GENERALIZING TO UNSEEN DOMAINS VIA DISTRIBUTION MATCHING

Isabela Albuquerque[1],[*], João Monteiro[1], Mohammad Darvishi[2], Tiago H. Falk[1], Ioannis Mitliagkas[3]
[1]INRS-EMT, Université du Québec
[2] Faubert Lab, Université de Montréal
[3]Mila & DIRO, Université de Montréal

ABSTRACT

Supervised learning results typically rely on assumptions of i.i.d. data. Unfortunately, those assumptions are commonly violated in practice. In this work, we tackle such problem by focusing on domain generalization: a formalization where the data generating process at test time may yield samples from never-before-seen domains (distributions). Our work relies on the following lemma: by minimizing a notion of discrepancy between all pairs from a set of given domains, we also minimize the discrepancy between any pairs of mixtures of domains. Using this result, we derive a generalization bound for our setting. We then show that low risk over unseen domains can be achieved by representing the data in a space where (i) the training distributions are indistinguishable, and (ii) relevant information for the task at hand is preserved. Minimizing the terms in our bound yields an adversarial formulation which estimates and minimizes pairwise discrepancies. We validate our proposed strategy on standard domain generalization benchmarks, outperforming a number of recently introduced methods. Notably, we tackle a real-world application where the underlying data corresponds to multi-channel electroencephalography time series from different subjects, each considered as a distinct domain.

1 Introduction

The main assumption within the empirical risk minimization framework is that all examples used for training and testing predictors are independently drawn from a fixed distribution, i.e. the i.i.d. assumption. A number of generalization guarantees were derived upon that assumption and those results induced several algorithms for the solution of supervised learning problems. However, important limitations in this setting can be highlighted: i) the i.i.d. property is *unverifiable* [1] given that one doesn't have access to the data distribution, and ii) it doesn't account for distribution shifts which often occur in practice. Representative examples of these distribution shifts include changes in data acquisition conditions, such as illumination in images for object segmentation, or new data sources such as unseen speakers when performing speech recognition.

A number of alternative settings was then introduced in order to better cope with more realistic cases. Risk minimization under the *domain adaptation* setting, for instance, relaxes part of the i.i.d. assumption by allowing a source distribution (or domain)[2] as well as a different target distribution observed at test time. The domain adaptation results introduced in [2] showed that the generalization gap in terms of risk difference across the two considered distributions for a fixed predictor is upper bounded by a notion of distance measured between the training and testing domains. While less restrictive than the previous setting, the domain adaptation case is still limited in that only pairs of distributions seen during training are expected to yield low risk, and shifts beyond those domains will likely induce poor performance. Moreover, algorithms devised for this setting *rely on access at training time to an unlabeled sample from the target*

[*] Correspondence to `isabelamcalbuquerque@gmail.com`

[2] We use the terms *domain*, *data distribution*, and *data source* interchangeably throughout the text.

distribution so that representations can be learned inducing invariance across train and target domains [3]. This is a limiting factor for practical applications where target domain data may be inaccessible; for example, a speech recognition service cannot be (re)trained on data obtained from every new speaker it observes.

A more general setting is often referred to as *domain generalization* [4]. In this case, it is assumed that a set of distributions over the data is available at training time. At test time, however, both observed distributions as well as unseen novel domains might appear, and a low risk is expected regardless of the underlying domain. More importantly, unlike domain adaptation in which the goal is to find a representation that aligns training data distributions with a specific target domain, *domain generalization strategies aim at finding a representation space that yields good performance on novel distributions, unknown at training time.* Recent work on domain generalization has included the use of data augmentation [5, 6] at training time, meta-learning to simulate domain shift [7], adding a self-supervised task to encourage an encoder to learn robust representations [8, 9], and learning domain-invariant representations [10], among other approaches.

In this paper, we propose an innovation within the domain generalization setting. We first argue and prove that, given a set of distributions over data, if the distances measured between any pair of such distributions is small, so is the distance between mixtures obtained from the same set. This leads to the development of a bound on the risk measured against any distribution, and further shows that generalization can be expected if one considers distributions on the neighborhood of the "convex hull"[3] defined by the set of domains accessible during training. Inspired by these findings, we define an approach so that an encoder is enforced to map the data to a space where domain-dependent cues are filtered away while relevant information to the task of interest is conserved. While doing so, unlike standard domain adaptation approaches, no data from test distributions is observed.

We summarize our contributions in the following:

1. We introduce assumptions on the data generating process tailored to the domain generalization setting, which we argue are more general than standard i.i.d. requirements and more likely to hold in practice. In other words, given a data sample, it is more likely that our assumptions will hold compared to the more restrictive i.i.d. property;
2. We prove a generalization bound for the risk over unseen domains and show that generalization can be expected for domains on the neighborhood of a notion of convex hull of distributions observed at training time;
3. Aiming to minimize the terms of the introduced bound, we devise an adversarial approach so that pairwise domain divergences are estimated and minimized. In order to do so, several practical improvements are proposed on top of previous approaches for domain adaption including the use of random projection layers prior to domain discriminators.
4. We provide evidence through empirical evaluation showing that the proposed approach yields improvements relative to alternative methods across scenarios where different assumptions over the observed domains hold, including realistic cases where the labeling functions might shift.

The remainder of this paper is organized as follows: In Section 2 we introduce the notation adopted within this work, and review related work and generalization guarantees. In section 3, we define the domain generalization setting and present our main results, as well as the resulting algorithm. Section 4 provides the experiment descriptions and their respective results. Section 5 concludes the paper.

2 Background

2.1 Notation

Let the data be represented by $\mathcal{X} \subset \mathbb{R}^D$, while $\mathcal{Y}$ denotes the label space. Examples correspond to a pair $(x, y) : x \in \mathcal{X}, y \in \mathcal{Y}$, such that $y = f_\mathcal{D}(x)$, and $f_\mathcal{D} : \mathcal{X} \to \mathcal{Y}$ is a deterministic labeling function.

A domain is defined as a tuple $\langle \mathcal{D}, f_\mathcal{D} \rangle$ where $\mathcal{D}$ corresponds to a probability distribution over $\mathcal{X}$. Moreover, we define a mapping $h : \mathcal{X} \to \mathcal{Y}$, such that $h \in \mathcal{H}$, where $\mathcal{H}$ is a set of candidate hypothesis, and finally define the risk R associated with a given hypothesis h on domain $\langle \mathcal{D}, f_\mathcal{D} \rangle$ as:

$$R[h] = \mathbb{E}_{x \sim \mathcal{D}} \ell[h(x), f_\mathcal{D}(x)], \tag{1}$$

where the loss $\ell : \mathcal{Y} \times \mathcal{Y} \to R_+$ quantifies how different $h(x)$ is from the true labeling function $y = f_\mathcal{D}(x)$ for a given instance (x, y).

[3] i.e., the set of all mixtures obtained from given distributions.

2.2 Related work

A number of contributions under the domain generalization setting borrowed tools from causal inference to enforce the learned representations to be invariant across the different domains presented to the model at training time [11, 12, 13]. Other contributions such as [8] and [9] proposed different strategies to leverage self-supervised tasks to improve the out-of-distribution performance of a given model. Inspired by the domain adaptation literature [14, 2, 3], previous work on domain generalization also proposed to add a regularization term based on the minimization of a notion of divergence between the source domains to the empirical loss computed on the training data. This is the case of CIDDG [10], where class-specific domain classifiers are employed to induce the encoder to learn representations where the mismatch between the labels conditional distributions is minimized. Moreover, [15] proposed MMD-AAE, an approach that relies on an adversarial autoencoder used along with a maximum mean discrepancy penalty [16] to remove domain-specific information.

Recent work has proposed settings where domain-shifts are simulated at training time by splitting the source domains into meta-train and meta-test sets [7, 17, 18, 19]. Strategies based on learning domain-invariant representations [4], data augmentation [6, 5], and on decomposing the model's parameters into domain-agnostic and domain-specific components [20] have also been introduced. Work on other settings with more restrictive assumptions than domain generalization are also related to our contribution. For example, recent work on multi-domain learning [21], a setting where multiple domains are available at training time and test data is drawn from the same distributions seen during training [22], also leveraged an adversarial approach to perform $\mathcal{H}$-divergence minimization.

A straightforward approach to extend the empirical risk minimization (ERM) setting for domain generalization would be to learn h minimizing the empirical risk $\hat{R}[h]$ measured over all N_S source domains and *hope* generalization would be achieved to the target data, i.e.:

$$h = \arg\min \hat{R} = \frac{1}{N_S}\sum_{j=1}^{N_S}\frac{1}{M_j}\sum_{i=1}^{M_j}\ell[h(x_i), f(x_i)]. \tag{2}$$

In fact, as will be discussed in more detail in next sections, such a rather simplistic approach often yields strong baselines.

2.3 Generalization guarantees for domain adaptation

We now state results from the domain adaptation literature which are relevant in the context of this work. The discussion in [23] established the theoretical foundations for studying cross-domain generalization properties for domain adaptation problems. They showed that, given a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$, the risk of a given hypothesis $h \in \mathcal{H}$, $h : \mathcal{X} \to \{0, 1\}$, on the target is bounded by:

$$R_T[h] \leq R_S[h] + d_{\mathcal{H}\Delta\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T] + \lambda, \tag{3}$$

where λ corresponds to the minimal total risk over both domains which can be achieved within a given hypothesis class $\mathcal{H}$. The term $d_{\mathcal{H}\Delta\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T]$ corresponds to the $\mathcal{H}$-divergence introduced in [14] for a hypothesis class $\mathcal{H}\Delta\mathcal{H} = \{h(x) \oplus h'(x) | h, h' \in \mathcal{H}\}$, where $\oplus$ is the XOR function. The $\mathcal{H}$-divergence between two distributions $\mathcal{D}_S$ and $\mathcal{D}_T$ is defined as:

$$d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T] = 2 \sup_{\eta \in \mathcal{H}} |\text{Pr}_{x \sim \mathcal{D}_S}[\eta(x) = 1] - \text{Pr}_{x \sim \mathcal{D}_T}[\eta(x) = 1]|. \tag{4}$$

As discussed in [2], an estimate of $d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T]$ can be directly computed from the error of a binary classifier trained to distinguish domains.

3 Learning domain agnostic representations for domain generalization

3.1 Formalizing domain generalization

We start by defining a set of assumptions over the data generating process considering the domain generalization case as well as the notion of risk we are concerned with. We then define $\mathfrak{D}$, referred to as *meta-distribution*, corresponding to a probability distribution over a countable set of possible domains. Under this view, a query for a data example consists of: i) sampling a domain from the meta-distribution, and ii) sampling a data point according to that particular domain. Such process is repeated m times so as to yield a training sample $(x^m \sim \mathfrak{D}^m, y^m)$. We remark the described model of data generating processes is sufficiently general so as to include the i.i.d. case (the meta-distribution yields a single domain) as well as the domain adaptation setting (if two domains are allowed), but further supports several other cases where multiple domains exist.

Once a finite train sample is collected, a set of N_S domains is observed. Each distribution $\mathcal{D}_S^i$, $i \in [N_S]$, in such set will be referred to as source domain. At test time, however, drawing samples from $\mathfrak{D}$ might yield data distributed according to new unseen domains. We then introduce extra notation and represent the set of possible domains unobserved while train data is acquired by $\mathcal{D}_U^j$, $j \in [N_U]$. The labeling rules corresponding to each domain are denoted as f_{S_i} and f_{U_j}, for the source and unseen domains, respectively. For the sake of clarity, we hereinafter omit the index from the notation corresponding to unseen domains whenever it can be inferred from the context.

We proceed and define a risk minimization framework similar to that corresponding to the i.i.d. setting: find the predictor $h^* \in \mathcal{H}$ that minimizes the meta-risk $R_{\mathfrak{D}}[h]$ defined as follows:

$$\begin{aligned} &h^* \in \underset{h\in\mathcal{H}}{\operatorname{argmin}}\ R_{\mathfrak{D}}[h], \\ &R_{\mathfrak{D}}[h] = \mathbb{E}_{\mathcal{D}\sim\mathfrak{D}}[\mathbb{E}_{x\sim\mathcal{D}}[\ell(h(x), f_{\mathcal{D}}(x))]]. \end{aligned} \tag{5}$$

However, within the domain generalization setting, no information regarding possible test distributions is available at training time, which renders estimating $R_{\mathfrak{D}}[h]$ uninformative for a practical number of source domains. Moreover, we argue that no-free-lunch type of impossibility results may be used to conclude that it is impossible to generalize to any possible unknown distribution[4], so that one must assume something about the test domains in order to enable generalization. In the following results, we tackle this issue and introduce generalization guarantees for a particular set of domains lying close to the set of mixtures of *source distributions*, i.e., those observed once train data is collected.

3.2 Matching distributions in the convex hull

Let a set S of source domains such that $|S| = N_S$ be denoted by $\mathcal{D}_S^i$, $i \in [N_S]$. The convex hull Λ_S of S is defined as the set of mixture distributions given by: $\Lambda_S = \{\bar{\mathcal{D}} : \bar{\mathcal{D}}(\cdot) = \sum_{i=1}^{N_S} \pi_i \mathcal{D}_S^i(\cdot), \pi_i \in \Delta_{N_S-1}\}$, where Δ_{N_S-1} is the $N_S - 1$-th dimensional simplex. The following lemma shows that for any pair of domains such that $\mathcal{D}', \mathcal{D}'' \in \Lambda_S^2$, the $\mathcal{H}$-divergence between $\mathcal{D}'$ and $\mathcal{D}''$ is upper-bounded by the largest $\mathcal{H}$-divergence measured between elements of S.

Lemma 1 *(Bounding the $\mathcal{H}$-divergence between domains in the convex hull of the sources). Let $d_{\mathcal{H}}[\mathcal{D}_S^i, \mathcal{D}_S^k] \leq \epsilon$, $\forall\, i, k \in [N_S]$. The following inequality holds for the $\mathcal{H}$-divergence between any pair of domains $\mathcal{D}', \mathcal{D}'' \in \Lambda_S^2$:*

$$d_{\mathcal{H}}[\mathcal{D}', \mathcal{D}''] \leq \epsilon. \tag{6}$$

Proof. C.f. Supplementary material.

We thus argue that if one minimizes the maximum pairwise $\mathcal{H}$-divergence between source domains, which can be achieved by an encoding process that filters away domain discriminative cues, the $\mathcal{H}$-divergence between any two domains in Λ_S also decreases.

3.3 Generalizing to unseen domains

Now we turn our attention to the set of unseen distributions $\mathcal{D}_U^j$, $j \in [N_U]$, i.e., those in the support of the meta-distribution but not observed within the training sample. Given an unseen domain $\mathcal{D}_U$, we further introduce $\bar{\mathcal{D}}_U$, the element within Λ_S which is closest to $\mathcal{D}_U$, i.e., $\bar{\mathcal{D}}_U$ is given by $\operatorname{argmin}_{\pi_1,\ldots,\pi_{N_S}} d_{\mathcal{H}} \left[\mathcal{D}_U, \sum_{i=1}^{N_S} \pi_i \mathcal{D}_S^i\right]$. We now use Lemma 1 and previously proposed generalization bounds for the domain adaptation setting [24, 25] to derive a generalization bound for the risk $R_U[h]$.

Theorem 1 *(Upper-bounding the risk on unseen domains). Given the previous setup, let S be the set of source domains and $\mathcal{Y} = [0, 1]$. The risk $R_U[h]$, $\forall h \in \mathcal{H}$, for **any** unseen domain $\mathcal{D}_U$ such that $d_{\mathcal{H}}[\bar{\mathcal{D}}_U, \mathcal{D}_U] = \gamma$ is bounded as:*

$$R_U[h] \leq \sum_{i=1}^{N_S} \pi_i R_S^i[h] + \gamma + \epsilon + \min\{\mathbb{E}_{\bar{\mathcal{D}}_U}[|f_{S_\pi} - f_U|], \mathbb{E}_{\mathcal{D}_U}[|f_U - f_{S_\pi}|]\}, \tag{7}$$

where ϵ is the highest pairwise $\tilde{\mathcal{H}}$-divergence measured between pairs within S, $\tilde{\mathcal{H}} = \{sign(|h(x) - h'(x)| - t) |\, h, h' \in \mathcal{H}, 0 \leq t \leq 1\}$ and $f_{S_\pi}(x) = \sum_{i=i}^{N_S} \pi_i f_{S_i}(x)$ is the labeling function for any $x \in Supp(\bar{\mathcal{D}}_U)$ resulting from combining all f_{S_i} with weights π_i, $i \in [N_S]$, determined by $\bar{\mathcal{D}}_U$.

Proof. C.f. Supplementary material.

[4] For a fixed h, one can always define a distribution yielding high risk.

Remark 1: Notice that the right-most term in Theorem 1 accounts for the mismatch between the labeling functions f_{S_π} and f_U, which reduces to 0 in most adopted scenarios within domain adaption/generalization applications, since it is often considered that the *covariate shift assumption* holds [26]. Under such setting, the labeling functions are the same across all domains in the support of $\mathfrak{D}$, i.e. $f_{S_i} = f_{U_j} = f$ for all $i \in [N_S]$ and $j \in [N_U]$. Besides the covariate shift assumption, previous work on multi-source domain adaptation [27] considered the case where the unseen domain $\mathcal{D}_U$ can be represented as a mixture of the sources with weights π_i, $i \in [N_S]$, i.e. $\mathcal{D}_U = \bar{\mathcal{D}}_U$. When such assumption holds, the term indicated by γ in Theorem 1 will vanish. We thus re-state in Corollary 1 the previous result under such simplifying assumptions.

Corollary 1 *(Generalization to unseen domains within Λ_S under the covariate shift assumption). Let all domains within the the support of the meta-distribution $\mathfrak{D}$ have labeling function f. Let S be set of source domains and its convex-hull be denoted as Λ_S. The risk $R_U[h]$ of a hypothesis h on an unseen domain $\mathcal{D}_U \in \Lambda_S$, is upper-bounded by:*

$$R_U[h] \leq \sum_{i=1}^{N_S} \pi_i R_S^i[h] + \epsilon. \tag{8}$$

Proof. C.f. Supplementary material.

Remark 2: Based on the introduced results we define in the following an algorithm relying solely on source data, *unlike domain adaptation approaches*. While the total source risk can be minimized as usual, ϵ can be minimized by encoding source data to a space where source domains are hard to distinguish. We remark that we empirically found (c.f. Sections 4.1.1 and 4.2) the proposed algorithm was able to succeed even in scenarios where the considered assumptions are not likely to hold.

Remark 3: We further highlight that the introduced results also provide insights regarding *the importance of acquiring diverse datasets* in practice when targeting domain generalization (and hint as to why data augmentation is often helpful). The more diverse a dataset is regarding the number of domains present at training time, more likely it is that an unseen distribution lies within the convex hull of the source domains (i.e. $\gamma \to 0$). Therefore, not only the amount of data is important to achieve better generalization on unseen domains, but also the diversity of the training data is crucial.

Remark 4: Another practical aspect worth remarking is that, even though our domain generalization setting is more general than ERM, Theorem 1 suggests that source domain labels should also be available, since they are required to estimate ϵ, which is not the case for ERM. However, collecting domain labels is inherent to the data acquisition procedure for several tasks and commonly available as meta-data in cases such as, speech recognition, where different speakers or recording devices can be viewed as different domains.

3.4 Practical contributions

Motivated by the previous results, we propose to design algorithms that minimize the terms in the bound in (14) that can be estimated even if only source data is observed, i.e., ϵ as well as the risks over the train sample. We thus aim at learning an encoder $E : \mathcal{X} \to \mathcal{Z}$, where $\mathcal{Z} \subset \mathbb{R}^d$ preserves information relevant for separating classes, while removing domain-specific cues in such a way that it is harder to distinguish examples from different domains in comparison to the original space $\mathcal{X}$.

Efficiently estimating ϵ: Previous work on domain adaptation introduced strategies based on minimizing the empirical $\mathcal{H}$-divergence between sources and a given target domain [3, 24]. Instead, as per the discussion following Theorem 1, the domain generalization setting requires estimating pairwise $\mathcal{H}$-divergences across all available sources, not considering target data of any sort. Naively extending previous methods to our case would require $\mathcal{O}(N_S^2)$ estimators, which is unpractical given real-world cases where several source domains are available. We thus propose to use *one-vs-all* classifiers. In this case, there is one domain discriminator per source domain and the k-th discriminator estimates $\sum_{l \neq k} d_\mathcal{H}[\mathcal{D}_S^k, \mathcal{D}_S^l]$, and improves the method to a number of $\mathcal{H}$-divergence estimators linear on N_S.

Training: The proposed approach contains three main modules, all parameterized by neural networks: an encoder E with parameters ϕ, a task classifier C with parameters θ_C, and a set of $\mathcal{H}$-divergence estimators D_k with parameters θ_k, $k \in [N_S]$. Intuitively, E attempts to minimize a classification loss $\mathcal{L}_C(\cdot; \theta_C)$ (standard cross-entropy in our case) and empirical $\mathcal{H}$-divergences, which is achieved through the maximization of domain discrimination losses, denominated $\mathcal{L}_k$. Each domain discriminator, on the other hand, aims at minimizing $\mathcal{L}_k$. The procedure for estimating ϕ, θ_T, and all θ_k's can be thus formulated as the following multiplayer minimax game:

$$\min_{\phi, \theta_C} \max_{\theta_1, \ldots, \theta_{N_S}} \mathcal{L}_C(C(E(x; \phi); \theta_C), y_C) - \sum_{k=1}^{N_S} \mathcal{L}_k(D_k(E(x; \phi); \theta_k), y_k), \tag{9}$$

where y_C corresponds to the task label for the example x, and y_k is equal to 1 in case $x \sim \mathcal{D}_S^k$, or 0 otherwise. Training is carried out with alternate updates. A pseudocode describing the training procedure is presented in Algorithm 1. To further illustrate our proposed approach, we provide in Figure 2 (in the Supplementary material) a diagram showing the main components of the model in a case where three domains are available at training time.

Algorithm 1 Generalizing to unseen Domains via Distribution Matching

```
Requires: classifier and encoder learning rate ($\beta_C$), domain discriminators learning rate ($\beta_D$), scaling ($\alpha$), mini-batch size ($m$).
Initialize $\phi$, $\theta_C$, $\theta_1, \dots, \theta_{N_S}$ as $\phi^0$, $\theta_C^0$, $\theta_1^0, \dots, \theta_{N_S}^0$.
for $t = 1, \dots,$ number of iterations do
    Sample one mini-batch from each source domain $\{(x_1^i, y_C^i, y_1^i, \dots, y_{N_S}^i)\}_{i=1}^m$
    # Update domain discriminators
    for $k = 1, \dots, N_S$ do
        $\theta_k^t \leftarrow \theta_k^{t-1} + \frac{\beta_D}{N_S \cdot m} \sum_{i=1}^{N_S \cdot m} \nabla_{\theta_k} \mathcal{L}_k(D_k(E(x^i; \phi^{t-1}); \theta_k^{t-1}), y_k^i)$
    end for
    # Update task classifier
    $\theta_C^t \leftarrow \theta_C^{t-1} + \frac{\beta_C}{N_S \cdot m} \sum_{i=1}^{N_S \cdot m} \nabla_{\theta_C} \mathcal{L}_C(C(E(x^i; \phi^{t-1}); \theta_C^{t-1}), y_C^i)$
    # Update encoder
    $\phi^t \leftarrow \phi^{t-1} + \frac{\beta_C}{N_S \cdot m} (\sum_{i=1}^{N_S \cdot m} \alpha \nabla_{\phi} \mathcal{L}_C(C(E(x^i; \phi^{t-1}); \theta_C^{t-1}), y_C^i)$
        $-(1-\alpha) \nabla_{\theta_k} \mathcal{L}_k(D_k(E(x^i; \phi^{t-1}); \theta_k^t), y_k^i))$
end for
```

Improving training stability: Previous work on domain adaptation/generalization [3, 10] proposed solving the problem stated in (9) using a gradient reversal layer [28]. We empirically observed such approach to be heavily dependent on the choice of hyperparameters in order for training to converge. We propose to augment the described adversarial approach using strategies originally utilized for stabilizing the training of generative adversarial networks with multiple discriminators [29, 30]. Namely, we include a random projection layer in the input of each domain discriminator with the goal of making examples from different distributions harder to be distinguished. In addition, we use the negative log-hypervolume instead of the summation in the game represented in (9) in order to assign more preference to solutions which decrease all pairwise divergences uniformly even in cases where there is a trade-off in their minimization. We refer to the proposed approach as G2DM (Generalizing to unseen Domains via Distribution Matching).

Differences to multi-source domain adaptation: We further remark the differences between G2DM and previous adversarial approaches which are often employed in domain adaptation. Essentially, G2DM compares examples *only from source domains* to learn domain-agnostic representations, i.e., there is no notion of target distribution. Other settings such as [31, 32] are more restricted in that a particular distribution is targeted and data from that distribution is required, besides the source data we use in our case. Moreover, those approaches do not aim at matching source distributions and only consider $\mathcal{H}$-divergences computed between each source domain and the given target. In the case of G2DM, on the other hand, the goal is to match source domain distributions to decrease ϵ, and thus only pairwise discrepancies between training domains are considered.

4 Experimental Setup and Results

We design our empirical evaluation to validate G2DM in conditions where different assumptions are satisfied. In the first scenario, we chose experimental conditions such that the covariate shift assumption holds. For that, we employ G2DM on object recognition tasks. In this case, we aim to answer the following research questions: i) Can G2DM perform better than standard ERM under i.i.d. assumptions by using information of source domains only? ii) Where does G2DM performance stand in comparison to previously proposed domain generalization strategies? iii) Is G2DM indeed enforcing distribution matching across source and unseen domains? And iv) What is the effect on the resulting performance given by different access models to test distributions during training?

We then evaluate whether G2DM is able to attain good out-of-domain performance even in the challenging scenario where the covariate shift assumption is likely to be violated. For that, we consider a real-world task that involves classifying electroencephalography (EEG) time series for affective state prediction, a burgeoning area within the human-machine systems field. In applications involving EEG data, subjects are often considered as distinct domains with different labeling functions [33]. Such shift can be attributed to environmental and anatomic factors such as device placement and scalp characteristics [34, 35].

4.1 Evaluation under covariate shift

The VLCS benchmark [36] is composed by examples from five overlapping classes from the VOC2007 [37], LabelMe [38], Caltech-101 [39], and SUN [40] datasets. PACS [20], in turn, consists of images distributed into seven classes from four different datasets: photo (P), art painting (A), cartoon (C), and sketch (S). We compare the performance of our proposed approach with a model trained with no mechanism to enforce domain generalization (referred to as ERM throughout this Section). Moreover, we consider the recently introduced invariant risk minimization (IRM) strategy [11] and include results reported in the literature achieved by Epi-FCR [19], JiGen [8] along with the ERM results they provided (referred to as ERM-JiGen), and MMD-AAE [15]. Finally, the adaptation of DANN for domain generalization reported in [19] was also considered. All such methods have as encoder the convolutional stack of AlexNet [41]. Further implementation details can be found in the Supplementary material.

In Tables 1 and 2, we report the average best accuracy across three runs with different random seeds on the test partition of the unseen domain under a leave-one-domain-out validation scheme. Results show that G2DM outperforms ERM in terms of average performance across the unseen domains for both benchmarks, and supports the claim that leveraging source domain information as done by G2DM provides an improvement on generalization to unseen distributions in comparison to simply considering the i.i.d. requirement is satisfied. G2DM further presented better average performance when compared to our implementation of IRM, as well as results from other methods previously reported in the literature. We finally highlight that G2DM showed an improvement in performance in more challenging domains [20], such as LabelMe and Sketch.

Table 1: Classification accuracy (%) on VLCS datasets for models trained with leave-one-domain-out validation.

Unseen domain (→)	V	L	C	S	Average
DANN	66.40	64.00	92.60	63.60	71.70
MMD-AAE	67.70	62.60	94.40	64.40	72.28
Epi-FCR	67.10	64.30	94.10	65.90	72.90
JiGen	70.62	60.90	96.93	64.30	73.19
ERM - JiGen	71.96	59.18	96.93	62.57	72.66
IRM	72.16	62.36	98.35	67.82	75.17
ERM	73.44	60.44	97.88	67.92	74.92
G2DM	71.14	67.63	95.52	69.37	75.92

Table 2: Classification accuracy (%) on PACS datasets for models trained with leave-one-domain-out validation.

Unseen domain (→)	P	A	C	S	Average
DANN	88.10	63.20	67.50	57.00	69.00
Epi-FCR	86.10	64.70	72.30	65.00	72.00
JiGen	89.00	67.63	71.71	65.18	73.38
ERM - JiGen	89.98	66.68	69.41	60.02	71.52
IRM	89.97	64.84	71.16	63.63	72.39
ERM	90.02	64.86	70.18	61.40	71.61
G2DM	88.12	66.60	73.36	66.19	73.55

4.1.1 Checking $\mathcal{H}$-divergences across sources and unseen domains

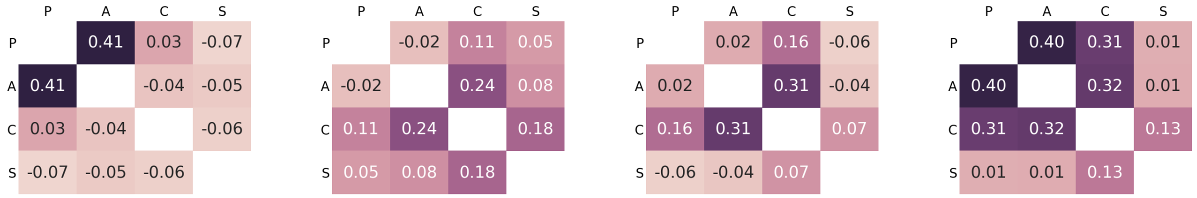
We now investigate whether cross-domain $\mathcal{H}$-divergences are being in fact reduced by G2DM. We use ERM as a baseline as it does not include any mechanism to enforce distribution matching. We estimate $\mathcal{H}$-divergences by computing the proxy pairwise $\mathcal{A}$-distance [2] for each pair of domains on the PACS benchmark. Classifiers are trained on top of the representations $\mathcal{Z}$ obtained with ERM and G2DM. We show in Figures 1 the differences in estimated discrepancies between ERM and G2DM for each unseen domain. Each entry corresponds to a pair of domains indicated in the row and the column and positive values indicate that G2DM *decreased* the corresponding pairwise $\mathcal{A}$-distance in comparison to ERM. Notice that the diagonals are left blank as we do not compute the classification accuracy between the same domains.

We observe that, apart from the case where 'photo' is the test domain, G2DM was in fact able to better match most of the source distributions, thus yielding smaller ϵ which favours generalization as predicted by Theorem 1. Notably, we highlight that although our proposed approach has no access to data from the unseen domain at training time and, therefore, does not directly implement a strategy to decrease the divergence between the unseen domain and the convex hull of the sources (i.e. γ), the results presented in Figure 1 show that the estimated pairwise $\mathcal{H}$-divergence between the unseen domain and sources also decreased in most of the considered cases.

In fact, the only mechanism the encoder has in order to reduce ϵ corresponds to learning how to filter domain information from the data, in the sense that once samples from two distinct distributions are encoded, one cannot distinguish from which distribution each sample came from. Observed results thus suggest that such encoder also removes domain information from the unseen distributions observed at test time, preventing the learning algorithm to yield a high γ.

4.1.2 The effect of different access methods to test data during training

Results of previous experiments correspond to an optimistic scenario where data from the unseen domain is made available for selecting the best performing model. This is not the case in practice since varying unseen distributions might appear. In Table 3, we compare results obtained further considering different access methods to the test data. Namely, we consider the case where no access to the unseen distribution is allowed and only source data can be used



(a) Photo. (b) Art. (c) Cartoon. (d) Sketch.

Figure 1: Differences between estimated pairwise $\mathcal{H}$-divergences under ERM and G2DM on PACS (captions denote unseen domains). Higher values indicate that G2DM better matched domains. Overall, G2DM is able to decrease pairwise discrepancies.

in order to define stopping criteria. In such cases, both validation accuracy and training loss computed on left-out in-domain data are employed (referred in Table 3 as source accuracy and source loss, respectively). Moreover, as a reference of performance, we further report the accuracy achieved assuming access to unseen domain data during training in order to select the best model (referred in Table 3 as unseen accuracy). For comparison, we further present the performance reported by [10] for CIDDG, since a stopping criterion using solely data from source domains was employed in that case. We observe that, when using the training loss as stopping criterion, our strategy outperforms CIDDG for almost all domains, while the baseline performance severely degrades when ‘sketch’ is the unseen domain.

As an alternative to AlexNet, we further evaluate the performance of the proposed approach using the convolutional stack of a ResNet-18 [42], since it has shown promising results in recent work [8]. We compare our approach with JiGen[5] adopting the same previously-discussed test data access methods for both approaches. We further report in Table 3 the performance obtained by JiGen as reported in [8] although it is unclear which stopping criteria were adopted for that case. We observe that replacing AlexNet by ResNet-18 yields a more stable average performance across stopping criteria. Based on the results obtained with AlexNet, we remark that results might be too optimistic/pessimistic depending on the assumed access method to unseen distributions, and as such, in order to allow fair comparison between different approaches, *the performance across different access methods should be reported.*

Table 3: Accuracy (%) on PACS with different stopping criteria.

Method	Criterion	P	A	C	S	Average
		AlexNet				
CIDDG [10]	From [10]	78.65	62.70	69.73	64.45	68.88
G2DM	Source acc.	85.33	57.76	69.71	49.45	65.56
	Source loss	87.37	66.70	70.26	50.98	68.82
	Unseen acc.	88.80	66.70	73.29	65.03	73.45
		ResNet-18				
JiGen [8]	Source acc.	95.83	78.52	73.31	69.14	79.20
	Source loss	95.83	78.89	73.32	70.73	79.69
	Unseen acc.	96.11	79.56	74.25	71.00	80.23
	From [8]	96.03	79.42	75.25	71.35	80.51
G2DM	Source acc.	93.70	79.22	76.34	75.14	81.10
	Source loss	93.75	77.78	75.54	77.58	81.16
	Unseen acc.	94.63	81.44	79.35	79.52	83.34

4.2 Evaluation beyond the covariate shift assumption

We proceed to evaluate G2DM on a unfavorable scenario where the covariate shift is unlikely to hold. The goal of the selected task is to perform affective state estimation with three classes (positive, neutral, or negative) based on EEG signals from the SEED dataset [43] collected from 15 subjects. We use the architecture described in [44] for both G2DM and ERM. For each subject left out for testing, we use 10 out of the remaining 14 domains for training and use the other 4 as validation data.

We report in Table 4 the classification accuracy (%) averaged across all unseen subjects and three independent training runs. Under **source data validation**, the reported performance was computed on the epoch of highest accuracy on the

[5] Results are generated using JiGen authors’ source code (`https://github.com/fmcarlucci/JigenDG`).

Table 4: Average accuracy (%) on the SEED dataset across 15 subjects. Semi-privileged approaches correspond to the best performing model under the domain generalization setting. Privileged baselines (domain adaptation setting) have access to unseen domain data at training time.

Setting	Method	Average accuracy (%)
	Source data validation	
Domain generalization	ERM	51.98
	G2DM	55.77
	Semi-privileged	
	ERM	56.82
	G2DM	60.26
Privileged baselines		
Domain adaptation	DAN [45, 46]	50.28
	DANN [3, 46]	55.87
	MDAN [24, 46]	56.65
	MDMN [46]	60.59

validation partition. The results under **semi-privileged** were obtained on the epoch of highest accuracy on the unseen subject data. The comparison between G2DM and ERM shows that even in this challenging case where the mismatch between labeling functions is not negligible, G2DM is able to successfully leverage the available domain information (which in this case comes with no additional effort at the data collection) and presents an improvement of more than 3.4% in accuracy in comparison to ERM in both considered scenarios for the DG setting.

Comparison with domain adaptation strategies: We further report in Table 4 results obtained by domain adaptation strategies (DA). Such methods, reported in Table 4 under **privileged baselines**, are privileged in the sense that unlabeled data belonging to the unseen domain (unknown in our case) is used to adapt representations at training time in order to yield **subject-specific** models. When comparing the DA strategies with our proposed domain generalization (DG) approach, we remark that DG strategies aim to obtain domain-agnostic models, as opposed to DA methods which target a specific distribution. As such, one would expect DA approaches to achieve better performance than DG. However, we observe G2DM's performance to be on par with, or even better than, some of the considered DA strategies. We conjecture a larger number of source domains available at training time would decrease the gap between DG and DA even further; i.e., it would be more likely that unseen domains are exactly represented in the convex hull of the sources yielding low γ (c.f. Theorem 1).

5 Conclusion

In this paper, we tackled the domain generalization problem and showed that generalization can be achieved in the neighborhood of the set of mixtures of distributions observed during training. Based on this result, we introduced G2DM, an efficient approach in yielding invariant representations across unseen distributions. Our method employs multiple one-vs-all domain discriminators, such that pairwise divergences between source distributions are estimated and minimized at training time. We provide empirical evidence supporting the claim that making use of domain information improves performance relative to standard settings relying on i.i.d. requirements. Moreover, the introduced approach outperformed recent methods which also leverage domain labels. We further showed that our proposed method resulted in strong results on a realistic setting, with performance comparable to privileged systems tailored to test distributions.

References

[1] J. Langford, “Tutorial on practical prediction theory for classification,” *Journal of machine learning research*, vol. 6, no. Mar, pp. 273–306, 2005.

[2] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in *Advances in neural information processing systems*, 2007, pp. 137–144.

[3] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[4] K. Muandet, D. Balduzzi, and B. Schölkopf, “Domain generalization via invariant feature representation,” in *International Conference on Machine Learning*, 2013, pp. 10–18.

[5] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, "Generalizing across domains via cross-gradient training," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Dx7fbCW

[6] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," in *Advances in Neural Information Processing Systems*, 2018, pp. 5334–5344.

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

[8] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi, "Domain generalization by solving jigsaw puzzles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2229–2238.

[9] I. Albuquerque, N. Naik, J. Li, N. Keskar, and R. Socher, "Improving out-of-distribution generalization via multi-task self-supervised pretraining," *arXiv preprint arXiv:2003.13525*, 2020.

[10] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, "Deep domain generalization via conditional invariant adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 624–639.

[11] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," *arXiv preprint arXiv:1907.02893*, 2019.

[12] D. Mahajan, S. Tople, and A. Sharma, "Domain generalization using causal matching," *arXiv preprint arXiv:2006.07500*, 2020.

[13] K. Ahuja, K. Shanmugam, K. Varshney, and A. Dhurandhar, "Invariant risk minimization games," *arXiv preprint arXiv:2002.04692*, 2020.

[14] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 180–191.

[15] H. Li, S. Jialin Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5400–5409.

[16] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[17] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," in *Advances in Neural Information Processing Systems*, 2018, pp. 998–1008.

[18] Q. Dou, D. C. de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," in *Advances in Neural Information Processing Systems*, 2019, pp. 6447–6458.

[19] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," *arXiv preprint arXiv:1902.00113*, 2019.

[20] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5542–5550.

[21] A. Schoenauer-Sebag, L. Heinrich, M. Schoenauer, M. Sebag, L. F. Wu, and S. J. Altschuler, "Multi-domain adversarial learning," *arXiv preprint arXiv:1903.09239*, 2019.

[22] M. Dredze, A. Kulesza, and K. Crammer, "Multi-domain learning by confidence-weighted parameter combination," *Machine Learning*, vol. 79, no. 1-2, pp. 123–149, 2010.

[23] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[24] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 8559–8570.

[25] H. Zhao, R. T. Des Combes, K. Zhang, and G. Gordon, "On learning invariant representations for domain adaptation," in *International Conference on Machine Learning*, 2019, pp. 7523–7532.

[26] S. Ben-David, T. Lu, T. Luu, and D. Pál, "Impossibility theorems for domain adaptation," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 129–136.

[27] J. Hoffman, M. Mohri, and N. Zhang, "Algorithms and theory for multiple-source adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 8246–8256.

[28] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[29] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti, "Stabilizing gan training with multiple random projections," *arXiv preprint arXiv:1705.07831*, 2017.

[30] I. Albuquerque, J. Monteiro, T. Doan, B. Considine, T. Falk, and I. Mitliagkas, "Multi-objective training of generative adversarial networks with multiple discriminators," in *International Conference on Machine Learning*, 2019, pp. 202–211.

[31] S. Sun, H. Shi, and Y. Wu, "A survey of multi-source domain adaptation," *Information Fusion*, vol. 24, pp. 84–92, 2015.

[32] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1406–1415.

[33] I. Albuquerque, J. Monteiro, O. Rosanne, A. Tiwari, J.-F. Gagnon, and T. H. Falk, "Cross-subject statistical shift estimation for generalized electroencephalography-based mental workload assessment," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 3647–3653.

[34] D. Wu, V. J. Lawhern, and B. J. Lance, "Reducing bci calibration effort in rsvp tasks using online weighted adaptation regularization with source domain selection," in *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2015, pp. 567–573.

[35] C.-S. Wei, Y.-P. Lin, Y.-T. Wang, C.-T. Lin, and T.-P. Jung, "A subject-transfer framework for obviating inter-and intra-subject variability in eeg-based drowsiness detection," *NeuroImage*, vol. 174, pp. 407–419, 2018.

[36] C. Fang, Y. Xu, and D. N. Rockmore, "Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1657–1664.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[38] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[39] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[40] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, "Exploiting hierarchical context on a large database of object categories," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 129–136.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, 2015.

[44] Y. Li, K. Dzirasa, L. Carin, D. E. Carlson *et al.*, "Targeting eeg/lfp synchrony with neural nets," in *Advances in Neural Information Processing Systems*, 2017, pp. 4620–4630.

[45] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*, 2015, pp. 97–105.

[46] Y. Li, D. E. Carlson *et al.*, "Extracting relationships by multi-domain matching," in *Advances in Neural Information Processing Systems*, 2018, pp. 6798–6809.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

Supplementary Material

A Proof of Lemma 1

Lemma 1. *Let $d_{\mathcal{H}}[\mathcal{D}_S^i, \mathcal{D}_S^k] \leq \epsilon,\ \forall\, i, k \in [N_S]$. The following inequality holds for the $\mathcal{H}$-divergence between any pair of domains $\mathcal{D}', \mathcal{D}'' \in \Lambda_S^2$:*

$$d_{\mathcal{H}}[\mathcal{D}', \mathcal{D}''] \leq \epsilon. \tag{10}$$

Proof. Consider two unseen domains, $\mathcal{D}'_U$ and $\mathcal{D}''_U$ on the convex-hull Λ_S of N_S source domains with support Ω. Consider also $\mathcal{D}'_U(\cdot) = \sum_{k=1}^{N_S} \pi_k \mathcal{D}_S^k(\cdot)$ and $\mathcal{D}''_U(\cdot) = \sum_{l=1}^{N_S} \pi_l \mathcal{D}_S^l(\cdot)$ The $\mathcal{H}$-divergence between $\mathcal{D}'_U$ and $\mathcal{D}''_U$ can be written as:

$$\begin{aligned}
d_{\mathcal{H}}[\mathcal{D}'_U, \mathcal{D}''_U] =& 2 \sup_{h\in\mathcal{H}} |\mathrm{Pr}_{x\sim\mathcal{D}'_U}[h(x)=1] - \mathrm{Pr}_{x\sim\mathcal{D}''_U}[h(x)=1]|, \\
=& 2 \sup_{h\in\mathcal{H}} |\mathbb{E}_{x\sim\mathcal{D}'_U}[\mathbf{I}(h(x))] - \mathbb{E}_{x\sim\mathcal{D}''_U}[\mathbf{I}(h(x))]|, \\
=& 2 \sup_{h\in\mathcal{H}} \left| \int_\Omega \mathcal{D}'_U(x)\mathbf{I}(h(x))dx - \int_\Omega \mathcal{D}''_U(x)\mathbf{I}(h(x))dx \right|, \\
=& 2 \sup_{h\in\mathcal{H}} \left| \int_\Omega \sum_{k=1}^{N_S} \pi_k \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_\Omega \sum_{l=1}^{N_S} \pi_l \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right|, \\
=& 2 \sup_{h\in\mathcal{H}} \left| \int_\Omega \sum_{l=1}^{N_S}\sum_{k=1}^{N_S} \pi_l\pi_k \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_\Omega \sum_{l=1}^{N_S}\sum_{k=1}^{N_S} \pi_l\pi_k \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right|, \\
=& 2 \sup_{h\in\mathcal{H}} \left| \sum_{l=1}^{N_S}\sum_{k=1}^{N_S} \pi_l\pi_k \left(\int_\Omega \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_\Omega \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right) \right|.
\end{aligned} \tag{11}$$

Using the triangle inequality, we can write:

$$d_{\mathcal{H}}[\mathcal{D}'_U, \mathcal{D}''_U] \leq 2 \sup_{h\in\mathcal{H}} \sum_{l=1}^{N_S}\sum_{k=1}^{N_S} \pi_l\pi_k \left| \int_\Omega \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_\Omega \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right|. \tag{12}$$

Finally, using the sub-additivity of the sup:

$$\begin{aligned}
d_{\mathcal{H}}[\mathcal{D}'_U, \mathcal{D}''_U] \leq& \sum_{l=1}^{N_S}\sum_{k=1}^{N_S} \pi_l\pi_k 2 \sup_{h\in\mathcal{H}} \left| \int_\Omega \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_\Omega \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right|, \\
=& \sum_{l=1}^{N_S}\sum_{k=1}^{N_S} \pi_l\pi_k d_{\mathcal{H}}[\mathcal{D}_S^k, \mathcal{D}_S^l].
\end{aligned} \tag{13}$$

Given $d_{\mathcal{H}}[\mathcal{D}_S^k, \mathcal{D}_S^l] \leq \epsilon\ \forall\ k, l \in [N_S]$:

$$d_{\mathcal{H}}[\mathcal{D}'_U, \mathcal{D}''_U] \leq \epsilon. \qquad \square$$

B Proof of Theorem 1

Theorem 1. *Let S be the set of source domains and $\mathcal{Y} = [0, 1]$. The risk $R_U[h]$, $\forall h \in \mathcal{H}$, for **any** unseen domain $\mathcal{D}_U$ such that $d_{\mathcal{H}}[\bar{\mathcal{D}}_U, \mathcal{D}_U] = \gamma$, is bounded as:*

$$R_U[h] \leq \sum_{i=1}^{N_S} \pi_i R_S^i[h] + \gamma + \epsilon + \min\{\mathbb{E}_{\bar{\mathcal{D}}_U}[|f_{S_\pi} - f_U|], \mathbb{E}_{\mathcal{D}_U}[|f_U - f_{S_\pi}|]\}, \tag{14}$$

where ϵ is the highest pairwise $\tilde{\mathcal{H}}$-divergence measured between pairs of domains within S, $\tilde{\mathcal{H}} = \{sign(|h(x) - h'(x)| - t)|h, h' \in \mathcal{H}, 0 \leq t \leq 1\}$ and $f_{S_\pi}(x) = \sum_{i=i}^{N_S} \pi_i f_{S_i}(x)$ is the labeling function for any $x \in$ Supp$(\bar{\mathcal{D}}_U)$ resulting from combining all f_{S_i} with weights π_i, $i \in [N_S]$, determined by $\bar{\mathcal{D}}_U$.

Proof of Theorem 1. Let the source and target domains be $\langle \mathcal{D}_S, f_S \rangle$ and $\langle \mathcal{D}_T, f_T \rangle$, respectively. For the single-source, single-target domain adaptation case, it was previously shown that the risk of any $h \in \mathcal{H}$, $h : \mathcal{X} \to [0, 1]$ is bounded by [25]:

$$R_T[h] \leq R_S[h] + d_{\tilde{\mathcal{H}}}[\mathcal{D}_S, \mathcal{D}_T] + \min\{\mathbb{E}_{\mathcal{D}_S}[|f_S - f_T|], \mathbb{E}_{\mathcal{D}_T}[|f_T - f_S|]\}, \tag{15}$$

where $\tilde{\mathcal{H}} = \{sign(|h(x) - h'(x)| - t)|h, h' \in \mathcal{H}, 0 \leq t \leq 1\}$.

In order to devise a generalization bound for the risk on any unseen domain in terms of quantities related to the distributions seen at training time, we start by writing (15) considering $\mathcal{D}_U$ and its "projection" onto the convex-hull of the sources $\bar{\mathcal{D}}_U = \text{argmin}_{\pi_1,\ldots,\pi_{N_S}} d_{\mathcal{H}} \left[\mathcal{D}_U, \sum_{i=1}^{N_S} \pi_i \mathcal{D}_S^i\right]$. For that, we introduce the labeling function $f_{S_\pi}(x) = \sum_{i=i}^{N_S} \pi_i f_{S_i}(x)$, which is an ensemble of the respective labeling functions from each source domain weighted by the mixture coefficients that determine $\bar{\mathcal{D}_U}$. $R_U[h]$ can thus be bounded as:

$$R_U[h] \leq R_{\bar{U}}[h] + d_{\tilde{\mathcal{H}}}[\bar{\mathcal{D}}_U, \mathcal{D}_U] + \min\{\mathbb{E}_{\bar{\mathcal{D}}_U}[|f_{S_\pi} - f_U|], \mathbb{E}_{\mathcal{D}_U}[|f_U - f_{S_\pi}|]\}.$$

Similarly to the proof of our Lemma 1 for the case where $\mathcal{D}' = \mathcal{D}_U$ and $\mathcal{D}'' = \bar{\mathcal{D}}_U$ (and to [24]), it follows that:

$$R_U[h] \leq \sum_{i=1}^{N_S} \pi_i R_S^i[h] + \sum_{i=1}^{N_S} \pi_i d_{\tilde{\mathcal{H}}}[\mathcal{D}_S^i, \mathcal{D}_U] + \min\{\mathbb{E}_{\bar{\mathcal{D}}_U}[|f_{S_\pi} - f_U|], \mathbb{E}_{\mathcal{D}_U}[|f_U - f_{S_\pi}|]\}. \tag{16}$$

Using the triangle inequality for the $\mathcal{H}$-divergence along with Lemma 1, we can bound the $\tilde{\mathcal{H}}$-divergence between $\mathcal{D}_U$ and any source domain $\mathcal{D}_S^i$, $d_{\tilde{\mathcal{H}}}[\mathcal{D}_U, \mathcal{D}_S^i]$, according to:

$$\begin{aligned} d_{\tilde{\mathcal{H}}}[\mathcal{D}_U, \mathcal{D}_S^i] &\leq d_{\tilde{\mathcal{H}}}[\mathcal{D}_U, \bar{\mathcal{D}}_U] + d_{\tilde{\mathcal{H}}}[\bar{\mathcal{D}}_U, \mathcal{D}_S^i], \\ &\leq \gamma + \epsilon, \end{aligned}$$

where $\gamma = d_{\tilde{\mathcal{H}}}[\mathcal{D}_U, \bar{\mathcal{D}}_U]$. Using this result, we can now upper-bound $\sum_{i=1}^{N_S} \pi_i d_{\tilde{\mathcal{H}}}[\mathcal{D}_U, \mathcal{D}_S^i]$ by $\gamma + \epsilon$ and finally re-write (16) as:

$$R_U[h] \leq \sum_{i=1}^{N_S} \pi_i R_S^i[h] + \gamma + \epsilon + \min\{\mathbb{E}_{\bar{\mathcal{D}}_U}[|f_{S_\pi} - f_U|], \mathbb{E}_{\mathcal{D}_U}[|f_U - f_{S_\pi}|]\}. \qquad \square$$

C Proof of Corollary 1

Corollary 1. *Let all domains within the the support of the meta-distribution $\mathfrak{D}$ have labeling function f. Let S be set of source domains and its convex-hull be denoted as Λ_S. The risk $R_U[h]$ of a hypothesis h on an unseen domain $\mathcal{D}_U \in \Lambda_S$, is upper-bounded by:*

$$R_U[h] \leq \sum_{i=1}^{N_S} \pi_i R_S^i[h] + \epsilon. \tag{17}$$

Proof of Corollary 1. The right-most term of (14) accounts for the mismatch between the labeling functions of $\mathcal{D}_U$ and $\bar{\mathcal{D}}_U$. Since all domains within $\mathfrak{D}$ have the same labeling function, this term is equal to 0. As $\mathcal{D}_U \in \Lambda_S$, $\mathcal{D}_U = \bar{\mathcal{D}}_U$, which results in $d_{\tilde{\mathcal{H}}}[\mathcal{D}_U, \bar{\mathcal{D}}_U] = \gamma = 0$. □

D One-vs-all $\mathcal{H}$-divergence estimation

We illustrate the estimation of $\mathcal{H}$-divergences using one-vs-all discriminators by considering an example in which 3 source domains are available. Consider samples of size M from $N_S = 3$ source domains which are available at training time. The loss $\mathcal{L}_1$ for the domain discriminator D_1 accounting for estimating $d_{\mathcal{H}}[\mathcal{D}_1, \mathcal{D}_2]$ and $d_{\mathcal{H}}[\mathcal{D}_1, \mathcal{D}_3]$ can be written as:

$$\begin{aligned}\mathcal{L}_1 &= \frac{1}{3M}\sum_{i=1}^{3M} \ell(D_1(x_i), y_1), \\ &= \frac{1}{M}\sum_{i=1}^{M} \ell(D_1(x_i), y_1) + \frac{1}{M}\sum_{i=M+1}^{2M} \ell(D_1(x_i), y_1) + \frac{1}{M}\sum_{i=2M+1}^{3M} \ell(D_1(x_i), y_1),\end{aligned} \tag{18}$$

where ℓ represents a loss function (e.g. 0-1 loss) and each term accounts for the loss provided by examples from one domain. Splitting the first term in two parts and replacing the domain labels y_1 by their corresponding values, we obtain:

$$\begin{aligned}\mathcal{L}_1 = \frac{1}{M}\sum_{i=1}^{M/2} \ell(D_1(x_i), 1) + \frac{1}{M}\sum_{i=M+1}^{2M} \ell(D_1(x_i), 0) \\ + \frac{1}{M}\sum_{i=\frac{M}{2}+1}^{M} \ell(D_1(x_i), 1) + \frac{1}{M}\sum_{i=2M+1}^{3M} \ell(D_1(x_i), 0).\end{aligned} \tag{19}$$

The first two terms from Eq.19 account for $d_{\mathcal{H}}[\mathcal{D}_1, \mathcal{D}_2]$ and the last two terms account for $d_{\mathcal{H}}[\mathcal{D}_1, \mathcal{D}_3]$.

E Illustration

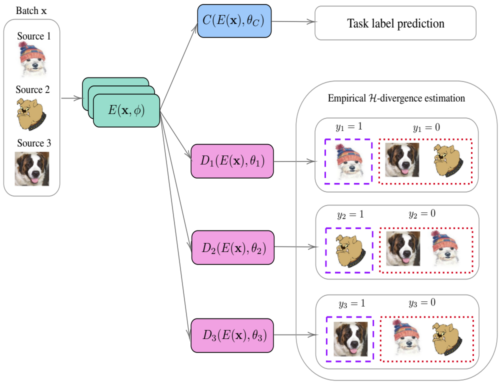

Batch $\mathbf{x}$
Source 1
Source 2
Source 3
$E(\mathbf{x}, \phi)$
$C(E(\mathbf{x}), \theta_C)$
Task label prediction
Empirical $\mathcal{H}$-divergence estimation
$D_1(E(\mathbf{x}), \theta_1)$
$y_1 = 1$
$y_1 = 0$
$D_2(E(\mathbf{x}), \theta_2)$
$y_2 = 1$
$y_2 = 0$
$D_3(E(\mathbf{x}), \theta_3)$
$y_3 = 1$
$y_3 = 0$

Figure 2: Proposed approach illustration.

F Extra experiments

F.1 Impact of source domains diversity on unseen domain accuracy

In this experiment, we verify whether removing examples from one source domain impacts the performance on the target domain. We evaluate each target domain on models trained using all possible combinations of the remaining domains as sources. The ERM baseline is also included for reference. Results presented in Table 5 show that for all unseen domains, decreasing the number of source domains from 3 (see Table 1) to 2 hurt the classification performance for almost all combinations of source domains. We notice that in some cases, excluding a particular source from the training severely decreases the target loss. As an example, for the Caltech-101, excluding from training examples from the VOC dataset decreased the accuracy in more than 10% for the proposed approach, as well as for ERM.

F.2 Effect of random projection size

We further investigate the effectiveness on providing a more stable training of the random projection layer in the input of each discriminator. For that, we run experiments with 7 different projection sizes, as well as directly using the output of

Table 5: Impact of decreasing the number of source domains on VLCS. Rows represent the two source domains used.

		Source					
Target	Method	VC	VL	VS	LC	LS	CS
V	ERM	-	-	-	66.14	72.16	69.89
	Ours	-	-	-	62.39	69.89	67.23
L	ERM	58.32	-	62.11	-	-	59.85
	Ours	65.37	-	65.87	-	-	64.37
C	ERM	-	98.82	98.58	-	84.67	-
	Ours	-	95.75	96.70	-	81.84	-
S	ERM	69.04	66.29	-	59.80	-	-
	Ours	69.54	68.43	-	57.06	-	-

the feature extractor model. Besides the random projection size, we use the same hyperparameters values (the same used in the previous experiment) and initialization for all models. We report in Figure 3 the best target accuracy achieved with all random projection sizes on the PACS benchmark considering the Sketch dataset as unseen domain. Overall, we observed that the random projection layer has indeed an impact on the generalization of the learned representation and that the best result was achieved with a size equal to 1000. Moreover, we notice that, in this case, having a smaller (500) random projection layer is less hurtful for the performance than using a larger one. We also found that removing the random projection layer did not allow the training to converge with this experimental setting.

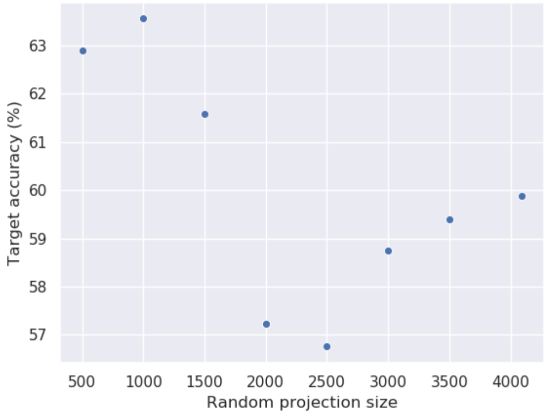


Figure 3: Accuracy obtained on the PACS benchmark using Sketch as target domain.

G Domain generalization benchmarks

The VLCS benchmark is composed by 4 datasets with 5 common classes, namely, bird, car, chair, dog, and person. The number of data points per dataset is detailed as follows. We split each dataset in 80%/20% train/test partitions.

- Pascal VOC2007: 3376;
- LabelMe: 2656;
- Caltech-101: 1415;
- SUN09: 3282.

The PACS benchmark is composed by 4 datasets with 7 common classes, namely, dog, elephant, giraffe, guitar, horse, house, and person. The number of data points per dataset is detailed as follows. We use the original train/validation partitions provided by the benchmark authors.

- Photos: 1670;
- Art painting: 2048;

- Cartoon: 2344;
- Sketch: 3929.

H Implementation details

H.1 VLCS and PACS benchmarks

In order to obtain a consistent comparison with the aforementioned baseline models, we follow previous work and employ the weights of a pre-trained AlexNet [41] and ResNet-18 [42] as the initialization for the feature extractor model on the experiments. The last layer is discarded and the representation of size 4096 for AlexNet and 512 for ResNet-18 is used as input for the task classifier and the domain discriminators. The domain discriminator architecture with AlexNet, consists of a four-layer fully-connected neural network of size $4096 \rightarrow$ random projection size $\rightarrow 1024 \rightarrow 1$ and five-layer fully connected network of size $512 \rightarrow$ random projection size $\rightarrow 512 \rightarrow 256 \rightarrow 1$ for ResNet-18. The random projection layer is implemented as a linear layer with weights normalized to have unitary L2-norm. The task classifier is a one-layer fully-connected network of size $4096 \rightarrow$ number of classes in the case of AlexNet and $512 \rightarrow$ number of classes in the case of ResNet. Following previous work on domain generalization [20, 19], we use models pre-trained on the ILSVRC dataset [47] as initialization. For fair comparison, all models we implemented were given a budget of 200 epochs. We use label smoothing [48] on the task classifier in order to prevent overfitting. Models were trained using SGD with Polyak's acceleration. One epoch corresponds to the length of the largest source domain training sample. The learning rate was "warmed-up" for a number of training iterations equal to nw. Hyperparameter tuning was performed through random search over a pre-defined grid so as to find the best values for the learning rate (lr), momentum, weight decay, label smoothing parameter ls, nw, random projection size[6], learning rate reduction factor, and weighting (α). Each model was run with three different initializations (random seeds 1, 10, and 100 selected *a priori*) and the average best accuracy on the test partition of the target domain is reported. Details of the hyperparameters grid used in the search are provided in the Supplementary material. For our ERM we used the same hyperparameters as in [8], while for IRM we employed the same hyperparameter values reported in the authors implementation of the colored MNIST experiments.

The grids used on the hyperparameter search for each hyperparameter are presented in the following. A budget of 200 runs was considered and for each combination of hyperparameters each model was trained for 200 and 30 epochs in the case of AlexNet and ResNet-18, respectively. The best hyperparamters values for AlexNet on PACS and VLCS benchmarks are respectively denoted by *, †. For the ResNet-18 experiments on PACS we indicate the hyperparameters by $^+$. Moreover, in the case of ResNet-18, we aggregated the discriminators losses by computing the corresponding hypervolume as in [30], with a nadir slack equal to 2.5. All experiments were run considering a minibatch size of 64 (training each iteration took into account 64 examples from each source domain) on single GPU hardware (either an NVIDIA V100 or NVIDIA GeForce GTX 1080Ti).

- Learning rate for the task classifier and feature extractor: $\{0.01^{*,+}, 0.001^{\dagger}, 0.0005\}$;
- Learning for the domain classifiers: $\{0.0005^{*}, 0.001, 0.005^{\dagger,+}\}$;
- Weight decay: $\{0.0005^{*}, 0.001, 0.005^{\dagger+}\}$;
- Momentum: $\{0.5, 0.9^{*,\dagger,+}\}$
- Label smoothing: $\{0.0^{+}, 0.1, 0.2^{*,\dagger}\}$;
- Losses weighting (α): $\{0.35, 0.8^{*,\dagger,+}\}$;
- Random projection size: $\{1000^{*}, 3000, 3500^{\dagger}, \text{None}^{+}\}$;
- Task classifier and feature extractor learning rate warm-up iterations: $\{1, 300^{*,\dagger}, 500^{+}\}$;
- Warming-up threshold: $\{0.00001^{*}, 0.0001^{\dagger,+}, 0.001\}$;
- Learning rate schedule patience: $\{25^{+}, 60^{\dagger}, 80^{*}\}$;
- Learning rate schedule decay factor: $\{0.1^{+}, 0.3^{\dagger}, 0.5^{*}\}$.

H.2 Affective state prediction

We use SyncNet [44] as the encoder for the experiments with the SEED dataset. We follow previous work and apply a simple pre-processing that consists of clipping artifacts with amplitude 5 times higher than the mean of the channel

[6] The option of not having the random projection layer is included in the grid search.

signal and windowing data with chunks of 60 seconds. Each window was normalized to have zero mean and unit variance. For the encoder network, we adopt an one layer parameterized convolutional filter with 2 filters (designed to extract synchrony coherence which interpretable features based on the previous neuroscience literature [44]). We train all models for 100 epochs using SGD with Polyak's acceleration. The learning rate was "warmed-up" for a number of training iterations equal to 500.

The output of the encoder with size 602 is used as input for the task classifier and the domain discriminators. The domain discriminator architecture consists of a four-layer fully-connected neural network of size $602 \rightarrow$ random projection size $\rightarrow 256 \rightarrow 128 \rightarrow 2$. The random projection layer is implemented as a linear layer with weights normalized to have unitary L2-norm. The task classifier is a two-layer fully-connected network of size $602 \rightarrow 100 \rightarrow$ number of classes.

The summary of parameters is presented in the following.

- Window size: 60 seconds
- Number of filters: 2
- Filters length: 40
- Pooling size: 40
- Input drop out rate: 0.2
- Initial learning rate task classifier: 9.963e-04
- Initial learning rate discriminator: 9.963e-05
- Random projection size: 602

H.3 Proxy $\mathcal{A}$-distance estimation

We implement the domain discriminators using tree ensemble classifiers with 100 estimators. We thus report the average classification accuracy using 5-fold cross-validation independently run for each domain pair. Each domain is represented by a random sample of size 500.