concerned with the problem of *Domain Adaptation* (DA) which assumes access to samples from \mathbb{Q} , but restricts access to the labels of these samples. More recently, there has also been an active investigation on the problem of *Domain Generalization* (DG) which instead assumes absolutely *no* access to the target domain. In spite of these restrictions, in both cases, the goal is for our learning algorithm trained on *sources* to perform well when evaluated on the *target*.

One popular approach to DA is the use of a Domain Adversarial Neural Network (DANN) originally proposed by Ganin and Lempitsky (2015). Intuitively, this approach attempts to align the source and target domains by learning feature representations of both which are indiscernible by a domain discriminator trained to distinguish between the two distributions. Informally speaking, this seems like a sensible approach to DA. By accomplishing this *domain alignment*, the neural network should still be adept at the learned task when it is evaluated on the target domain at test-time. While DANN was originally proposed for DA, the adoption of this reasoning has motivated adaptations of this approach for DG (Albuquerque et al., 2020; Li et al., 2018b, c; Matsuura & Harada, 2020). In fact, very early works in DG (Muandet et al., 2013) are similarly motivated by the goal of domain-agnostic feature representations.

Still, it is worth noting that the original proposal of DANN (Ganin & Lempitsky, 2015) was motivated by theory. In particular, Ganin and Lempitsky base their algorithm on the target-error bound given by Ben-David et al. (2007, 2010a). Under appropriate assumptions, interpretation of the bound suggests domain alignment as achieved through DANN should improve performance on the target distribution, but importantly, it motivates alignment between the source and target. Counter to this, DANN variants for DG generally align multiple source domains because no access to target data is permitted. This shortcoming gives rise to the question of primary interest to this paper:

Is there a justification for source alignment using DANN in DG?

Specifically, we are concerned with a target-error bound similar to those provided by Ben-David et al. (2010a). To answer this question, we appeal to a recent theoretical proposal by Albuquerque et al. (2020) which uses a reference object (i.e., the set of mixture distributions of the sources) to derive a target-error bound in the domain generalization setting. Building on this framework, we provide answers to two important considerations:

1. What additional reference objects (besides sets of mixture distributions) satisfy the primary condition used to derive target-error bounds in DG?

2. How does the target-error bound behave as a *dynamic* quantity during the training process?

Ultimately, answering these two questions allows us to formulate a novel extension of the Domain Adversarial Neural Network. We validate experimentally that this extension improves performance and otherwise agrees with our theoretical expectations.

2 Domain Adversarial Neural Network (DANN)

In this section, we cover the necessary background on Domain Adversarial Neural Networks (DANN). We first present the original bound on target-error in the case of unsupervised DA (Ben-David et al., 2007, 2010a) which motivates the DANN algorithm proposed by Ganin and Lempitsky (Ganin & Lempitsky, 2015). Following this, we outline the key differences introduced by a DANN variant proposed by Matsuura and Harada (2020). Although this variant achieves state-of-the-art (DANN) performance in DG, we point out the main concerns we have regarding the justification of this approach.

2.1 In domain adaptation

As mentioned, we begin with a motivating result of Ben-David et al. (2010a). Intuitively, this result describes bounds on the target-error controlled, in part, by a computable measure of *divergence* between distributions. While we provide a more detailed exposition of the problem setup in Appendix A, we begin by listing here the key terms to familiarize the reader.

2.1.1 Setup

For a binary hypothesis h , a distribution \mathbb{P} , and a labeling function f for \mathbb{P} , we define the error $\mathcal{E}_{\mathbb{P}}(h)$ of h on the distribution \mathbb{P} as follows

$$\mathcal{E}_{\mathbb{P}}(h) = \mathbf{E}_{x \sim \mathbb{P}} |h(x) - f(x)| = \mathbf{E}_{x \sim \mathbb{P}} [1[h(x) \neq f(x)]] \quad (1)$$

This is our primary measure of the *quality* of a hypothesis when predicting on a distribution \mathbb{P} . To measure differences in distribution, we use the \mathcal{H} -divergence which is an adaptation of the \mathcal{A} -distance Kifer et al. (2004). In particular, given two distributions \mathbb{P}, \mathbb{Q} over a space \mathcal{X} and a corresponding hypothesis class $\mathcal{H} \subseteq \{h \mid h : \mathcal{X} \rightarrow \{0, 1\}\}$, the \mathcal{H} -divergence Ben-David et al. (2010a) is defined

$$d_{\mathcal{H}}(\mathbb{P}, \mathbb{Q}) = 2 \sup_{h \in \mathcal{H}} |\Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{Q}}(I_h)| \quad (2)$$

where $I_h = \{x \in \mathcal{X} \mid h(x) = 1\}$. Generally, it is more useful to consider the the $\mathcal{H}\Delta\mathcal{H}$ -divergence, specifically, where Ben-David et al. (2010a) define the symmetric difference hypothesis class $\mathcal{H}\Delta\mathcal{H}$ as the set of functions characteristic to disagreements between hypotheses.¹ This special case of the \mathcal{H} -divergence will be the measure of divergence in all considered bounds.

¹ Specifically, $g \in \mathcal{H}\Delta\mathcal{H} \Leftrightarrow g(x) = h_1(x) \oplus h_2(x) = |h_1(x) - h_2(x)|$ for $h_1, h_2 \in \mathcal{H}$

2.1.2 The motivating bound

We can now present the result of Ben-David et al. (2010a) based on the triangle inequality of classification error (Crammer et al., 2007; Ben-David et al., 2007). This bound is the key motivation behind DANN (Ganin & Lempitsky, 2015). For proof and a discussion on sample complexity, see Appendix A.

Theorem 1 (modified from Ben-David et al. (2010a), Theorem 2) *Let \mathcal{X} be a space and \mathcal{H} be a class of hypotheses corresponding to this space. Suppose \mathbb{P} and \mathbb{Q} are distributions over \mathcal{X} . Then for any $h \in \mathcal{H}$,*

$$\mathcal{E}_{\mathbb{Q}}(h) \leq \lambda + \mathcal{E}_{\mathbb{P}}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}) \quad (3)$$

with λ the error of an ideal joint hypothesis for \mathbb{Q}, \mathbb{P} .

This statement provides an upper bound on the target-error. Thus, minimizing this upper bound is a good proxy for the minimization of the target-error itself. The first term λ is a property of the dataset and hypothesis class which we typically assume to be small, but should not be ignored. As Ben-David et al. (2010a) note, this may be interpreted as a realizability assumption which requires the existence of some hypothesis in our search space that does well on both distributions (simultaneously). If this hypothesis does not exist, we cannot hope to do adaptation by minimizing the source-risk (Ben-David et al., 2010b). Notably, λ also plays an important role in algorithms like DANN which modify the distributions over which they learn since these algorithms implicitly change λ . We discuss this issue in detail in Sect. 2.3.

The latter terms are more explicitly controllable. The source-error $\mathcal{E}_{\mathbb{P}}(h)$ can be minimized as usual by Empirical Risk Minimization (ERM). The divergence can be empirically computed using another result of Ben-David et al. (2010a). While we give this result in the Appendix (Propositions 7 and 8, respectively), previous interpretation by Ganin and Lempitsky (2015) suggests to minimize the divergence by learning indiscernible representations of the distributions—i.e., aligning the domains.² As we describe in the following, this may be accomplished by maximizing the errors of a domain discriminator trained to distinguish the distributions.

2.1.3 The DANN algorithm

Ganin and Lempitsky (2015) separate the neural network used to learn the task into a *feature extractor* network r_{θ} and *task-specific* network c_{σ} , parameterized respectively by θ and σ . A binary domain discriminator d_{μ} outputting probabilities is trained to distinguish between the source and target distribution based on the representation learned by r_{θ} . Meanwhile, r_{θ} is trained to learn a representation that is not only useful for the task at hand, but also adept at “fooling” the domain discriminator (i.e., maximizing its errors). In details, given an empirical sample $\hat{\mathbb{P}} = (x_i)_{i=1}^n$ from the source distribution \mathbb{P} and a sample $\hat{\mathbb{Q}} = (x'_i)_{i=1}^n$ from the target distribution \mathbb{Q} , the domain adversarial training objective is described

² Note, the motivation of this representation learning is not entirely precise. In fact, this is the cause of the issues we discuss later in Sect. 2.3.

$$\min_{\mu} \max_{\theta} \frac{1}{2n} \sum_{i=1}^n \mathcal{L}_D(\mu, \theta, x_i, 0) + \mathcal{L}_D(\mu, \theta, x'_i, 1) \tag{4}$$

where

$$-\mathcal{L}_D(\mu, \theta, x, y) = (1 - y) \log(1 - d_{\mu} \circ r_{\theta}(x)) + y \log(d_{\mu} \circ r_{\theta}(x)). \tag{5}$$

By this specification, $d_{\mu} \circ r_{\theta}(x)$ is meant to estimate the probability x was drawn from \mathbb{Q} and \mathcal{L}_D represents the binary cross-entropy loss for a domain discriminator trained to distinguish \mathbb{P} and \mathbb{Q} . Combining this with a task-specific loss $\mathcal{L}_T^{\mathbb{P}}$ we get the formulation given by Ganin and Lempitsky (2015)

$$\min_{\sigma, \theta} \max_{\mu} \frac{1}{2n} \sum_{i=1}^n \mathcal{L}_T^{\mathbb{P}}(\sigma, \theta, x_i) - \frac{\lambda}{2n} \sum_{j=1}^n \mathcal{L}_D(\mu, \theta, x_j, 0) + \mathcal{L}_D(\mu, \theta, x'_j, 1) \tag{6}$$

where λ (in this context) is a trade-off parameter. The above is generally implemented by simultaneous gradient descent. We remark a solution to this optimization problem is easily approximated by incorporating a Gradient Reversal Layer (GRL) between r_{θ} and d_{μ} (Ganin & Lempitsky, 2015).

2.2 In domain generalization

Recent adaptations to the above formulation have been proposed in context of DG. Here, we focus on the proposal of Matsuura and Harada (2020) since their empirical results are one of the more competitive DG methods to date. In DG, since *no* access to \mathbb{Q} is given, one cannot actually compute \mathcal{L}_D as described above—it assumes at least unlabeled examples from \mathbb{Q} . Given this, Matsuura and Harada (2020) propose a modification which operates on k source samples

$$-\mathcal{L}_{SD}(\mu, \theta, x, y) = \sum_{i=1}^k \mathbb{1}[i = y] \log((d_{\mu} \circ r_{\theta}(x))_i) \tag{7}$$

where $\mathbb{1}[\cdot]$ is the indicator function. Now, d_{μ} is a multi-class domain discriminator trained to distinguish between *sources*; it outputs the estimated probabilities that x is drawn from each source. Hence, \mathcal{L}_{SD} is essentially a multi-class cross-entropy loss. Given the source samples $\hat{\mathbb{P}}_j = (x_i^j)_{i=1}^n \forall j \in [k]$ drawn respectively from the source distributions $\mathbb{P}_1, \mathbb{P}_2, \dots, \mathbb{P}_k$, we substitute this into Eq. (6):

$$\min_{\sigma, \theta} \max_{\mu} \frac{1}{kn} \sum_{i=1}^n \sum_{j=1}^k \mathcal{L}_T^{\mathbb{P}_j}(\sigma, \theta, x_i^j) + \frac{\lambda}{kn} \sum_{i=1}^n \sum_{j=1}^k \mathcal{L}_{SD}(\mu, \theta, x_i^j, j) \tag{8}$$

which gives a domain adversarial training objective aimed at aligning the sources (while also maintaining good task performance). Hereon, we often refer to this as a *source-source* DANN, rather than a *source-target* DANN as was given in Eq. (6). On the surface, there seems to be no justification for the source-source DANN. If we recall the interpretation of Theorem 1, there is one key difference: rather than aligning the source and target domains \mathbb{P} and \mathbb{Q} as suggested by the divergence term in Theorem 1, the objective in Eq. (8) aligns source domains \mathbb{P}_i and $\mathbb{P}_j \forall (i, j) \in [k]^2$ whose divergences do *not* appear in the upper

bound. Thus, the motivating argument is lost in this new formulation. If we look to recent literature, preliminary theoretical work to motivate this modification of DANN does exist (Albuquerque et al., 2020). We start from this work in the derivation of our own results.

2.3 A gap between theory and algorithm

To be totally precise, the algorithm given above does not actually minimize $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j)$ for any i, j . As we have noted, the idea to “align domains” through a common feature representation is simply an interpretation following the convention of Ganin and Lempitsky (2015). If the class from which we select d_μ is \mathcal{G} and the class from which we select r_θ is \mathcal{F} , the algorithm actually approximates minimization of $d_{\mathcal{G}\Delta\mathcal{G}}(\mathbb{P}_i \circ r_\theta^{-1}, \mathbb{P}_j \circ r_\theta^{-1})$ with respect to θ . Here, the notation $\mathbb{P}_i \circ r_\theta^{-1}$ denotes the pushforward of \mathbb{P}_i by r_θ which is (intuitively) the image of \mathbb{P}_i in the feature space. While this technicality will be unimportant for our discussions in the remainder of this text, it can potentially have significant negative ramifications. So, we discuss it in some detail here.

In particular, this gap between theory and algorithm implies that learning indiscernible representations of the source and target distributions while also minimizing the source error is not always sufficient for reducing the bound in Theorem 1. The problem arises because the ideal joint error (which is usually assumed small in the original problem) does not always *remain* small after feature transformation as in DANN. That is, while the ideal-joint error between \mathbb{P}_i and \mathbb{P}_j may be small, this may not be true of $\mathbb{P}_i \circ r_\theta^{-1}$ and $\mathbb{P}_j \circ r_\theta^{-1}$. This fact was recently observed independently by Johansson et al. (2019) and Zhao et al. (2019). Johansson et al. point out that learning a particular feature representation will always increase the ideal joint error (as compared to the original problem) whenever this feature representation is not invertible. Zhao et al. compliment this result by providing a lowerbound on target error in case the marginal label distributions³ have large deviation. In particular, the Jenson–Shannon (JS) divergence between the the label distributions should be at least as large as the JS divergence between the source and target feature distributions for the lowerbound to hold. If it is, the lowerbound shows simultaneous minimization of the source-error and the $\mathcal{H}\Delta\mathcal{H}$ -divergence actually increases target-error.

In practice, as we are aware, it is not clear to what extent non-invertible feature representations increase the ideal joint error. Further, it is not easy to test whether the JS-divergence of the label distributions is larger than the JS-divergence of the source and target feature distributions. For this reason, in this work, we will simply assume the ideal joint error remains small after feature transformation; i.e., we do not explicitly consider any settings in which there are negative ramifications of the known gap between theory and algorithm for DANN. If these issues are of significant concern for a particular application (i.e., if the marginal label shift is known to be large), a recent modification of DANN which uses importance weighting has been proposed by Tachet et al. (2020). This modification aims to correct the short-comings of standard DANN in case of label-shift. While we do not explicitly experiment with this method, our theoretical discussion and algorithmic extension still apply in context of this variation on DANN.

³ The marginal label distribution of the source or target is, formally, the pushforward of the source or target distribution by the respective labeling function.

3 Understanding domain alignment in domain generalization

Our discussion of source-source DANN for DG begins with the motivating target-error bound proposed by Albuquerque et al. (2020). Originally, given a set of source distributions $\{\mathbb{P}_i\}$, the bound uses the set of mixture distributions having these sources as components—we refer to this set as \mathcal{M} . Below, we consider a more general adaptation of this result. Although the proof strategy is largely similar, we do provide proof for this more general re-statement.

Proposition 2 (adapted from Albuquerque et al. (2020); Proposition 2) *Let \mathcal{X} be a space and let \mathcal{H} be a class of hypotheses corresponding to this space. Let \mathbb{Q} and the collection $\{\mathbb{P}_i\}_{i=1}^k$ be distributions over \mathcal{X} and let $\{\varphi_i\}_{i=1}^k$ be a collection of non-negative coefficients with $\sum_i \varphi_i = 1$. Let the object \mathcal{O} be a set of distributions such that for every $\mathbb{S} \in \mathcal{O}$ the following holds*

$$\sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) \leq \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j). \tag{9}$$

Then, for any $h \in \mathcal{H}$,

$$\begin{aligned} \mathcal{E}_{\mathbb{Q}}(h) &\leq \lambda_{\varphi} + \sum_i \varphi_i \mathcal{E}_{\mathbb{P}_i}(h) + \frac{1}{2} \min_{\mathbb{S} \in \mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q}) \\ &\quad + \frac{1}{2} \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j) \end{aligned} \tag{10}$$

where $\lambda_{\varphi} = \sum_i \varphi_i \lambda_i$ and each λ_i is the error of an ideal joint hypothesis for \mathbb{Q} and \mathbb{P}_i .

Proof Let $h \in \mathcal{H}$. For each \mathbb{P}_i apply Theorem 1 and multiply the equation by φ_i to achieve

$$\varphi_i \mathcal{E}_{\mathbb{Q}}(h) \leq \varphi_i \lambda_i + \varphi_i \mathcal{E}_{\mathbb{P}_i}(h) + \frac{\varphi_i}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}_i) \tag{11}$$

Taking $\lambda_{\varphi} = \sum_i \varphi_i \lambda_i$, we may sum over all k of these inequalities as below

$$\sum_i \varphi_i \mathcal{E}_{\mathbb{Q}}(h) \leq \lambda_{\varphi} + \sum_i \varphi_i \mathcal{E}_{\mathbb{P}_i}(h) + \frac{\varphi_i}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}_i). \tag{12}$$

Since $\sum_i \varphi_i = 1$ we can rewrite this as

$$\mathcal{E}_{\mathbb{Q}}(h) \leq \lambda_{\varphi} + \sum_i \varphi_i \mathcal{E}_{\mathbb{P}_i}(h) + \frac{1}{2} \sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}_i). \tag{13}$$

Now, for each \mathbb{P}_i , the following is true because the \mathcal{H} -divergence abides by the triangle inequality

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}_i) \leq d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{P}_i) \tag{14}$$

where

$$\mathbb{S}^* \in \arg \min_{\mathbb{S} \in \mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}). \tag{15}$$

Since this is true for each \mathbb{P}_i , we may write

$$\begin{aligned}
\frac{1}{2} \sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}_i) &\leq \frac{1}{2} \sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + \frac{1}{2} \sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{P}_i) \\
&= \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + \frac{1}{2} \sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{P}_i) \\
&\leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + \frac{1}{2} \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j)
\end{aligned} \tag{16}$$

where the last inequality is due to the choice $\mathbb{S}^* \in \mathcal{O}$. Recalling \mathbb{S}^* is also a minimizer of $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \cdot)$ yields the result. \square

As suggested by Albuquerque et al. (2020), interpreting this result provides a reasonable motivation for the use of source-source DANN in DG. The *first term* is a convex combination of ideal-joint errors between each source and the target. As before, we assume this is small and remains small after feature transformation by r_θ when we apply DANN; i.e., recall Sect. 2.3. Later, we discuss some differences between the ideal-error terms we give in our bound and the ideal-error terms in the original bound of Albuquerque et al. (2020). The *second term* is a convex combination of the source errors. ERM on a mixture of the sources is appropriate for controlling this term. In both of the previous convex sums, the coefficients are assumed to be fixed, but arbitrary, replicating a natural data generation process where amounts of data from each source are not assumed. Ben-David et al. (2010a) model data arising from multiple sources in this way and provide generalization bounds as well. For the *third term*, when \mathcal{O} is fixed as the set of mixtures \mathcal{M} , Albuquerque et al. (2020) suggest this term demonstrates the importance of diverse source distributions, so that the unseen target \mathbb{Q} might be “near” \mathcal{M} . We extend this discussion later, showing how this term can change *dynamically* throughout the training process. The *final term* is a maximum over the source-source divergences. Application of the interpretation by Ganin and Lempitsky (2015)—to align domains through representation learning—motivates the suggestion of Matsuura and Harada (2020) to maximize the errors of a multi-class (source-source) domain discriminator. A more precise application might be to train all combinations of binary domain discriminator, but as Albuquerque et al. (2020) point out, this leads to a polynomial number of discriminators. As a practical surrogate, we opt to employ the best empirical strategy to date Matsuura and Harada (2020). Another option might be to instead use a collection of one-versus-all classifiers in place of a multi-class classifier Albuquerque et al. (2020). Note, neither method *precisely* minimizes Eq. (10), so we treat this as an implementation choice.

A remark on differences As mentioned briefly, a reader familiar with the original statement of Albuquerque et al. (2020) will notice two differences: (1) rather than limiting consideration to the set of mixtures \mathcal{M} , this statement holds for all sets \mathcal{O} which satisfy Condition (9) and (2) λ_φ is a different quantity for the ideal joint-error between \mathbb{Q} and $\{\mathbb{P}_i\}$.

On the latter point, rather than λ_φ , Albuquerque et al. (2020) use the following definition of the ideal joint error given by Zhao et al. (2018) as below

$$\lambda_* = \min_{h \in \mathcal{H}} \mathcal{E}_{\mathbb{Q}}(h) + \mathcal{E}_{\mathbb{S}^*}(h) \tag{17}$$

where $\mathbb{S}^* \in \mathcal{M}$ is the mixture distribution closest to \mathbb{Q} . As the original statement of Albuquerque et al. (2020) defines $\mathcal{O} = \mathcal{M}$, this definition is a perfectly reasonable choice. But, since our re-statement considers more general objects \mathcal{O} , we have removed this dependence on \mathcal{M} . As is visible in the proof, λ_φ does remove this dependence. In general, λ_* and λ_φ are

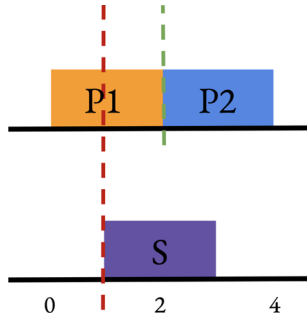


Fig. 1 A visualization of Example 1. Best viewed in color. The green line gives the value b of a hypothesis in $\{h_{a,b}(\cdot)\}_{(a,b)}$ with $a \leq 0$. Such a hypothesis would perfectly discern \mathbb{P}_1 and \mathbb{P}_2 . From this, it follows that $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_1, \mathbb{P}_2) = 2$ because a hypothesis in $\{h_{a,b}(\cdot)\}_{(a,b)}$ can achieve 2 and 2 is the maximum value for any divergence. Note, from this, it already follows that Eq. (18) holds because each term on the right-hand-side is bounded above by 2, and therefore, so is their convex combination. Still, we can analyze the example further. If we imagine the red line also gives the value b of a hypothesis in $\{h_{a,b}(\cdot)\}_{(a,b)}$ with $a \leq 0$ and slide it back and forth, we can never perfectly discern \mathbb{P}_1 or \mathbb{P}_2 from \mathbb{S} and therefore we will never achieve the maximum divergence 2

incomparable. If one attempts to compare them, it will become evident that some assumptions must be made—e.g., on the relationship between the $\{\varphi_i\}_i$ (which are arbitrary but fixed) and the coefficients used to form the mixture for \mathbb{S}^* (which are dependent on \mathbb{Q}). One reason to prefer λ_φ is that it does not require a single hypothesis to have low error on *all* sources simultaneously. Ben-David et al. (2010a) provide a larger discussion on the benefits of various approaches when combining data from multiple sources.

The former difference is of primary interest in this paper. Condition (9) may be considered to be the *key* fact about \mathcal{M} which allows the derivation of Eq. (10). By identifying this, we open the possibility of considering more general objects satisfying Condition (9). In the following, we demonstrate the existence of such objects \mathcal{O} and discuss the benefit they add.

3.1 Beyond mixture distributions

Consideration of general objects \mathcal{O} which satisfy Condition (9) is only useful if such objects exist (besides \mathcal{M}). The following example provides proof. See Fig. 1 for an illustrative picture.

Example 1 Let \mathcal{X} be the real line $(-\infty, \infty)$ and let \mathcal{H} be the set of hypotheses $\{h_a(\cdot)\}_{a \in \mathbb{R}}$ where $h_a(\cdot)$ is characteristic to the ray $(-\infty, a]$. Then, $\mathcal{H}\Delta\mathcal{H}$ is the set of hypotheses $\{h_{a,b}(\cdot)\}_{(a,b) \in \mathbb{R}^2}$ where $h_{a,b}(\cdot)$ is characteristic to the interval $[a, b]$. Let \mathbb{P}_1 be the uniform distribution $\mathcal{U}(0, 2)$, let \mathbb{P}_2 be $\mathcal{U}(2, 4)$, and let \mathbb{S} be $\mathcal{U}(1, 3)$. Then \mathbb{S} is not a mixture distribution of the components \mathbb{P}_1 and \mathbb{P}_2 , but

$$2 = \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j) \geq \sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) \tag{18}$$

for all non-negative coefficients $\{\varphi_i\}_i$ which sum to 1.

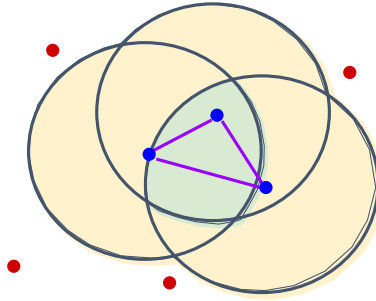


Fig. 2 An informal visualization. Blue dots represent sources. Purple lines define the boundaries of \mathcal{M} . Grey lines give the boundaries of the closed \mathcal{H} , ρ -balls around each source (defined in Proposition 3). Green colored areas define the boundary of $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$. Distributions within the yellow area may satisfy Condition (9). Distributions outside the yellow area (red dots) do not satisfy Condition (9) (Color figure online)

In the context of this example, we might consider the object $\mathcal{O} = \mathcal{M} + \{\mathbb{S}\}$ to quickly see that more than just \mathcal{M} can satisfy Condition (9). If \mathbb{S} is a unique minimizer of the third term in Eq. (10) and does not increase the final term, then using \mathcal{O} in place of \mathcal{M} actually produces a strictly tighter bound. Later we more generally expand on this and other benefits of considering $\mathcal{O} \neq \mathcal{M}$.

Still, one simple example cannot fully justify the existence of useful $\mathcal{O} \neq \mathcal{M}$. For a more general perspective, it is useful to think of things geometrically. Albuquerque et al. (2020) often refer to \mathcal{M} as the *convex-hull* of the sources. In this same vein, we point out that $d_{\mathcal{H}\Delta\mathcal{H}}$ is a *pseudometric*⁴ and therefore, shares most of the nice properties required of *metrics* used in the vast mathematical literature on metric spaces. Viewing a metric space as a *topological space*, it is common to think of open balls as the “the fundamental unit” or “basis” of the metric space. Loosely, borrowing this idea, we can define the (closed) \mathcal{H}, ρ -ball as below

$$\mathcal{B}_\rho(\mathbb{P}) = \{\mathbb{S} \mid d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{S}) \leq \rho\}. \quad (19)$$

Using this object, the following result provides some useful information on the types of objects \mathcal{O} which satisfy Condition (9). See Fig. 2 for a helpful visualization of our results.

Proposition 3 *Let \mathcal{X} be a space and let \mathcal{H} be a class of hypotheses corresponding to this space. Let the collection $\{\mathbb{P}_i\}_{i=1}^k$ be distributions over \mathcal{X} and let $\{\varphi_i\}_{i=1}^k$ be a collection of non-negative coefficients with $\sum_i \varphi_i = 1$. Now, set $\rho = \max_{u,v} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_u, \mathbb{P}_v)$. We show three results,*

1. $\mathcal{M} \subseteq \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$.
2. If $\mathbb{S} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$, then Condition (9) holds.
3. If $\mathbb{S} \notin \bigcup_i \mathcal{B}_\rho(\mathbb{P}_i)$, then Condition (9) fails to hold.

Proof We begin with a proof of (1). Let $\mathbb{S} \in \mathcal{M}$ arbitrarily. The result follows by first observing, for all \mathbb{P}_i ,

⁴ In Appendix A, we show the *commonly used* fact that $d_{\mathcal{H}\Delta\mathcal{H}}$ possesses a triangle-inequality. Symmetry and evaluation to 0 for identical distributions are easy to see.

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) \leq \sum_j \alpha_j d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j) \leq \rho. \tag{20}$$

The first inequality follows by a property of the \mathcal{M} shown by Albuquerque et al. (2020); for reference, we provide proof of this in Lemma 2 in the Appendix. The second inequality follows because ρ is defined as the largest source-source divergence. Now, if this is true for all \mathbb{P}_i , then \mathbb{S} is by definition contained in every \mathcal{H} , ρ -ball in the intersection $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$. If an element is contained in every component set of an intersection, then it is contained in the intersection. And, we have shown (1).

Next, we show (2). By definition of $\mathcal{B}_\rho(\mathbb{P}_i)$, if $\mathbb{S} \in \mathcal{B}_\rho(\mathbb{P}_i)$ then $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) \leq \rho$. Since $\mathbb{S} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ this is true for all $i \in [k]$. Then,

$$\sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) \leq \sum_i \varphi_i \rho = \rho. \tag{21}$$

We again recall that $\rho = \max_{u,v} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_u, \mathbb{P}_v)$. Hence, we have shown (2).

Finally, we demonstrate (3). To see this, note that if $\mathbb{S} \notin \bigcup_i \mathcal{B}_\rho(\mathbb{P}_i)$, then by definition for all i we have that $d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) > \rho$. We follow the chain of inequalities below to arrive at our result

$$\begin{aligned} & \sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{S}) \\ & > \sum_i \varphi_i \rho \\ & = \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j). \end{aligned} \tag{22}$$

Hence, we have shown (3) and are done. □

Statements 1 and 2 in conjunction show there are intuitive objects \mathcal{O} —i.e., $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ —which both contain \mathcal{M} and satisfy Condition (9). *Statement 3* provides an intuitive boundary for \mathcal{O} . Thus, comparison of \mathcal{O} to the union and intersection of closed balls, respectively, provides necessary and sufficient conditions for satisfying Condition (9).

3.2 The benefits of looking beyond mixtures

While the above discussion is useful in its own right, a more careful discussion of practical ramifications is needed.

Computationally tighter bounds First, we point out that different objects \mathcal{O} can lead to computationally tighter bounds in Eq. (10). For a concrete example, we prove $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ can lead to tighter bounds than \mathcal{M} below. The proof follows a similar logic as presented following Example 1. In fact, for Example 1, it is true that $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ contains $\mathcal{M} + \{\mathbb{S}\}$, and thus, may reap the discussed benefit.

Proposition 4 *Let \mathcal{X} be a space and let \mathcal{H} be a class of hypotheses corresponding to this space. Let \mathbb{Q} and the collection $\{\mathbb{P}_i\}_{i=1}^k$ be distributions over \mathcal{X} . Let \mathbb{P}^* be the distribution in $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ closest to \mathbb{Q} and let $\mathbb{S}^* \in \mathcal{M}$ be the mixture distribution closest to \mathbb{Q} . Then,*

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}^*, \mathbb{Q}) \leq d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{Q}). \tag{23}$$

Now, further, suppose the only solution to

$$\min_{\mathbb{P} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q}) \quad (24)$$

is contained in $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i) \setminus \mathcal{M}$. Then, we have

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}^*, \mathbb{Q}) < d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{Q}). \quad (25)$$

Proof To see the first claim, note by Proposition 3, $\mathcal{M} \subseteq \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$. So it is clear that

$$\min_{\mathbb{P} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q}) \leq \min_{\mathbb{S} \in \mathcal{M}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q}). \quad (26)$$

Since \mathbb{P}^* and \mathbb{S}^* are arguments minimizing left- and right-hand-side, respectively, we are done.

Now, we show the second claim. Equation 26 holds irregardless of our additional assumption, so we need only show that

$$\min_{\mathbb{P} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q}) \neq \min_{\mathbb{S} \in \mathcal{M}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q}). \quad (27)$$

But this is clear because if we assume the contrary—that the two quantities are equal—the implication is that a solution to Eq. 24 is contained in \mathcal{M} , a contradiction. Therefore, we have our result. \square

Now, for DANN, our hypothesis will usually be a neural network. In this case, the benefit of tightness may be considered irrelevant because the large VC-Dimension of neural networks (Bartlett et al., 2019) is the dominant term in any bound on error (i.e., using the PAC framework). Still, this conversation is not complete without considering the recent success of PAC-Bayesian formulations (e.g., see Dziugaite and Roy (2017)) which provide much tighter bounds when the hypothesis is a stochastic neural network. In Appendix A, we discuss a PAC-Bayesian distribution pseudometric (Germain et al., 2020) analogous to $d_{\mathcal{H}\Delta\mathcal{H}}$. Because this pseudometric shares the important properties of $d_{\mathcal{H}\Delta\mathcal{H}}$, these results are easily re-framed in this more modern formulation as well—where tightness may be a primary concern.

Intuitive analysis Second, we point out that a particular object \mathcal{O} can be easier to analyze. This fact will become evident as we develop an algorithmic extension to DANN for DG. Ultimately, we find that the novel object $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ may be manipulated to provide key motivating insights in algorithm design.

3.3 The $\mathcal{H}\Delta\mathcal{H}$ -divergence as a dynamic quantity

As mentioned, Albuquerque et al. (2020) interpret Proposition 2 as showing the necessity of diverse source distributions to control the third term $\min_{\mathbb{S} \in \mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q})$ when $\mathcal{O} = \mathcal{M}$. Logically, when distributions are heterogeneous, \mathcal{M} presumably contains more elements, and so, the unseen target is more likely to be “close.” When $\mathcal{O} = \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$, this is easier to see because the size of \mathcal{O} is directly dependent on the maximum divergence between the sources (by the definition of ρ). In particular, reducing the maximum divergence and

re-computing \mathcal{O} could lead to removal of a unique minimizer for $\min_{\mathbb{S} \in \mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q})$.⁵ In the context of the DANN algorithm, this is worrisome. Namely, during training, the point of using DANN is to effectively reduce the maximum divergence between sources and we expect this divergence to be decreasing as the feature representations of the source distributions are modified. In fact, under mild assumptions, we can formally show that DANN acts like a contraction mapping, and therefore, can *only* decrease the pairwise source-divergences. So, it is possible $\min_{\mathbb{S} \in \mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q})$ increases as the changing object \mathcal{O} shrinks during training. Below we consider gradient descent on a smooth proxy of the $\mathcal{H}\Delta\mathcal{H}$ -Divergence in the simple, two-distribution case. The map r_θ acts as the feature extractor affected by DANN.

Proposition 5 *Let \mathfrak{D} be a space of empirical samples over \mathcal{X} . Let $r_\theta : \mathcal{X} \rightarrow \mathcal{X}$ be a deterministic representation function parameterized by the real vector $\theta \in \mathbb{R}^m$. Further, denote by $r_\theta(\hat{\mathbb{P}})$ the application of r_θ to every point of $\hat{\mathbb{P}} \in \mathfrak{D}$. Fix $\hat{\mathbb{P}}, \hat{\mathbb{Q}} \in \mathfrak{D}$, let $\mathcal{L} : \mathfrak{D} \times \mathfrak{D} \rightarrow [0, \infty)$. Define $\ell(\theta) = \mathcal{L}(r_\theta(\hat{\mathbb{P}}), r_\theta(\hat{\mathbb{Q}}))$ and suppose it is differentiable with K -Lipschitz gradients. Further, suppose θ^* is the unique local minimum of ℓ on a bounded subset $\Omega \subset \mathbb{R}^m$. Then for $\theta \in \Omega$ such that $\theta \neq \theta^*$, the function $\tau : \Omega \rightarrow \mathbb{R}^m$ defined $\tau(\theta) = \theta - \gamma \nabla_\theta \ell(\theta)$ has the property*

$$\mathcal{L}(r_{\tau(\theta)}(\hat{\mathbb{P}}), r_{\tau(\theta)}(\hat{\mathbb{Q}})) \leq \beta_\theta \mathcal{L}(r_\theta(\hat{\mathbb{P}}), r_\theta(\hat{\mathbb{Q}})) \tag{28}$$

for some constant β_θ dependent on θ . In particular, for all $\theta \in \Omega$, there is γ so that $0 < \beta_\theta < 1$.

Proof We proceed by first showing an import inequality for functions ℓ with the assumed properties, in particular, using a derivation presented by Wright (2016). Note first, by Taylor’s Theorem, for vectors $u, v \in \mathbb{R}^n$, we have

$$\begin{aligned} \ell(u + v) &= \ell(u) + \int_0^1 \nabla \ell(u + \xi v)^T v \, d\xi \\ &= \ell(u) + \nabla \ell(u)^T v + \int_0^1 \nabla [\ell(u + \xi v) - \nabla \ell(u)]^T v \, d\xi \\ &\leq \ell(u) + \nabla \ell(u)^T v + \int_0^1 \|\nabla \ell(u + \xi v) - \nabla \ell(u)\| \|v\| \, d\xi \\ &\leq \ell(u) + \nabla \ell(u)^T v + \int_0^1 \xi K \|v\|^2 \, d\xi \\ &= \ell(u) + \nabla \ell(u)^T v + \frac{1}{2} K \|v\|^2. \end{aligned} \tag{29}$$

where the first line, as mentioned, is by Taylor’s Theorem, the second is by addition and subtraction of $\nabla \ell(u)^T v$, the third is because the norm of a vector product is never larger than the vector product, and the fourth is by the Lipschitz property assumed on the gradients of ℓ .

With this inequality, we let $\theta \in \Omega$ with $\theta \neq \theta^*$. Taking $u = \theta$ and $v = -\gamma \nabla \ell(\theta)$ achieves

⁵ Under conditions discussed later, the newly computed \mathcal{O} will be a subset and this unique minimizer might be absent in this subset.

$$\begin{aligned}\ell(\tau(\theta)) &\leq \ell(\theta) - \gamma \nabla \ell(\theta)^T \nabla \ell(\theta) + \frac{\gamma^2 K}{2} \|\nabla \ell(\theta)\|^2 \\ &= \ell(\theta) + \gamma \left(\frac{1}{2}\gamma K - 1\right) \|\nabla \ell(\theta)\|^2.\end{aligned}\quad (30)$$

Next, we note that for $\theta \neq \theta^*$ we have $0 \leq \ell(\theta^*) < \ell(\theta)$ because θ^* was assumed to be the *unique* local minimum of Ω . Then, we may set

$$\beta_\theta = 1 + \gamma \left(\frac{1}{2}\gamma K - 1\right) \frac{\|\nabla \ell(\theta)\|^2}{\ell(\theta)} \quad (31)$$

which, in combination with Eq. (30) yields our first desired result (Eq. (28)).

Next, we show that for all $\theta \neq \theta^*$, we can pick γ which forces $0 < \beta_\theta < 1$. We first note that it is sufficient to show

$$\frac{-\ell(\theta)}{\|\nabla \ell(\theta)\|^2} < \gamma \left(\frac{1}{2}\gamma K - 1\right) < 0 \quad (32)$$

since we may simply multiply by the reciprocal of the lower-bound and add one to realize the result. Next, we point out that there is some constant $M > 0$ such that $\|\nabla \ell(\theta)\| \leq KM$. This follows by

$$\|\nabla \ell(\theta)\| = \|\nabla \ell(\theta) - \nabla \ell(\theta^*)\| \leq K\|\theta - \theta^*\| \leq KM \quad (33)$$

where the equality holds because θ^* is a local minimum, the first inequality holds by the assumed Lipschitz property, and the second inequality holds because Ω was assumed to be bounded. Without loss of generality, suppose $M \geq 1$ (Eq. (33) holds regardless). Then our problem reduces further. In particular, it suffices to pick γ such that

$$\frac{-\ell(\theta)}{K^2 M^2} < \gamma \left(\frac{1}{2}\gamma K - 1\right) < 0 \quad (34)$$

since this lower bound is larger than or equal to that of Eq. (32). First, clearly, the upper bound holds when $0 < \gamma < \frac{2}{K}$, so this immediately restricts our choice of γ . For the lower bound, we consider two cases for the value of $\ell(\theta)$ and demonstrate there is γ with $0 < \gamma < \frac{2}{K}$ in both.

First, suppose $\ell(\theta) \geq \frac{1}{2}KM^2$. Then, if $\frac{2}{K} > \gamma > \frac{1}{K}$ we have

$$0 > \gamma \left(\frac{1}{2}\gamma K - 1\right) > \frac{-1}{2K} = \frac{-KM^2}{2K^2 M^2} > \frac{-\ell(\theta)}{K^2 M^2}. \quad (35)$$

Second, suppose $\ell(\theta) < \frac{1}{2}KM^2$. Then if γ is such that

$$\frac{2}{K} > \gamma > \frac{1 - \sqrt{1 - \frac{2\ell(\theta)}{KM^2}}}{K} \quad (36)$$

we have

$$\begin{aligned}
 & \gamma\left(\frac{1}{2}\gamma K - 1\right) + \frac{\ell(\theta)}{K^2M^2} \\
 & > \frac{K}{2} \left(\frac{1 - \sqrt{1 - \frac{2\ell(\theta)}{KM^2}}}{K} \right)^2 - \frac{1 - \sqrt{1 - \frac{2\ell(\theta)}{KM^2}}}{K} + \frac{\ell(\theta)}{K^2M^2} \tag{37} \\
 & = 0.
 \end{aligned}$$

Subtracting $\frac{\ell(\theta)}{K^2M^2}$ from both sides of this inequality yields the desired lower bound. Further, we still have $\gamma < \frac{2}{K}$, so the desired upper bound holds and we have our result.

Then, in any case, for each $\theta \neq \theta^*$, we can select γ so that $0 < \beta_\theta < 1$. □

A key takeaway from the above is the presence of competing objectives during training. These objectives require balance. While DANN reduces the source-divergences to account for the final term in Eq. (10), we should also (somehow) consider the diversity of our sources throughout training to account for the effected term $\min_{\mathbb{S} \in \mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q})$. Another insight the reader gains (i.e., from reading the proof) is that the upper bound on γ is constant and the lower bound goes to 0 as $\ell(\theta) \rightarrow 0$. An interpretation of these bounds suggests the practical importance of an annealing schedule on γ during DANN training. In our own experiments, we anneal γ by a constant factor (i.e., step decay).

4 An algorithmic extension to DANN

Motivated by the argument presented in Sect. 3.3, this section devises an extension to DANN. While DANN acts to align domains, as noted, its success in the context of domain generalization is also dependent on the heterogeneity of the source distributions *throughout* the training process. Therefore, in an attempt to balance these objectives, we propose an addition to source-source DANN which acts to diversify the sources throughout the training. Note, while the theoretical principles of our approach are certainly applicable to other feature matching methods in the literature (see Sect. 6), the implementation of the algorithm we devise in this section may be different (i.e., if the feature matching method is not based on loss-modification and gradient updates).

Theoretical motivation We recall the intersection of closed balls $\mathcal{O} = \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$; this is the main object of interest as it controls the size of the divergences in the upper bound of Proposition 2. More specifically, we are concerned with the quantity $\min_{\mathbb{P} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q})$. Intuitively, if we want to reduce this quantity we should find some means to increase ρ . One might propose to accomplish this by modifying our source distributions—e.g., through data augmentation –, but clearly, modifying our source distributions in an *uncontrolled* manner is not wise. This ignores the structure of the space of distributions under consideration and whichever distribution governs our sampling from this space – information that is, in part, given by our sample of sources itself. In this sense, while increasing ρ , we should preserve the structure of $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ as much as possible. Proposition 6 identifies conditions we must satisfy if we wish to increase ρ and modify our source distributions in a way that is *guaranteed* to reduce the third term of the upperbound in Eq. (10).

Proposition 6 Let \mathcal{X} be a space and let \mathcal{H} be a class of hypotheses corresponding to this space. Let \mathfrak{D} be the space of distributions over \mathcal{X} and let the collection $\{\mathbb{P}_i\}_{i=1}^k$ and the collection $\{\mathbb{R}_i\}_{i=1}^k$ be contained in \mathfrak{D} . Now, consider the collection of mixture distributions $\{\mathbb{S}_i\}_i$ defined so that for each set A , $\Pr_{\mathbb{S}_i}(A) = \alpha \Pr_{\mathbb{P}_i}(A) + \beta \Pr_{\mathbb{R}_i}(A)$. Further, set $\rho = \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{P}_j)$ and $\rho^* = \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}_i, \mathbb{S}_j)$. Then $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i) \subseteq \bigcap_i \mathcal{B}_{\rho^*}(\mathbb{S}_i)$ whenever $\rho^* - \beta \max_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{P}_i) \geq \rho$.

Proof Let $\mathbb{Q} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ be arbitrary. Then, by definition, for all i , we have that

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{Q}) \leq \rho. \quad (38)$$

Then, for all i , we have

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}_i, \mathbb{Q}) &\leq \alpha d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{Q}) + \beta d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{Q}) \\ &\leq \alpha \rho + \beta d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{Q}) \\ &\leq \alpha \rho + \beta d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{P}_i) + \beta d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}_i, \mathbb{Q}) \\ &\leq (\alpha + \beta)\rho + \beta d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{P}_i) \\ &= \rho + \beta d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{P}_i) \\ &\leq \rho + \beta \max_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{P}_i) \\ &\leq \rho^* \end{aligned} \quad (39)$$

where the first inequality follows by Lemma 2, the second inequality follows because $\mathbb{Q} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$ so the divergence is bounded by ρ for all i , the third inequality follows because, in general, the \mathcal{H} -divergence abides by the triangle-inequality, the fourth inequality follows again because $\mathbb{Q} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$, and the last inequality follows because we have assumed

$$\rho^* - \beta \max_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{P}_i) \geq \rho. \quad (40)$$

Now, this is true for all i , so by definition of $\bigcap_i \mathcal{B}_{\rho^*}(\mathbb{S}_i)$, we have that $\mathbb{Q} \in \bigcap_i \mathcal{B}_{\rho^*}(\mathbb{S}_i)$. Since \mathbb{Q} was an arbitrary element of $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$, we have shown $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i) \subseteq \bigcap_i \mathcal{B}_{\rho^*}(\mathbb{S}_i)$ and we have our result. \square

The above statement suggests that if we want to diversify our training distributions, we should train on a collection of modified source distributions $\{\mathbb{S}_i\}_i$. The modified distributions are mixture distributions whose components are pairs of our original source distributions $\{\mathbb{P}_i\}_i$ and new *auxiliary* distributions $\{\mathbb{R}_i\}_i$. The choice of $\{\mathbb{R}_i\}_i$ is constrained to guarantee the new intersection $\bigcap_i \mathcal{B}_{\rho^*}(\mathbb{S}_i)$ (with modified sources) contains the original intersection $\bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)$. Ultimately, this means we can guarantee $\min_{\mathbb{S} \in \bigcap_i \mathcal{B}_{\rho^*}(\mathbb{S}_i)} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q}) \leq \min_{\mathbb{P} \in \bigcap_i \mathcal{B}_\rho(\mathbb{P}_i)} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q})$.

Algorithm Empirically speaking, our modified source samples $\{\hat{\mathbb{S}}_i\}_i$ will be a mix of examples from the original sources $\{\mathbb{P}_i\}_i$ and the auxiliary distributions $\{\mathbb{R}_i\}_i$ —drawn from each proportionally to the mixture weights α and β . We plan to *generate* samples from the auxiliary distributions $\{\mathbb{R}_i\}_i$ and our interpretation of Proposition 6 suggests we should do so subject to the constraint below

$$\max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}_i, \mathbb{S}_j) - \beta \max_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{R}_i, \mathbb{P}_i) \geq \rho. \quad (41)$$

Because ρ is a property of our original dataset, it is independent of the distributions $\{\mathbb{R}_i\}_i$. This suggests that we should generate $\{\hat{\mathbb{R}}_i\}_i$ to maximize the left hand side. Maximizing this requires: (*Req.I*) maximizing the largest divergence between the new source samples $\{\hat{\mathbb{S}}_i\}_i$ and (*Req.II*) minimizing the largest divergence between our auxiliary samples $\{\hat{\mathbb{R}}_i\}_i$ and our original source samples $\{\hat{\mathbb{P}}_i\}_i$. Algorithmically, we can coarsely approximate these divergences, again appealing to the interpretation provided by Ben-David et al. (2010a) and Ganin and Lempitsky (2015): (*Req.I*) requires that our domain discriminator make fewer errors when discriminating the new source samples $\{\hat{\mathbb{S}}_i\}_i$ and (*Req.II*) requires that the auxiliary samples $\{\hat{\mathbb{R}}_i\}_i$ and the original sources $\{\hat{\mathbb{P}}_i\}_i$ be indiscernible by our domain discriminator.

To implement these requirements, we modify our dataset through gradient descent. Suppose that $\hat{\mathbb{P}}_i$ is an empirical sample from the distribution \mathbb{P}_i . We can alter data-points $a^j \sim \hat{\mathbb{P}}_j$ to generate data-points $b^j \sim \hat{\mathbb{R}}_j$ by setting $x^j(0) = a^j$ and iterating the below update rule to minimize \mathcal{L}_{SD} for T steps

$$x^j(t) \leftarrow x^j(t - 1) - \eta \nabla_x \mathcal{L}_{SD}(\mu, \theta, x^j(t - 1), j) \tag{42}$$

and then taking $b^j = x^j(T)$. *Importantly*, we do not modify the domain labels in this modification. So, our updates satisfy requirement (*Req.I*) because minimization of \mathcal{L}_{SD} approximates minimization of our domain discriminator’s errors, and further, satisfy (*Req.II*) because a_i and b_i are identically labeled, so minimization of the domain discriminator’s errors suggests that these examples should be indiscernible (i.e., assigned the same correct label).

While this update rule seemingly accomplishes our algorithmic goals, we must recall the final upper bound we wish to minimize (see Eq. (10)). The first two terms in this bound, λ_ϕ and $\sum_i \phi_i \mathcal{E}_{\mathbb{P}_i}(h)$, relate to our *classification* error—i.e., to the task-specific network c_σ . If our generated distributions $\{\hat{\mathbb{R}}_i\}_i$ distort the underlying class information, these terms may grow uncontrollably. To account for this, we further modify the update rule of Eq. (42) to minimize the change in the probability distribution output by the task classifier. We measure the change caused by our updates using the loss \mathcal{L}_{KL} —i.e., the KL-Divergence (Kullback, 1997). This gives the modified update rule

$$x_i^j(t) \leftarrow x_i^j(t - 1) - \eta \nabla_x \left[\mathcal{L}_{SD}(\mu, \theta, x_i^j(t - 1), j) + \mathcal{L}_{KL}(c_\sigma \circ r_\theta(x_i^j(0)), c_\sigma \circ r_\theta(x_i^j(t - 1))) \right]. \tag{43}$$

Algorithm 1 DANNCE (DANN with Cooperative Examples)

Input: A collection of mini-batches $\{x_i^j\}_i$ with labels $\{y_i^j\}_i$ for each $j \in [k]$, classifier c_σ parameterized by σ , feature extractor r_θ parameterized by θ , domain discriminator d_μ parameterized by μ

Parameters: Perturbation probability β between 0 and 1. Number of update steps T . Learning rate η .

Dependencies: SourceSourceDANN (i.e., any DANN algorithm to optimize for main-text Eq. 8 given a batch)

```

for  $j = 1 : k$  do
  for example  $x_\ell^j \in \text{batch } \{x_i^j\}_i$  do
     $p \leftarrow$  random uniform draw from  $[0, 1]$ 
    if  $p \leq \beta$  then
      for  $t = 1 : T$  do
        Update  $x_\ell^j(t)$  using Eq. (43) ▷ DANNCE Update
      end for
       $\{x_i^j\}_i \leftarrow \{x_i^j(0)\}_i - \{x_\ell^j(0)\} + \{x_\ell^j(T)\}$  ▷ Replace  $x_\ell^j(0)$ 
    end if
  end for
end for
SourceSourceDANN( $\{x_i^j\}_i, \{y_i^j\}_i, c_\sigma, r_\theta, d_\mu$ ) ▷ Run DANN as usual.
Return trained model components

```

Interpretation In totality, this algorithm may be seen as employing a style of *adversarial training* where, rather than generating examples to fool a task classifier—e.g., the single-source DG approach of Volpi et al. (2018), we instead generate examples to exploit the weaknesses of the feature extractor r_θ whose goal is to fool the domain discriminator. In this sense, the generated examples can be interpreted as *cooperating* with the domain discriminator. Hence, we refer to the technique as DANN with Cooperative Examples, or DANNCE. For details on our implementation of DANNCE please see the pseudo-code in Algorithm Block 1. Additional details can also be found in Appendix B.

5 Experimentation

In this section, we aim at addressing the primary point argued throughout this paper: the application of DANN to DG can benefit from (algorithmic) consideration of source diversity. While our theoretical discussion heavily focuses on convex hulls and \mathcal{H}, ρ -balls, we remind the reader that our theoretical results and algorithm are actually applicable to *any* distribution; these aforementioned geometric objects are only used as a theoretical reference to compare the target to the training data. So, since these geometric objects are challenging to compute for distributions, we instead validate our theoretical insights through the algorithm they produce. Namely, our modus operandi is comparison to recent state-of-the-art methods using a source-source DANN, or other domain alignment techniques, for

domain generalization. See Appendix B and code provided in supplement for all implementation details and additional experiments.

Datasets and hyper-parameters We evaluate our method on two multi-source DG datasets. (1) PACS (Li et al., 2017) contains 4 different styles of images (Photo, Art, Cartoon, and Sketch) with 7 common object categories. (2) Office-Home (Venkateswara et al., 2017) also contains 4 different styles of images (Art, Clipart, Product, and Real-World) with 65 common categories of daily objects. For both datasets, we follow standard experimental setups. We use 1 domain as target and the remaining 3 domains as sources. We report the average classification accuracy of the unseen target over 3 runs, using the model state at the last epoch to avoid peaking at the target. We select our hyper-parameters using leave-one-source-out CV (Balaji et al., 2018); this again avoids using the target in any way. Because some methods select parameters using a source train/val split, we use only the *training* data of the standard splits for fairness. Other parameters of our setup, unrelated to our own method, are selected based on the environment of Matsuura and Harada (Matsuura & Harada, 2020) (*MMLD*)—a SOTA source-source DANN technique. For full details, see Appendix B.

Our models For the feature extractor r_θ we use AlexNet (Krizhevsky et al., 2012) for PACS and ResNet-18 (He et al., 2016) for PACS and OfficeHome. Both are pretrained on ImageNet with the last fully-connected (FC) layer removed. For task classifier c_σ and domain discriminator d_μ we use only FC layers. For *ERM* (often called *Vanilla* or *Deep All*) only r_θ and c_σ are used and the model is trained on a mixture of all sources; this is a traditional DG baseline. For *DANN*, we add the domain discriminator d_μ and additionally update r_θ with \mathcal{L}_{SD} (see Eq. (8)). Because we ultimately compare against DANN as a baseline, we must ensure our implementation is state-of-the-art. Therefore, we generally follow the implementation described by Matsuura and Harada (2020), adding a commonly used Entropy Loss (Bengio et al., 1992; Shu et al., 2018) and phasing-in the impact of \mathcal{L}_{SD} on r_θ by setting $\lambda = 2/(1 + \exp(-\kappa \cdot p)) - 1$ in Eq. (8) with $p = \text{epoch}/\text{max_epoch}$ and $\kappa = 10$.

For our proposed method, *DANNCE*, we use the same baseline DANN, but update 50% of the images (i.e., $\beta = 0.5$) to *cooperate* with the domain discriminator following Eq. (43). The number of update steps per image is 5 (i.e., $T = 5$).

Experimental baselines As mentioned, we focus on comparison to other methods proposing domain alignment for DG. Albuquerque et al. (2020) (*G2DM*) and Li et al. (2018b) (*MMD-AAE*) propose variants of DANN,⁶ and in particular, align domains by making updates to the feature extractor. As noted, Matsuura and Harada (2020) (*MMLD*) propose the DANN setup most similar to our baseline DANN. For *MMLD*, Matsuura and Harada (2020) additionally propose a source domain *mixing* algorithm—we denote this by *MMLD-K* with K the number of domains after re-clustering. Shankar et al. (2018) (*CrossGRAD*) and Zhou et al. (2020) (*DDAIG*), contrary to our work, generate examples which *maximize* the domain loss. Because, they *do not* update the feature extractor with the domain loss \mathcal{L}_{SD} as we do, this may actually be viewed as domain-alignment by data generation (see Liu et al. (2019) who first propose this technique). For *MMD-AAE* and

⁶ *MMD-AAE* is based on the maximum-mean discrepancy (Gretton et al., 2012) rather than \mathcal{H} -divergence.

Table 1 PACS and OfficeHome results in accuracy (%)

PACS	Art	Cartoon	Sketch	Photo	Avg	Gain
<i>AlexNet</i>						
ERM	69.1	70.2	61.8	88.9	72.3	-
MMLD	68.5	72.2	66.3	89.3	74.1	1.8
MMLD-2	67.0	70.6	67.8	89.4	73.7	1.4
MMLD-3	69.3	72.8	66.4	89.0	74.4	2.1
ERM	64.9	70.2	61.4	90.0	71.6	-
G2DM	66.6	73.4	66.2	88.1	73.6	2.0
ERM	67.6	70.2	60.3	89.3	71.9	-
DANN	71.2	72.1	66.3	88.3	74.5	2.6
DANNCE	70.9	72.0	67.9	89.6	75.1	3.2
<i>ResNet-18</i>						
ERM	77.0	75.9	69.2	96.0	79.5	-
MMD-AAE	75.2	72.7	64.2	96.0	77.0	-2.5
CrossGRAD	79.8	76.8	70.2	96.0	80.7	1.2
DDAIG	84.2	78.1	74.7	95.3	83.1	3.6
ERM	78.3	75.0	65.2	96.2	78.7	-
MMLD-2	81.3	77.2	72.3	96.1	81.8	3.1
MMLD-3	79.6	76.8	71.2	95.9	80.9	3.1
G2DM	77.8	75.5	77.6	93.8	81.2	-
ERM	78.1	75.6	66.0	95.4	78.7	-
DANN	80.2	77.6	70.0	95.4	80.8	2.0
DANNCE	82.1	78.2	71.9	94.7	81.7	3.0
OfficeHome	Art	Clipart	Product	Real-w	Avg	Gain
ERM	58.9	49.4	74.3	76.2	64.7	-
MMD-AAE	56.5	47.3	72.1	74.8	62.7	-2.0
CrossGRAD	58.4	49.4	73.9	75.8	64.4	-0.4
DDAIG	59.2	52.3	74.6	76.0	65.5	0.8
ERM	60.0	49.0	75.4	76.8	65.3	-
DANN	61.6	48.9	75.8	76.2	65.6	0.3
DANNCE	61.6	50.2	75.6	75.9	65.8	0.5

avg: average of the target domain accuracies. *gain*: avg gain over the respective ERM (if reported)

CrossGRAD, we use results reported by Zhou et al. (2020) because the original methods do not test on our datasets.

Analysis of performance Generally, in DG, the comparison of performance is subjective across different experimental setups—a problem highlighted by a recent commentary on the experimental rigor of DG setups (Gulrajani & Lopez-Paz, 2020). As such, we include reported results from other experimental setups, predominantly, to show our DANN implementation is a competitive baseline. This much is visible in Table 1. For 2 out of 3 setups, our DANN *alone* has higher overall accuracy than any other method.

Our focus, then, is the validation of our main argument using our strong DANN baseline. In this context, shown in Table 1, ablation of DANNCE reveals substantial improvement



Fig. 3 Cooperative Examples (bottom) and corresponding original image after pre-processing (top) for PACS setup with target sketch and ResNet-18. Gradient updates appear to introduce relatively large changes in color hues/tints. Changes in image texture are also present (see Fig. 4)



Fig. 4 Cooperative Examples (bottom) and original images (top) magnified to illustrate change in image texture. Setup is identical to Fig. 3

upon the traditional source-source DANN in all PACS setups and (seemingly) marginal improvement in the OfficeHome setup. While performance improvements on OfficeHome may seem marginal, they actually present a reasonable improvement, since OfficeHome has a staggering 65 categories to classify compared to 7 in PACS.⁷ Ultimately, the performance gains demonstrated by addition of DANNCE agrees with our main argument: increasing diversity when aligning domains can have practical benefits in DG.

Analysis of loss curves To measure domain diversity, we use the loss of the domain discriminator (averaged per epoch). This loss is used to proxy the \mathcal{H} -divergence (an inverse relationship). A lower loss should then indicate more domain diversity, and, has the benefit of *dynamically* measuring diversity during training. Figure 5 shows the domain

⁷ To better understand this, we assume the distribution of classes is uniform and consider *gain* as a percentage of a uniformly random classifier's accuracy (RCA). This accounts for the difficulty of the problem. Then, for ResNet-18, the gain on OfficeHome would be 32.5% RCA and the gain on PACS would 21% RCA.

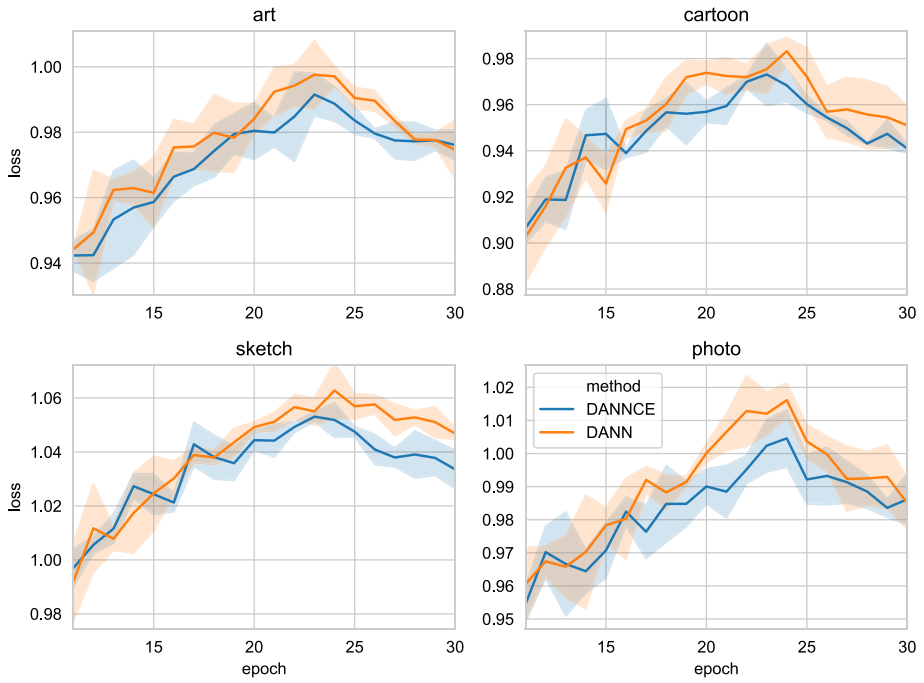


Fig. 5 Domain Discriminator Loss of DANN and DANNCE on PACS. For each target, we show the loss of its corresponding sources during training

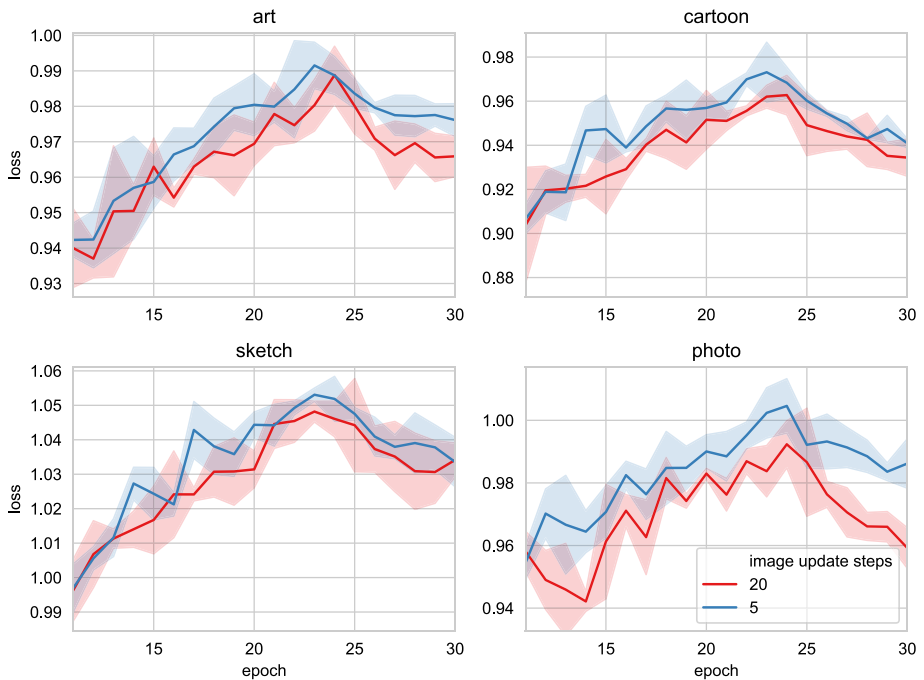


Fig. 6 Domain Discriminator Loss of DANNCE with 5 and 20 Steps of Image Updates

discriminator loss across epochs for our implementations of DANN and DANNCE using AlexNet on PACS. We generally see after epoch 15, the loss for DANNCE is lowest. Figure 6 further shows the effect of increasing the number of steps per image update. This suggests that increasing the number of updates has some control over the source domain diversity as intended. Finally, in both Figures, epochs 10 to 24 show the (inverted) smooth proxy for the domain divergence is increasing. This agrees with the formal claim made in Proposition 5. Although the trend changes after epoch 24, this is likely due to a decrease in γ at this epoch, and thus, does not necessarily disagree with our formal claim.

6 Related works

6.1 Domain adaptation theory

Many works extend the theoretical framework of Ben-David et al. (2010a) to motivate new variants of DANN. Zhao et al. (2018) consider the multi-source setting, Schoenauer-Sebag et al. (2019) consider a multi-domain setting in which all domains have labels but large variability across domains must be handled, and Zhang et al. (2019, 2020) consider theoretical extensions to the multi-class setting using a margin-loss. Besides the theoretical perspective of Ben-David et al. in DA, there are many other works to consider. Mansour et al. (2009) consider the case of general loss functions rather than the 01-error. Kuroki et al. (2019) consider a domain-divergence which depends on the source-domain and, through this dependence, always produces a tighter bound. Flamary et al. (2016) frame domain adaptation in terms of optimal transport. Many works also consider integral probability metrics including: Redko et al. (2017), Shen et al. (2018), and Johansson et al. (2019). As has been discussed in this paper, the assumptions of various domain adaptation theories are of particular importance. Consequently, these assumptions are also important for DG. We discuss some assumptions in more detail in the next part.

6.2 Assumptions in DA

Ben-David et al. (2010b) show control of their divergence term as well as the ideal joint error λ (so that both are small) give necessary and sufficient conditions for a large class of domain adaptation learners. These are the conditions which we control (in case of the divergence term) and assume (in case of the ideal joint error). Other assumptions for DA include the co-variate shift assumption in which the marginal feature distributions are assumed to change but the feature conditional label distributions across domains remain constant. As we have discussed, Zhao et al. (2019) show that this assumption is not always enough in the context of DANN and Johansson et al. (2019) provide similar conceptualizations. Still, this assumption can be useful in the context of model selection (Sugiyama et al., 2007; You et al., 2019). Another common assumption is label shift: the marginal label distributions disagree, but the label conditional feature distributions are the same. Again, this is related to the concern of Zhao et al. (2019) since significant disagreement in the label distributions can cause DANN to fail miserably. Lipton et al. (2018) provide adaptation algorithms for this particular situation. Another assumption one can make for the benefit of algorithm design is the notion of *generalized* label shift in which the label distributions may disagree and the feature conditional label distributions agree in an *intermediate* feature space. As we

have noted, Tachet et al. (2020) propose this assumption, devise new theoretical arguments under this assumptions, and suggest a number of algorithms based on their proposal.

6.3 Domain generalization theory

For DG, there is decidedly less theoretical work, but throughout our text, we have attempted to compare to the most relevant (and recent)—a bound proposed by Albuquerque et al. (2020). Albeit, some different theoretical perspectives on DG do exist. Li et al. (2020) consider the case where the feature conditional distribution of the target’s latent space is a linear combination of the sources; effectively moving the convex-hull concept to a learned feature-conditional latent space. Ye et al. (2021) consider the learnability of a DG problem, providing rigorous definitions of which problems one can expect to solve and which problems one cannot. The accompanying generalization bounds assume this definition of learnability, whereas bounds in our work do not. Instead, our bounds are applicable to *all* distributions and may be thought of as incorporating some idea of “learnability” into the bound itself via the hypothesis class and reference objects like the set of mixtures. In case the number of domains sampled may be larger, Blanchard et al. (2011, 2021) and Deng et al. (2020) consider domain generalization from the perspective of a meta-distribution which governs our observation of domains. Asymptotically, as we observe more domains, we can be more confident on the success of our algorithm. While this approach is interesting, our paper instead focuses on the case where we only have a relatively small number of domains from which to learn. In general, it is important to realize DG is a challenging problem where some assumptions must be made in order to provably guarantee the success of a learning algorithm. Different theoretical frameworks with different assumptions may be more or less applicable to different real-world problems.

6.4 Algorithms in DG

Besides DANN and other domain-aligning algorithms mentioned in this text, there are of course additional algorithmic perspectives on DG too. An early work in DG by Muandet et al. (2013) proposes a kernel-based algorithm aimed at producing domain-invariant features with a strong theoretical justification. More recently, a common thread is the use of meta-learning (e.g. to simulate domain-holdout) seen in Li et al. (2018a), Balaji et al. (2018), and Dou et al. (2019). Some authors, such as Wang et al. (2019) and Carlucci et al. (2019), make additional assumptions on the domains to be seen and use this in algorithm design. As mentioned, similar to our own algorithm, many works emphasize the importance of increasing the diversity of the source-data during training: Volpi et al. (2018), Albuquerque et al. (2020), Zhou et al. (2021), and Zhang et al. (2022). In addition in the distinctions present in algorithm design, our work also differs from these in its emphasis on the competing objectives this produces in feature-matching algorithms and accompanying theoretical analysis. Lastly, some works focus on the neural network components themselves, e.g., Li et al. (2017). These architecture changes can be very effective (see Seo et al. (2019) for impressive results when modifying batch-normalization). Related to our paper’s main point, we primarily focus on comparison to other methods proposing domain alignment for DG, especially those which are, in some sense, model agnostic. These additional references are discussed in *Experimental Baselines*.

7 Discussion

In this work, we investigate the applicability of source-source DANN for domain generalization. Our theoretical results and interpretation suggest a complex relationship between the heterogeneity of the source domains and the usual process of domain alignment. Motivated by this, we construct an algorithmic extension for DANN which diversifies the sources via gradient-based image updates. Our empirical results and analyses support our findings.

One of the motivations of our algorithm is also one of the predominant limitations of our study. In particular, the behavior of DANN as a dynamic process is not well understood. Studying it as such can reveal to us new information. For example, in the proof of Proposition 5, we saw the importance of annealing the learning rate for DANN. We also use Proposition 5 to motivate our algorithm design, but there are certainly open questions on the dynamic behavior of DANN and DANNCE. For example, it would be interesting to consider the competing objectives we have discussed in a more analytically tractable environment. Even for simple distributions, it is an open question as to how the hyper-parameters of DANNCE—which intuitively balance the competing objectives—may be optimally selected. On a related note, although we have assumed the ideal joint error is generally small, we have also pointed out that this is not always the case Zhao et al. (2019). While our promising results indicate this may not be an issue in practice, it is still interesting to consider this from a more theoretical perspective as well. Finally, it is important to point out that our empirical investigation was limited to images. It is interesting to consider how our technique might extend to natural language or other areas where gradient-based algorithms are used for learning.

Appendix A

On Theorem 1 in the main text

This section covers much of the theoretical background our work relies on in detail. Statements by Ben-David et al. (2007, 2010a) used to motivate the DANN algorithm (Ganin & Lempitsky, 2015) as well as statements on sample complexity (Kifer et al., 2004) are included.

Setup

We begin with a more detailed exposition of the setup assumed. We assume a space \mathcal{X} and a class of deterministic hypotheses $\mathcal{H} \subseteq \{h \mid \mathcal{X} \rightarrow \{0, 1\}\}$. In accordance with Ben-David et al. (2010a), for two functions h and f mapping from a space \mathcal{X} into the set $\{0, 1\}$, we define a disagreement measure with respect to a distribution \mathbb{P} over \mathcal{X} as below

$$\begin{aligned} \mathcal{E}_{\mathbb{P}}(h, f) &= \mathbf{E}_{x \sim \mathbb{P}} |h(x) - f(x)| \\ &= \mathbf{E}_{x \sim \mathbb{P}} [\mathbb{1}[h(x) \neq f(x)]] \end{aligned} \quad (44)$$

where $\mathbb{1}$ is the indicator function; i.e., $\mathcal{E}_{\mathbb{P}}(h, f)$ is the probability that h disagrees with f . If h is a hypothesis and f is a labeling function for \mathbb{P} which we would like to approximate by h , we call this term the error of h and write $\mathcal{E}_{\mathbb{P}}(h)$. We remark that the labeling function for \mathbb{P} need not be in \mathcal{H} . Further, we sometimes permit the labeling function f to have the continuous image $[0, 1]$ to capture the possibility of a non-deterministic label. Lastly, for each distribution \mathbb{P} , we write an empirical sample as $\hat{\mathbb{P}}$, and generally, specify its size.

We also recall from the main text, the measure considered in this paper is based on the \mathcal{H} -divergence which itself is an adaptation of the \mathcal{A} -distance Kifer et al. (2004). In particular, given two distributions \mathbb{P}, \mathbb{Q} over a space \mathcal{X} and a corresponding hypothesis class \mathcal{H} , the \mathcal{H} -divergence (Ben-David et al., 2010a) is defined

$$d_{\mathcal{H}}(\mathbb{P}, \mathbb{Q}) = 2 \sup_{h \in \mathcal{H}} |\Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{Q}}(I_h)| \quad (45)$$

where $I_h = \{x \in \mathcal{X} \mid h(x) = 1\}$. To arrive at the $\mathcal{H}\Delta\mathcal{H}$ -divergence, Ben-David et al. (2010a) define the symmetric difference hypothesis class $\mathcal{H}\Delta\mathcal{H}$. In particular, given a hypothesis class \mathcal{H} , the class $\mathcal{H}\Delta\mathcal{H}$ is the set of functions which are characteristic to disagreements between hypotheses. In details, we have

$$g \in \mathcal{H}\Delta\mathcal{H} \Leftrightarrow g(x) = h_1(x) \oplus h_2(x) = |h_1(x) - h_2(x)| \quad h_1, h_2 \in \mathcal{H}. \quad (46)$$

Therefore, the $\mathcal{H}\Delta\mathcal{H}$ -divergence is just a special case of the \mathcal{H} -divergence. As mentioned, this will be the measurement of divergence used in all considered bounds.

Computing the \mathcal{H} -divergence empirically

Here, we present Proposition 7. This result is an important consideration for the design of the DANN algorithm. In particular, both Ben-David et al. (2007, 2010a) and Ganin and Lempitsky (2015) suggest approximating the empirical $\mathcal{H}\Delta\mathcal{H}$ -divergence by training a classifier to distinguish between the source and target distributions. To minimize the empirical $\mathcal{H}\Delta\mathcal{H}$ -divergence, we should maximize this classifiers errors. Thus, this proposition can be viewed as motivation for our—and many other authors’—choice to approximate minimization of the divergence by maximization of a domain classifier’s errors.

Proposition 7 (Ben-David et al. (2010a) Lemma 2) *Provided a symmetric hypothesis class⁸ and samples $\hat{\mathbb{P}}, \hat{\mathbb{Q}}$ of size n*

$$\hat{d}_{\mathcal{H}}(\hat{\mathbb{P}}, \hat{\mathbb{Q}}) = 2 \left(1 - \min_{h \in \mathcal{H}} \left[\frac{1}{n} \sum_{x|h(x)=0} \mathbb{I}[x \in \hat{\mathbb{P}}] + \frac{1}{n} \sum_{x|h(x)=1} \mathbb{I}[x \in \hat{\mathbb{Q}}] \right] \right) \quad (47)$$

Proof We proceed in a similar fashion to Ben-David et al. (2010a). Let $h \in \mathcal{H}$ and consider the quantity

$$1 - \left[\frac{1}{n} \sum_{x|h(x)=0} \mathbb{I}[x \in \hat{\mathbb{P}}] + \frac{1}{n} \sum_{x|h(x)=1} \mathbb{I}[x \in \hat{\mathbb{Q}}] \right]. \quad (48)$$

We note two obvious facts. Every x must belong to the sample $\hat{\mathbb{Q}}$ or $\hat{\mathbb{P}}$ and every x must have $h(x) \in \{0, 1\}$. Therefore, we can rewrite $1 = \frac{2n}{2n}$ and we have

$$1 = \frac{1}{2n} \sum_{x|h(x)=0} \left(\mathbb{I}[x \in \hat{\mathbb{P}}] + \mathbb{I}[x \in \hat{\mathbb{Q}}] \right) + \frac{1}{2n} \sum_{x|h(x)=1} \left(\mathbb{I}[x \in \hat{\mathbb{P}}] + \mathbb{I}[x \in \hat{\mathbb{Q}}] \right) \quad (49)$$

By taking the common denominator $2n$, we may then write

⁸ A hypothesis class is symmetric if and only if for every $h \in \mathcal{H}$, we also have $1 - h \in \mathcal{H}$.

$$\begin{aligned}
 & 1 - \left[\frac{1}{n} \sum_{x|h(x)=0} \mathbb{I}[x \in \hat{\mathbb{P}}] + \frac{1}{n} \sum_{x|h(x)=1} \mathbb{I}[x \in \hat{\mathbb{Q}}] \right] \\
 &= \frac{1}{2n} \sum_{x|h(x)=0} \left(\mathbb{I}[x \in \hat{\mathbb{Q}}] - \mathbb{I}[x \in \hat{\mathbb{P}}] \right) + \frac{1}{2n} \sum_{x|h(x)=1} \left(\mathbb{I}[x \in \hat{\mathbb{P}}] - \mathbb{I}[x \in \hat{\mathbb{Q}}] \right).
 \end{aligned} \tag{50}$$

Now, for any sample $\hat{\mathbb{P}}$ of size n , we have

$$\Pr_{\hat{\mathbb{P}}}(I_h) = \frac{1}{n} \sum_{x|h(x)=1} \mathbb{I}[x \in \hat{\mathbb{P}}] \tag{51}$$

and

$$1 - \Pr_{\hat{\mathbb{P}}}(I_h) = \frac{1}{n} \sum_{x|h(x)=0} \mathbb{I}[x \in \hat{\mathbb{P}}]. \tag{52}$$

Therefore,

$$\begin{aligned}
 & \frac{1}{2n} \sum_{x|h(x)=0} \left(\mathbb{I}[x \in \hat{\mathbb{Q}}] - \mathbb{I}[x \in \hat{\mathbb{P}}] \right) + \frac{1}{2n} \sum_{x|h(x)=1} \left(\mathbb{I}[x \in \hat{\mathbb{P}}] - \mathbb{I}[x \in \hat{\mathbb{Q}}] \right) \\
 &= \frac{1}{2} (1 - \Pr_{\hat{\mathbb{Q}}}(I_h) - (1 - \Pr_{\hat{\mathbb{P}}}(I_h))) + \frac{1}{2} (\Pr_{\hat{\mathbb{P}}}(I_h) - \Pr_{\hat{\mathbb{Q}}}(I_h)) \\
 &= \Pr_{\hat{\mathbb{P}}}(I_h) - \Pr_{\hat{\mathbb{Q}}}(I_h).
 \end{aligned} \tag{53}$$

Following the chain of inequalities and taking a maximum on both sides, we therefore have

$$\max_{h \in \mathcal{H}} \Pr_{\hat{\mathbb{P}}}(I_h) - \Pr_{\hat{\mathbb{Q}}}(I_h) = 1 - \min_{h \in \mathcal{H}} \left[\frac{1}{n} \sum_{x|h(x)=0} \mathbb{I}[x \in \hat{\mathbb{P}}] + \frac{1}{n} \sum_{x|h(x)=1} \mathbb{I}[x \in \hat{\mathbb{Q}}] \right]. \tag{54}$$

Finally, to complete the proof, we note that

$$\Pr_{\hat{\mathbb{P}}}(I_h) - \Pr_{\hat{\mathbb{Q}}}(I_h) = \Pr_{\hat{\mathbb{Q}}}(I_{1-h}) - \Pr_{\hat{\mathbb{P}}}(I_{1-h}). \tag{55}$$

Since \mathcal{H} is assumed symmetric, we therefore have

$$\max_{h \in \mathcal{H}} \Pr_{\hat{\mathbb{P}}}(I_h) - \Pr_{\hat{\mathbb{Q}}}(I_h) = \max_{h \in \mathcal{H}} \left| \Pr_{\hat{\mathbb{P}}}(I_h) - \Pr_{\hat{\mathbb{Q}}}(I_h) \right| \tag{56}$$

and we are done. □

Proof of Theorem 1

Here, we present Theorem 2.1 of the main text (referenced in the Appendix as Theorem 1). We begin with a required Lemma for the final proof.

Lemma 1 (Ben-David et al. (2010a) Lemma 3) *Let \mathcal{X} be a space and \mathcal{H} a class of hypotheses corresponding to this space. Let \mathbb{P} and \mathbb{Q} be distributions over \mathcal{X} . Then for any hypotheses $h_1, h_2 \in \mathcal{H}$, we have*

$$|\mathcal{E}_{\mathbb{P}}(h_1, h_2) - \mathcal{E}_{\mathbb{Q}}(h_1, h_2)| \leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q}) \quad (57)$$

Proof We proceed in a similar fashion to Ben-David et al. (2010a). By definition of the \mathcal{H} -divergence we have

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q}) &= 2 \sup_{g \in \mathcal{H}\Delta\mathcal{H}} \left| \Pr_{\mathbb{P}}(I_g) - \Pr_{\mathbb{Q}}(I_g) \right| \\ &= 2 \sup_{h, h' \in \mathcal{H}} |\Pr_{\mathbb{P}}(\{x \mid h(x) \neq h'(x)\}) - \Pr_{\mathbb{Q}}(\{x \mid h(x) \neq h'(x)\})| \\ &= 2 \sup_{h, h' \in \mathcal{H}} |\mathbf{E}_{x \sim \mathbb{P}}[\mathbb{1}[h(x) \neq h'(x)]] - \mathbf{E}_{x \sim \mathbb{Q}}[\mathbb{1}[h(x) \neq h'(x)]]| \quad (58) \\ &= 2 \sup_{h, h' \in \mathcal{H}} |\mathcal{E}_{\mathbb{P}}(h, h') - \mathcal{E}_{\mathbb{Q}}(h, h')| \\ &\geq 2|\mathcal{E}_{\mathbb{P}}(h_1, h_2) - \mathcal{E}_{\mathbb{Q}}(h_1, h_2)|. \end{aligned}$$

Here, the second equality follows directly from the definition of $\mathcal{H}\Delta\mathcal{H}$, the fourth equality follows from Main Text Eq. 6, and the last inequality follows by properties of the supremum. \square

Using Lemma 1, we may present the proof of Theorem 1. Our statement is modified, omitting empirical quantities. We invite the reader to view Theorem 2 of Ben-David et al. (2010a) for the full result.

Proof We proceed in a similar fashion to Ben-David et al. (2010a). First, note the triangle inequality of classification error (Cramer et al., 2007; Ben-David et al., 2007) which states that given any labeling functions h_1, h_2, h_3 , we have

$$\mathcal{E}_{\mathbb{A}}(h_1, h_2) \leq \mathcal{E}_{\mathbb{A}}(h_1, h_3) + \mathcal{E}_{\mathbb{A}}(h_2, h_3). \quad (59)$$

Then, let $h \in \mathcal{H}$, let $\eta = \arg \min_{h \in \mathcal{H}} \mathcal{E}_{\mathbb{Q}}(h) + \mathcal{E}_{\mathbb{P}}(h)$, and let $f_{\mathbb{P}}, f_{\mathbb{Q}}$ be the true labeling functions of \mathbb{P}, \mathbb{Q} on \mathcal{X} , respectively. Given this, we have

$$\begin{aligned} \mathcal{E}_{\mathbb{Q}}(h) &= \mathcal{E}_{\mathbb{Q}}(h, f_{\mathbb{Q}}) \\ &\leq \mathcal{E}_{\mathbb{Q}}(\eta, f_{\mathbb{Q}}) + \mathcal{E}_{\mathbb{Q}}(\eta, h) \\ &\leq \mathcal{E}_{\mathbb{Q}}(\eta, f_{\mathbb{Q}}) + \mathcal{E}_{\mathbb{P}}(\eta, h) + |\mathcal{E}_{\mathbb{P}}(\eta, h) - \mathcal{E}_{\mathbb{Q}}(\eta, h)| \\ &\leq \mathcal{E}_{\mathbb{Q}}(\eta, f_{\mathbb{Q}}) + \mathcal{E}_{\mathbb{P}}(\eta, h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}) \quad (60) \\ &\leq \mathcal{E}_{\mathbb{Q}}(\eta, f_{\mathbb{Q}}) + \mathcal{E}_{\mathbb{P}}(\eta, f_{\mathbb{P}}) + \mathcal{E}_{\mathbb{P}}(h, f_{\mathbb{P}}) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}) \\ &= \mathcal{E}_{\mathbb{Q}}(\eta) + \mathcal{E}_{\mathbb{P}}(\eta) + \mathcal{E}_{\mathbb{P}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{P}). \end{aligned}$$

Here, the second inequality comes from considering both the cases $\mathcal{E}_{\mathbb{B}}(\eta, h) > \mathcal{E}_{\mathbb{A}}(\eta, h)$ and $\mathcal{E}_{\mathbb{A}}(\eta, h) > \mathcal{E}_{\mathbb{B}}(\eta, h)$, the third inequality comes from Lemma 1, and all other inequalities follow from the triangle inequality of classification error. \square

Sample complexity of Theorem 1

Here, we present Proposition 8. This Proposition contributes the main result required to derive generalization bounds for Theorem 1. Since Theorem 1 is modified from Theorem 2 of Ben-David et al. (2010a), we direct the reader to this proof for further details.

Remark on sample complexity In general, we choose to omit discussion of sample complexity from the main text. In the usual case, where \mathcal{H} is a class of neural networks, the VC Dimension (Vapnik, 1999) is usually larger,⁹ than the number of samples. As can be seen in the statement of Proposition 8, this causes problems in the interpretation and assumptions of the generalization bound. Despite this fact, Ganin and Lempitsky (2015) have shown that (empirically) this is a non-issue for application of DANN. With this said, some readers may rightly desire tighter bounds on empirical quantities when dealing with neural networks. Recently, some works have shown success in deriving much tighter bounds on empirical quantities (like error) for stochastic neural networks using the PAC-Bayesian framework Dziugaite and Roy (2017). Within the PAC-Bayesian framework, Germain et al. (2020) provide a distribution divergence pseudometric very similar to the $\mathcal{H}\Delta\mathcal{H}$ -divergence. As mentioned in the main text, the important property we use in our results is the pseudometric property, so we expect our results to hold in this more modern formulation as well.

In any case, we remark that generalization bounds can be derived for Theorem 1 and other results in this paper by application of the below statement. For a more detailed discussion, where empirical quantities are considered and generalization bounds are discussed in a variety of circumstances, we direct the reader to the original work of Ben-David et al. (2010a).

Proposition 8 (Ben-David et al. (2010a) Lemma 2; Kifer et al. (2004) Theorem 3.2) *Let \mathcal{X} be a space and \mathcal{H} be a class of hypotheses corresponding to this space with VC dimension d . Suppose \mathbb{P} and \mathbb{Q} are distributions over \mathcal{X} with corresponding samples $\hat{\mathbb{P}}$ and $\hat{\mathbb{Q}}$ of size n . Suppose $\hat{d}_{\mathcal{H}}(\hat{\mathbb{P}}, \hat{\mathbb{Q}})$ is the empirical \mathcal{H} -divergence between samples. Then, for any $\delta \in (0, 1)$ the following holds with probability at least $1 - \delta$*

$$d_{\mathcal{H}}(\mathbb{P}, \mathbb{Q}) \leq \hat{d}_{\mathcal{H}}(\hat{\mathbb{P}}, \hat{\mathbb{Q}}) + O\left(\sqrt{\frac{d \log(\frac{2n}{d}) + \log(\frac{4}{\delta})}{n}}\right) \quad (61)$$

On the $\mathcal{H}\Delta\mathcal{H}$ -divergence with comparison to a PAC-Bayesian distribution divergence

Here, we prove some useful facts about the $\mathcal{H}\Delta\mathcal{H}$ -divergence. In fact, these are the essential properties used to prove our formal claims in the main text. Most of these are known and have been used by other authors, but we restate and prove them here for completeness. One important point of this discussion is to demonstrate the relation between a second distribution divergence proposed by Germain et al. (2020) within the PAC-Bayes framework. As we will show, this PAC-Bayesian divergence exhibits the same properties. The

⁹ For NNs, it is $O(WL \log W)$ where W/L are the number of weights/layers (Bartlett et al., 2019).

consequence is that much of our formal discussion holds for this more modern divergence as well.

A nice property of mixture distributions

Below we provide a nice property of mixture distributions when considering their divergence. We are aware of variants of this result which have been observed by both Zhao et al. (2018) and Albuquerque et al. (2020) in derivation of their own bounds. We use this result in most of our proofs involving mixtures.

Lemma 2 *Let \mathcal{X} be a space and let \mathcal{H} be a class of hypotheses corresponding to this space. Let the collection $\{\mathbb{P}_i\}_{i=1}^k$ be distributions over \mathcal{X} . Now, suppose also that \mathbb{Q} is a mixture of the component distributions $\{\mathbb{P}_i\}_i$; that is, for any set A , we have $\Pr_{\mathbb{Q}}(A) = \sum_i \alpha_i \Pr_{\mathbb{P}_i}(A)$ with $\sum_i \alpha_i = 1$ and $\alpha_i \geq 0, \forall i$. Then, for any distribution \mathbb{P} , the following holds*

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q}) \leq \sum_i \alpha_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{P}_i). \quad (62)$$

Proof The result follows from the chain below

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{Q}) &= 2 \sup_{h \in \mathcal{H}\Delta\mathcal{H}} |\Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{Q}}(I_h)| \\ &= 2 \sup_{h \in \mathcal{H}\Delta\mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \sum_j \alpha_j \Pr_{\mathbb{P}_j}(I_h) \right| \\ &= 2 \sup_{h \in \mathcal{H}\Delta\mathcal{H}} \left| \sum_j \alpha_j \Pr_{\mathbb{P}}(I_h) - \sum_j \alpha_j \Pr_{\mathbb{P}_j}(I_h) \right| \\ &= 2 \sup_{h \in \mathcal{H}\Delta\mathcal{H}} \left| \sum_j \alpha_j \left(\Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{P}_j}(I_h) \right) \right| \\ &\leq 2 \sum_j \alpha_j \sup_{h \in \mathcal{H}\Delta\mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{P}_j}(I_h) \right| \\ &= \sum_j \alpha_j d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{P}, \mathbb{P}_j). \end{aligned} \quad (63)$$

Here, the results follow mostly by definition or arithmetic, but we highlight some exceptions. The third equality follows because the coefficients $\{\alpha_j\}_j$ sum to 1. The only inequality follows by application of the triangle inequality (for absolute values) and properties of the supremum. In particular, for any $h^* \in \mathcal{H}\Delta\mathcal{H}$, we have

$$\begin{aligned} \sum_j \alpha_j \sup_{h \in \mathcal{H}\Delta\mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{P}_j}(I_h) \right| &\geq \sum_j \alpha_j \left| \Pr_{\mathbb{P}}(I_{h^*}) - \Pr_{\mathbb{P}_j}(I_{h^*}) \right| \\ &\geq \left| \sum_j \alpha_j \left(\Pr_{\mathbb{P}}(I_{h^*}) - \Pr_{\mathbb{P}_j}(I_{h^*}) \right) \right| \end{aligned} \quad (64)$$

where the first inequality follows because the supremum is defined as an upper-bound and the second follows from the triangle inequality. But, the supremum is also specified as the *least* upper bound, so with

$$\sum_j \alpha_j \sup_{h \in \mathcal{H}\Delta\mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{P}_j}(I_h) \right| \geq \left| \sum_j \alpha_j \left(\Pr_{\mathbb{P}}(I_{h^*}) - \Pr_{\mathbb{P}_j}(I_{h^*}) \right) \right| \quad (65)$$

for any $h^* \in \mathcal{H}\Delta\mathcal{H}$, we may use the *least* upper bound property of $\sup_{h \in \mathcal{H}\Delta\mathcal{H}} \left| \sum_j \alpha_j \Pr_{\mathbb{P}_j}(I_h) - \sum_j \alpha_j \Pr_{\mathbb{P}_j}(I_h) \right|$ to observe that

$$\begin{aligned} & \sum_j \alpha_j \sup_{h \in \mathcal{H} \Delta \mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{P}_j}(I_h) \right| \\ & \geq \sup_{h \in \mathcal{H} \Delta \mathcal{H}} \left| \sum_j \alpha_j \Pr_{\mathbb{P}}(I_h) - \sum_j \alpha_j \Pr_{\mathbb{P}_j}(I_h) \right| \end{aligned} \tag{66}$$

affirming the third inequality. □

Triangle inequality

Below, we provide proof that the $\mathcal{H} \Delta \mathcal{H}$ -divergence abides by the triangle inequality. This result is used by many authors, although we have not seen proof. As noted in the main text, this (along with some other easy to verify properties) make the $\mathcal{H} \Delta \mathcal{H}$ -divergence a *pseudometric*. We use the triangle inequality a great deal throughout our proofs.

Proposition 9 *Let \mathcal{H} be and arbitrary class of hypotheses over \mathcal{X} . Then, the \mathcal{H} -divergence abides by the triangle-inequality.*

Proof Let \mathbb{P} , \mathbb{Q} , and \mathbb{S} be distributions over \mathcal{X} . We observe the following chain of inequalities

$$\begin{aligned} d_{\mathcal{H}}(\mathbb{P}, \mathbb{Q}) &= 2 \sup_{h \in \mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{Q}}(I_h) \right| \\ &= 2 \sup_{h \in \mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{S}}(I_h) + \Pr_{\mathbb{S}}(I_h) - \Pr_{\mathbb{Q}}(I_h) \right| \\ &\leq 2 \sup_{h \in \mathcal{H}} \left| \Pr_{\mathbb{P}}(I_h) - \Pr_{\mathbb{S}}(I_h) \right| + 2 \sup_{h \in \mathcal{H}} \left| \Pr_{\mathbb{S}}(I_h) - \Pr_{\mathbb{Q}}(I_h) \right| \\ &= d_{\mathcal{H}}(\mathbb{P}, \mathbb{S}) + d_{\mathcal{H}}(\mathbb{S}, \mathbb{Q}). \end{aligned} \tag{67}$$

The inequality follows from an argument similar to that used in defense of Lemma 2, Eq. 63. □

Comparison to the domain disagreement Germain et al. (2020)

The domain disagreement is another distribution divergence proposed by Germain et al. (2020). As noted by the these authors, the divergence is in fact designed as the PAC-Bayesian analog of the $\mathcal{H} \Delta \mathcal{H}$ -divergence. We define the *domain disagreement* below for a distribution ρ over \mathcal{H}

$$\text{dis}_{\rho}(\mathbb{P}, \mathbb{S}) = \left| \mathbf{E}_{(h, h') \sim \rho^2} \left[\mathcal{E}_{\mathbb{P}}(h, h') - \mathcal{E}_{\mathbb{S}}(h, h') \right] \right|. \tag{68}$$

As it turns out, the domain disagreement abides by a triangle-inequality and further satisfies Lemma 2. The former is a simple consequence of the fact that the domain disagreement is also a *pseudometric* (Germain et al., 2020). The latter is not so trivial to see, but we provide a quick sketch of the required steps below. Assuming \mathbb{S} is a mixture as in Lemma 2 we have

$$\begin{aligned}
\text{dis}_\rho(\mathbb{P}, \mathbb{S}) &= \left| \mathbf{E}_{(h,h') \sim \rho^2} [\mathcal{E}_{\mathbb{P}}(h, h') - \mathcal{E}_{\mathbb{S}}(h, h')] \right| \\
&= \left| \mathbf{E}_{(h,h') \sim \rho^2} \left[\mathcal{E}_{\mathbb{P}}(h, h') - \sum_i \alpha_i \mathcal{E}_{\mathbb{P}_i}(h, h') \right] \right| \\
&= \left| \mathbf{E}_{(h,h') \sim \rho^2} \left[\sum_i \alpha_i (\mathcal{E}_{\mathbb{P}}(h, h') - \mathcal{E}_{\mathbb{P}_i}(h, h')) \right] \right| \quad (69) \\
&= \left| \sum_i \alpha_i \mathbf{E}_{(h,h') \sim \rho^2} [\mathcal{E}_{\mathbb{P}}(h, h') - \mathcal{E}_{\mathbb{P}_i}(h, h')] \right| \\
&\leq \sum_i \alpha_i \left| \mathbf{E}_{(h,h') \sim \rho^2} [\mathcal{E}_{\mathbb{P}}(h, h') - \mathcal{E}_{\mathbb{P}_i}(h, h')] \right|.
\end{aligned}$$

The steps above generally follow from arithmetic similar to Eq. (63) or by linearity of the expectation. The last inequality uses properties of the absolute value.

Harking back to the main point, we remind the reader that Lemma 2 and the triangle-inequality the primary tools needed for our results. As such, much of our formal discussion holds for this more modern divergence as well.

Appendix B

We provide full details of our experiments and implementation. In the supplementary material package, *we also provide the fully functioning implementation of our approach*. Scripts for notable experimental setups and associated dataset links are provided for reproducibility. These will be publicly available on github upon acceptance.

Datasets

PACS PACS is a relatively new domain generalization dataset based on different styles of images (Li et al., 2017). Across 4 domains (Photo, Art Painting, Cartoon, and Sketch), there are 7 common object categories: dog, elephant, giraffe, guitar, horse, house, and person. There are a total of 9991 images. We split each domain into 90% for training set and 10% for validation set. The detail of PACS splits can be found in our codebase (`/paths/PACS/`).

OfficeHome Office-Home (Venkateswara et al., 2017) also contains 4 different styles of images (Art, Clipart, Product, and Real-W[orld]) with 65 common categories of daily objects. We use the down-sampled and preprocessed dataset curated by Zhou et al. Zhou et al. (2020) who propose DDAIG (compared to in the main text).

Training details

As mentioned in the main text, our setup closely follows Matsuura and Harada (2020). We implement our model in PyTorch (Paszke et al., 2019). Our model was trained by using 30 epochs, batch size 128, and SGD optimizer configured with momentum 0.9 and weight decay $5e-4$. We augment images using the same strategy as Matsuura and Harada (2020), employing color-jitter, horizontal flips, and cropping. For PACS (AlexNet and ResNet), the initial learning rate of the feature extractors are $1e-3$. The classifiers (i.e., task classifier and domain discriminator) have 10 times the initial learning rates of the corresponding feature extractor's initial learning rate since they are not pre-trained. We decrease all

learning rates by a factor of 10 after 24 epochs. All losses are weighted equally by default, but as mentioned, following Matsuura and Harada (2020), we phase-in the impact of \mathcal{L}_{SD} and the entropy loss on r_θ by using the weight $\lambda = 2/(1 + \exp(-\kappa \cdot p)) - 1$. For Office-Home, experimental settings are almost identical. We deviate only slightly by lowering the learning rate of the domain-discriminator by a factor of 10 and lowering the magnitude of \mathcal{L}_{SD} by a factor of 4 when updating the feature extractor.¹⁰

All experiments were run on an NVIDIA GeForce RTX 2080 Ti GPU 11GB. We used the helpful Weights and Biases tool (Biewald, 2020) during experimentation for visualizing our model training and results.

Network architectures

In this section, we provide details of the network architectures of our model components.

Feature extractors We use AlexNet (Krizhevsky et al., 2012) for PACS and ResNet-18 (He et al., 2016) for PACS and OfficeHome. In both cases, we pretrain on ImageNet (Deng et al., 2009) with the last FC layer removed. We note that we used a Caffe version of AlexNet implemented in PyTorch to follow related recent works (Carlucci et al., 2019; Matsuura & Harada, 2020) which showed consistently competitive Deep All baseline accuracies. The exact implementations can be found in our codebase (`/src/models/caffenet/models.py` and `/src/models/resnet.py`). The pretrained model for AlexNet is also included in our supplementary materials. For ResNet-18, it is loaded from `torchvision` in the code.

Classifiers The class classifier for is a single fully connected (FC) layer. The domain discriminator follows the design of Matsuura and Harada (2020) and is a simple stack of fully connected layers:

$$\text{FC}(1024) - \text{ReLU} - \text{DropOut}(0.5) - \text{FC}(1024) - \text{ReLU} - \text{DropOut}(0.5) - \text{FC}(\text{num_of_class}) \quad (70)$$

Again following Matsuura and Harada (2020), the class classifier has an xavier (glorot) uniform initialization (Glorot & Bengio, 2010) with gain set to 0.1, while the domain classifier uses the PyTorch (Paszke et al., 2019) default initialization (version 1.4). The exact implementation can be found in the code, specifically module `/src/models`.

Hyper-parameters of DANNCE

While we generally try to follow Matsuura and Harada (2020) as closely as possible to ensure our baseline DANN is state-of-the-art, we cannot use existing hyper-parameter choices for our novel algorithm (i.e., DANNCE). To perform the image updates (Line 5, Algorithm 1) we use the Adam optimizer (Kingma & Ba, 2014). Generally, we fix $\beta = 0.5$ and $t = 5$ in Algorithm 1. To maintain realistic image values, we clamp the pixel-values of the resulting images after each update based on the max and min pixel values of the PACS dataset. Yosinski et al. (2015) also use image-space gradient updates and further identify the addition of Gaussian blurring to be an important

¹⁰ Studying learning curves early-on during experimentation suggested that (1) the domain-discriminator was learning to quickly for the feature extractor and (2) the feature extractor was ignoring the classification loss and focusing on \mathcal{L}_{SD} as a result. These changes aimed to accommodate both of these observations and generally improved the learning curves.

Table 2 Ablation of Gaussian Blurring for DANNCE on PACS AlexNet

PACS	Art	Cartoon	Sketch	Photo	Avg	Gain
AlexNet						
DANN	71.2	72.1	66.3	88.3	74.5	-
DANN with BLUR	70.1	70.9	66.7	89.14	74.2	-0.3
DANNCE without BLUR	70.9	72.1	66.0	88.8	74.5	0.0
DANNCE	70.9	72.0	67.9	89.6	75.1	0.6

gain is computed with respect to the baseline DANN. For *DANN with BLUR*, we use the same blurring schedule as used in our DANNCE implementation, but make no image updates. For *DANNCE without BLUR*, we simply omit blurring from our DANNCE implementation. The results demonstrate two things: (1) blurring without image updates worsens performance against DANN and (2) DANNCE without blurring has no performance gain over DANN. This ablation tells us that while our performance gains are *not* coming from blurring itself, blurring may be important for DANNCE on image datasets. This agrees with findings by Yosinski et al. (2015) which suggests blurring suppresses high frequency information produced by gradient updates so that image statistics do not differ too much from the real dataset

parameter for producing realistic images. Based on one of the optimal settings described by Yosinski et al. (2015), we use Gaussian blurring once every 4 steps of our algorithm. We provide ablation of the effect of blurring in Table 2 which reveals that blurring may indeed be important for our method when applied to images, but *importantly* also shows that our gain in performance does not *only* come from blurring because blurring without our algorithm actually hurts performance of the baseline DANN.

We select the learning-rate and weight-decay of our Adam optimizer from the set { $1e-2$, $1e-2$, $1e-3$ } based on a leave-one-source-out cross validation used by Balaji et al. (2018). For example, when Art is the unseen target, we run DANNCE with every parameter setting holding out each of Cartoon, Photo, and Sketch as a *pseudo* target domain – the pair of parameters performing best on the simulated holdout task (averaged over the three *pseudo* targets) is used for a final training phase including all three sources and evaluated on Art. Note that this does not use the unseen target (e.g., Art) at all in the hyper-parameter selection, mimicking a real-life DG scenario. For PACS (AlexNet and ResNet-18), we run this leave-one-source-out cross validation. In the interest of time, for OfficeHome, we use the best parameter setting of PACS AlexNet (averaged over *all* pseudo targets) instead of performing a leave-one-source-out cross validation. Clearly, this still does not use any information from the unseen target. The exact parameter settings in every case may be found in the code within each experiment’s bash script (see directory `final_scripts`).

Author contributions AS, XZ, and SJH all contributed to the conception and design of the theoretical work and algorithm. AS wrote initial drafts of the manuscript and proofs. AS, XZ, and SJH edited the manuscript and proofs. AS planned and wrote the initial draft of the code base. XZ made edits to the code base and ran experiments. AS, XZ, and SJH interpreted results and planned experimentation.

Funding S. Hwang was supported by Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT), Artificial Intelligence Graduate Program, Yonsei University (2020-0-01361-003), and the Yonsei University Research Fund of 2022 (2022-22-0131).

Data availability statement This research does not contribute any new data. All data used is existing and publicly available.

Code availability Code to conduct experiments will be made publicly available upon publication.

Declarations

Conflict of interests Not applicable.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication All authors consent to submission and publication.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Albuquerque, I., Monteiro, J., Falk, T.H., & Mitliagkas, I. (2020). Adversarial target-invariant representation learning for domain generalization. arXiv preprint [arXiv:1911.00804](https://arxiv.org/abs/1911.00804)
- Balaji, Y., Sankaranarayanan, S., & Chellappa, R. (2018). Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems* (pp. 998–1008).
- Bartlett, P. L., Harvey, N., Liaw, C., & Mehrabian, A. (2019). Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *JMLR*, 20, 2285–2301.
- Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 137–144
- Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010a). A theory of learning from different domains. *Machine Learning*, 79(1–2), 151–175.
- Ben-David, S., Lu, T., Luu, T., & Pal, D. (2010b). Impossibility theorems for domain adaptation. In: Teh, Y. W., Titterton, M. (Eds.) *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research* (Vol. 9, pp. 129–136). PMLR, Chia Laguna Resort, Sardinia, Italy. <https://proceedings.mlr.press/v9/david10a.html>
- Bengio, S., Bengio, Y., Cloutier, J., & Gececi, J. (1992). On the optimization of a synaptic learning rule. In *Preprints Conference on Optimality in Artificial and Biological Neural Networks* (Vol. 2). University of Texas.
- Biewald, L. (2020). *Experiment tracking with weights and biases*. Software available from wandb.com <https://www.wandb.com/>
- Blanchard, G., Deshmukh, A. A., Dogan, Ü., Lee, G., & Scott, C. (2021). Domain generalization by marginal transfer learning. *J. Mach. Learn. Res.*, 22, 2–1.
- Blanchard, G., Lee, G., & Scott, C. (2011). Generalizing from several related classification tasks to a new unlabeled sample. *Advances in Neural Information Processing Systems*, 24, 2178–2186.
- Carlucci, F.M., D'Innocente, A., Bucci, S., Caputo, B., & Tommasi, T. (2019). Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2229–2238).
- Crammer, K., Kearns, M., & Wortman, J. (2007). Learning from multiple sources. In *Advances in Neural Information Processing Systems* (pp. 321–328).
- Deng, Z., Ding, F., Dwork, C., Hong, R., Parmigiani, G., Patil, P., & Sur, P. (2020). Representation via representations: Domain generalization via adversarially learned invariant representations. arXiv preprint [arXiv:2006.11478](https://arxiv.org/abs/2006.11478)
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *CVPR*.

- Dou, Q., de Castro, D. C., Kamnitsas, K., & Glocker, B. (2019). Domain generalization via model-agnostic learning of semantic features. In *Advances in Neural Information Processing Systems* (pp. 6447–6458).
- Dziugaite, G. K., & Roy, D. M. (2017). Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. [arXiv:1703.11008v2](https://arxiv.org/abs/1703.11008v2)
- Flamary, R., Courty, N., Tuia, D., & Rakotomamonjy, A. (2016). Optimal transport for domain adaptation (p. 1). *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning* (pp. 1180–1189).
- Germain, P., Habrard, A., Laviolette, F., & Morvant, E. (2020). Pac-bayes and domain adaptation. *Neuro-computing*, 379, 379–397.
- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 249–256).
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1), 723–773.
- Gulrajani, I., & Lopez-Paz, D. (2020). In search of lost domain generalization. In *International Conference on Learning Representations*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.
- Johansson, F. D., Sontag, D., & Ranganath, R. (2019). Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics* (pp. 527–536). PMLR.
- Kifer, D., Ben-David, S., & Gehrke, J. (2004). *Detecting change in data streams*. *VLDB*, 4, 180–191.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. [arXiv preprint arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).
- Kullback, S. (1997). *Information theory and statistics*. North Chelmsford: Courier Corporation.
- Kuroki, S., Charoenphakdee, N., Bao, H., Honda, J., Sato, I., & Sugiyama, M. (2019). Unsupervised domain adaptation based on source-guided discrepancy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 4122–4129.
- Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. M. (2017). Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*. (pp. 5542–5550).
- Li, D., Yang, Y., Song, Y.-Z., & Hospedales, T. M. (2018a). Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Li, H., Jialin Pan, S., Wang, S., & Kot, A. C. (2018b). Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5400–5409).
- Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., & Tao, D. (2018c). Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 624–639).
- Lipton, Z., Wang, Y.-X., & Smola, A. (2018). Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning* (pp. 3122–3130). PMLR.
- Liu, H., Long, M., Wang, J., & Jordan, M. (2019). Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning* (pp. 4013–4022).
- Li, H., Wang, Y., Wan, R., Wang, S., Li, T.-Q., & Kot, A. (2020). Domain generalization for medical imaging classification with linear-dependency regularization. *Advances in Neural Information Processing Systems*, 33, 3118–3129.
- Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation with multiple sources. In: *Advances in Neural Information Processing Systems* (pp. 1041–1048).
- Matsuura, T., & Harada, T. (2020). Domain generalization using a mixture of multiple latent domains. In *AAAI*
- Muandet, K., Balduzzi, D., & Schölkopf, B. (2013). Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, (pp. 10–18).
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* (Vol. 32, pp. 8024–8035). Curran Associates, Inc., New York.

- Redko, I., Habrard, A., & Sebban, M. (2017). Theoretical analysis of domain adaptation with optimal transport. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 737–753). New York: Springer.
- Schoenauer-Sebag, A., Heinrich, L., Schoenauer, M., Sebag, M., Wu, L. F. & Altschuler, S. J. (2019) Multi-domain adversarial learning. In *International Conference on Learning Representation*.
- Seo, S., Suh, Y., Kim, D., Han, J., & Han, B. (2019). Learning to optimize domain specific normalization with domain augmentation for domain generalization. arXiv preprint [arXiv:1907.04275](https://arxiv.org/abs/1907.04275)
- Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., & Sarawagi, S. (2018) Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*
- Shen, J., Qu, Y., Zhang, W., & Yu, Y. (2018). Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
- Shu, R., Bui, H. H., Narui, H., & Ermon, S. (2018). A dirt-t approach to unsupervised domain adaptation. arXiv preprint [arXiv:1802.08735](https://arxiv.org/abs/1802.08735)
- Sugiyama, M., Krauledat, M., & Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 985–1005.
- Tachet des Combes, R., Zhao, H., Wang, Y.-X., & Gordon, G. J. (2020). Domain adaptation with conditional distribution matching and generalized label shift. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., Lin, H. (eds.) *Advances in Neural Information Processing Systems*, (Vol. 33, pp. 19276–19289). Curran Associates, Inc., <https://proceedings.neurips.cc/paper/2020/file/dfbfa7ddcfff2eb581f50edcf9a0204bb-Paper.pdf>
- Vapnik, V. (1999). *The nature of statistical learning theory*. New York: Springer.
- Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5018–5027)
- Volpi, R., Namkoong, H., Sener, O., Duchi, J.C., Murino, V., & Savarese, S. (2018). Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems* (pp. 5334–5344).
- Wang, H., He, Z., Lipton, Z. L., & Xing, E. P. (2019). Learning robust representations by projecting superficial statistics out. In *International Conference on Learning Representations* <https://openreview.net/forum?id=rJEjjoR9K7>
- Wright, S. (2016). Chapter 2 (from an upcoming textbook). In *IMA New Directions Workshop on Mathematical Optimization*, p. 20 <http://www.pages.cs.wisc.edu/~swright/nd2016/chapter2.pdf>
- Ye, H., Xie, C., Cai, T., Li, R., Li, Z., & Wang, L. (2021). Towards a theoretical framework of out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 34, 23519–23531.
- Yosinski, J., Clune, J., Fuchs, T., & Lipson, H. (2015). Understanding neural networks through deep visualization. In *ICML Workshop on Deep Learning, Citeseer*.
- You, K., Wang, X., Long, M., & Jordan, M. (2019). Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning* (pp. 7124–7133). PMLR.
- Zhang, Y., Deng, B., Tang, H., Zhang, L., & Jia, K. (2020). Unsupervised multi-class domain adaptation: Theory, algorithms, and practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Zhang, Y., Li, M., Li, R., Jia, K., & Zhang, L. (2022). Exact feature distribution matching for arbitrary style transfer and domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8035–8045).
- Zhang, Y., Liu, T., Long, M., & Jordan, M. (2019). Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning* (pp. 7404–7413).
- Zhao, H., Des Combes, R. T., Zhang, K., & Gordon, G. (2019). On learning invariant representations for domain adaptation. In *International Conference on Machine Learning* (pp. 7523–7532). PMLR.
- Zhao, H., Zhang, S., Wu, G., Moura, J. M., Costeira, J. P., & Gordon, G. J. (2018). Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems* (pp. 8559–8570).
- Zhou, K., Yang, Y., Hospedales, T., & Xiang, T. (2020). Deep domain-adversarial image generation for domain generalisation. arXiv preprint [arXiv:2003.06054](https://arxiv.org/abs/2003.06054)
- Zhou, K., Yang, Y., Qiao, Y., & Xiang, T. (2021). Domain generalization with mixstyle. arXiv preprint [arXiv:2104.02008](https://arxiv.org/abs/2104.02008)