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] Fauber 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 this 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 a simple 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, and those often occur in practice. Representative examples of such 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. Generalization results for this setting introduced in [2] thus showed 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 that pair of distributions is 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 distribution so that representations can be learned inducing invariance across train and target domains [3], which is

[*] Correspondence to isabelamcalbuquerque@gmail.com

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

further limiting considering practical cases, e.g. a speech recognition service cannot be trained on data obtained from every new speaker it observes.

A more general setting is often referred to as *domain generalization* [4]. In that case, it is assumed a set of distributions over the data is available at training time. At test time, however, both those 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 settings 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], learning domain-invariant representations [9], among other approaches.

In this contribution, we tackle the briefly described 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. That result yields a generalization guarantee to any distribution on the neighborhood of the "convex hull"[3] defined by the set of domains we started with. Inspired by that, 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, no data from test distributions is required, *which is a major advantage compared to more traditional domain adaptation settings* that target a specific distribution represented at training time through an unlabeled sample.

We summarize our contributions in the following:

- We introduce a set of assumptions on the data generating process tailored to the domain generalization setting which we argue are much more general than standard i.i.d requirements and more likely to hold in practice, i.e. given a data sample, it is more likely that our assumptions will hold compared to the more restrictive i.i.d. property;
- We prove a generalization bound for the risk over unseen domains and show generalization can be expected for domains on the neighborhood of a notion of convex hull of distributions observed at training time;
- Aiming to minimize the bound introduced, we devise an adversarial approach so that pairwise domain divergences are estimated and minimized. In order to do so, several practical improvements are performed on top of previously introduced approaches for domain adaption including the use of random projections prior to domains discriminators.

The remainder of this paper is organized as follows: In Section 2 we discuss past results which will be used in this work. In section 3 we define the domain generalization setting and present our main results as well as the resulting algorithm. Section 4 provides the experiments description and the respective results. Section 5 discusses related work while conclusions are drawn in Section 6 along with future directions.

2 Background

Let the data be represented by $\mathcal{X} \subset \mathbb{R}^D$, while labels are given by $\mathcal{Y}$, which would be $\{0, 1\}$ in the binary case, for instance. Examples correspond to a pair $(x, y) : x \in \mathcal{X}, y \in \mathcal{Y}$, such that $y = f(x)$, and $f : \mathcal{X} \to \mathcal{Y}$ is a deterministic labeling function.

A domain is defined as a tuple $\langle \mathcal{D}, f \rangle$ 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 $\langle \mathcal{D}, f \rangle$ as:

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

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

2.1 Generalization guarantees for domain adaptation

We now state results from the domain adaptation literature which are relevant for this work. The discussion in [10] established the theoretical foundations for studying the cross-domain generalization properties for domain adaptation

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

problems. Based on the *covariate shift assumption*, which considers that the labeling function is the same across domains (i.e. while $\mathcal{D}$ can change, f is fixed) they showed that, given a source domain $\mathcal{D}_S$ and target domain $\mathcal{D}_T$, the risk of a given hypothesis h on the target is bounded by:

$$R_T[h] \leq R_S[h] + d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T] + \lambda, \tag{2}$$

where λ corresponds to the minimal total risk over both domains which can be achieved within a given hypothesis class $\mathcal{H}$. The term $d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T]$ corresponds to the $\mathcal{H}$-divergence introduced in [11] and defined is as follows:

$$d_{\mathcal{H}}[\mathcal{D}_S, \mathcal{D}_T] = 2 \sup_{\eta \in \mathcal{H}} |\mathrm{Pr}_{x \sim \mathcal{D}_S}[\eta(x) = 1] - \mathrm{Pr}_{x \sim \mathcal{D}_T}[\eta(x) = 1]|. \tag{3}$$

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.

In [12], an extension of the bound above was presented for the case where multiple source domains are available at training time. Given N_S source domains D_S^i, $i \in [N_S]$, and a mixture of the source domains $\sum_{i=1}^{N_S} \alpha_i \mathcal{D}_S^i(\cdot)$, the risk $R_T[h]$ on the target domain is bounded by:

$$R_T[h] \leq \sum_{i=1}^{N_S} \alpha_i \left(R_S^i[h] + \frac{1}{2} d_{\mathcal{H}}[\mathcal{D}_T, \mathcal{D}_S^i] \right) + \lambda_\alpha, \tag{4}$$

where λ_α is the minimum total risk, i.e. the sum of the risks measured on the target and the mixture of the sources, one can get within the considered hypothesis class, and the set α_i of mixture coefficients is such that $\alpha_i \in [0, 1], i \in [N_S], \sum_{i=1}^{N_S} \alpha_i = 1$.

3 Learning domain agnostic representations for domain generalization

3.1 Formalizing domain generalization

We start by defining a set of assumptions we introduce 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 $(x^m \sim \mathfrak{D}^m, f(x^m))$. 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.

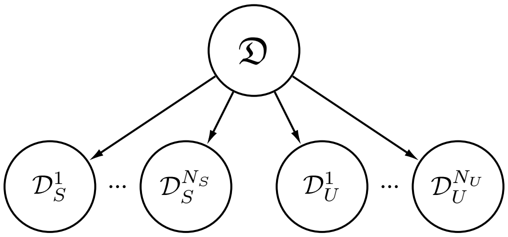


Figure 1: Illustration of the meta-distribution $\mathfrak{D}$ composed by the source and unseen domains.

Figure 1 illustrates the general model of data generating processes by representing the meta-distribution along with possible domains. We remark that, 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 $\mathcal{D}_U^j$, $j \in [N_U]$.

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:

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} R_{\mathfrak{D}}[h], \quad R_{\mathfrak{D}}[h] = \mathbb{E}_{\mathcal{D} \sim \mathfrak{D}}[\mathbb{E}_{x \sim \mathcal{D}}[\ell(h(x), f(x))]]. \tag{5}$$

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 that 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 [13].

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: $\Lambda_S = \{\bar{\mathcal{D}} : \bar{\mathcal{D}}(\cdot) = \sum_{i=1}^{N_S} \pi_i \mathcal{D}_S^i(\cdot), \pi_i \in \Delta_{N_S}\}$. The following proposition 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.

Proposition 1 *(Bounding the $\mathcal{H}$-divergence between domains in the convex hull). 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. We further introduce $\bar{\mathcal{D}}_U^j$, the element of Λ_S which is closest to $\mathcal{D}_U^j$, i.e. $\bar{\mathcal{D}}_U^j$ is given by $\operatorname{argmin}_{\pi_1,\ldots,\pi_{N_s}} d_{\mathcal{H}} \left[\mathcal{D}_U^j, \sum_{i=1}^{N_S} \pi_{i,j} \mathcal{D}_S^i\right]$. Based on Proposition 1, we derive a generalization bound for the risk $R_U^j[h]$ in terms of ϵ and $d_{\mathcal{H}}[\bar{\mathcal{D}}_U^j, \mathcal{D}_U^j]$:

Proposition 2 *(Generalization to unseen domains). Let $d_{\mathcal{H}}[\bar{\mathcal{D}}_U^j, \mathcal{D}_U^j] = \gamma$. Given the previous setup and assumptions, the following inequality holds for the risk $R_U^j[h]$, $\forall h \in \mathcal{H}$ for **any** domain $\mathcal{D}_U^j$:*

$$R_U^j[h] \leq \sum_{i=1}^{N_S} \pi_{i,j} R_S^i[h] + \frac{\gamma + \epsilon}{2} + \lambda_{\pi_j} \tag{7}$$

where λ_{π_j} is the minimum sum of the risks achieved by some $h \in \mathcal{H}$ on $\mathcal{D}_U^j$ and $\bar{\mathcal{D}}_U^j$.

Proof. C.f. supplementary material.

The result discussed on the above can be used to define algorithms 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 further highlight that such results also provide insights regarding the importance of acquiring diverse datasets in practice when targeting domain generalization. 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. In this case, $\gamma = 0$ and the bound stated in Proposition 2 is tighter. 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. Another practical aspect worth remarking is that, even though our domain generalization setting is more general than ERM, Proposition 2 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, for example, speech recognition, where different speakers or channels can be viewed as different domains.

3.4 Practical contributions

Motivated by the previous results, we propose to estimate and minimize ϵ along with 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,

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

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 $\boldsymbol{\epsilon}$: Previous work on domain adaptation introduced strategies based on minimizing the empirical $\mathcal{H}$-divergence between sources and a given target domain [3, 12]. Instead, as per the discussion following Proposition 2, 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[5] $\sum_{l \neq k} d_{\mathcal{H}}[\mathcal{D}_S^k, \mathcal{D}_S^l]$, and improves the method to a number of $\mathcal{H}$-divergence estimators linear on N_S.

Training: The proposed approach contains three main modules: 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:

$$\min_{\phi, \theta_C} \max_{\theta_1, \dots, \theta_{N_S}} \mathcal{L}_C(C(E(x;\phi);\theta_C), y_C) - \sum_{k=1}^{N_S} \mathcal{L}_k(D_k(E(x;\phi);\theta_k), y_k), \tag{8}$$

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.

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
```

We empirically found it helpful to augment our domain generalization approach with strategies for stabilizing the training of generative adversarial networks with multiple discriminators [14, 15]. A random projection layer is then introduced in the input of each domain discriminator with the goal of making examples from different distributions harder to be distinguished, and the negative log hypervolume is used instead of the summation in the game represented in (8)[5]. We refer to the proposed approach as G2DM (**G**eneralizing to unseen **D**omains via **D**istribution **M**atching).

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 [12] 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.

[5] See supplementary material for details.

4 Experiments

Our empirical evaluation aims to answer the following questions: (i) Can we do better than standard ERM under i.i.d. assumptions by using information of source domains only? (ii) where does G2DM's performance stand in comparison to previous work? (iii) is G2DM indeed enforcing distribution matching? (iv) what is the effect on the resulting performance of employing different stopping criteria? We start the investigation performing experiments on two domain generalization benchmarks (VLCS [16] and PACS [17]) which consist of object recognition tasks. We then evaluate G2DM on a real-world task that involves classifying electroencephalography (EEG) time series for affective state prediction. Additionally, in the supplementary material we provide results showing the impact of the random projection layer size and the number of source domains on final performance.

4.1 VLCS and PACS benchmarks

The VLCS benchmark is composed of 5 overlapping classes of objects obtained from the VOC2007 [18], LabelMe [19], Caltech-101 [20], and SUN [21] datasets. The object recognition benchmark referred to as PACS, in turn, consists of images distributed into 7 classes originated from four different datasets: Photo (P), Art painting (A), Cartoon (C), and Sketch (S). Details regarding each benchmark can be found in the supplementary material. We compare the performance of our proposed approach with a model trained with ERM over all source domains with no mechanisms to enforce domain generalization. Moreover, we consider the recently introduced invariant risk minimization (IRM) strategy [22] and include results reported in the literature achieved by Epi-FCR [23], JiGen [8] along with the ERM results they provided (referred to as ERM-JiGen), and MMD-AAE [24]. Finally, the adaptation of DANN for domain generalization reported in [23] was also considered. All such methods have an encoder implemented as the convolutional stack of AlexNet [25] and the weights are initialized from the pre-trained model on ImageNet [26]. Further implementation details necessary for reproducing the results reported herein 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 [17] such as LabelMe and Sketch.

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

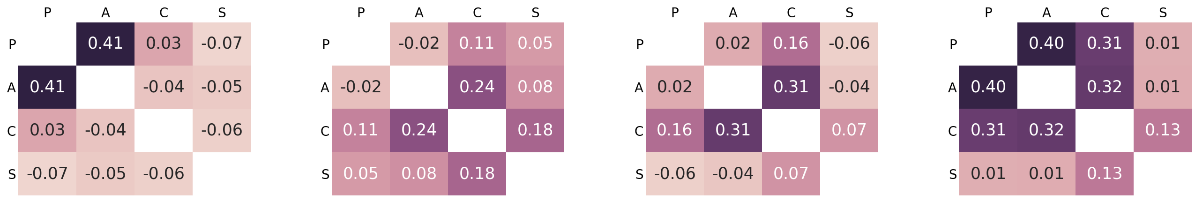
Unseen domain (→)	V	L	C	S	Average
DANN [23]	66.40	64.00	92.60	63.60	71.70
MMD-AAE [24]	67.70	62.60	94.40	64.40	72.28
Epi-FCR [23]	67.10	64.30	94.10	65.90	72.90
JiGen [8]	70.62	60.90	96.93	64.30	73.19
ERM - JiGen [8]	71.96	59.18	96.93	62.57	72.66
IRM	72.16	62.36	**98.35**	67.82	75.17
ERM	**73.44**	60.44	97.88	67.92	74.92
G2DM	71.14	**67.63**	95.52	**69.37**	**75.92**

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

Unseen domain (→)	P	A	C	S	Average
DANN [23]	88.10	63.20	67.50	57.00	69.00
Epi-FCR [23]	86.10	64.70	72.30	65.00	72.00
JiGen [8]	89.00	**67.63**	71.71	65.18	73.38
ERM - JiGen [8]	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 Estimating pairwise $\mathcal{H}$-divergences

We investigate whether cross-domain $\mathcal{H}$-divergences are being in fact reduced by G2DM. We use ERM as a baseline for comparison since 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 2 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 domain classification accuracy between the same domains. We observe that apart from the case where Photo is the unseen domain, G2DM was able to better match most of the source domains, thus yielding a smaller ϵ which favours generalization. Interestingly, we also noticed that the estimated pairwise $\mathcal{H}$-divergence between the unseen domain and sources also decreased in most of the cases even though G2DM did not have access to data from the unseen domain at training time to explicitly match those distributions. This effect is explained by the fact that the $\mathcal{H}$-divergence satisfies the triangle inequality (c.f. Eq. 13 in the supplementary material), which can be used to show that an upper-bound for the discrepancy between the unseen domain and any source gets tighter once ϵ decreases.



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

Figure 2: 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.

4.1.2 Domain generalization in practical scenarios

Results of previous experiments correspond to an optimistic scenario where target data is available for at least selecting the best performing model. This is not the case in practice since varying target distributions might appear. In Table 3, we compare results obtained further considering stopping criteria that only use information from the source domains, such as validation accuracy on the source domains and training task loss. For comparison, we also present the performance reported by [9] for CIDDG, since a stopping criterion using solely data from source domains was employed in that case. We notice that, when using the task loss as stopping criterion, our strategy outperforms CIDDG for almost all domains while its 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 [27], since it has shown promising results in recent work [8]. We compare our approach with JiGen[6] adopting the same previous stopping criteria for both methods. 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 mostly on the results obtained with AlexNet, we remark that different criteria might be too optimistic/pessimistic, and as such, one practical recommendation we can draw from our results is that the best methodology to be adopted when studying domain generalization strategies is to *report their performance across different stopping criteria.*

4.2 Real-world case: Affective state prediction from multi-variate time-series

We proceed to evaluate the proposed approach beyond domain generalization benchmarks. The goal of the selected task is to perform affective state estimation based on EEG signals from multiple subjects. EEG is a modality known to present high variability across different individuals given the same stimuli due to factors such as anatomic and environment variations [28]. Therefore, since it cannot be assumed data from different individuals are identically distributed, this scenario consists in a challenging test bed for domain generalization approaches. We use the SEED dataset [29], which is composed of 62-channel EEG signals from 15 participants. During the data collection, subjects are asked to rate video clips extracted from movies as positive, neutral, or negative. We follow [30] and use the architecture described in [31] for both G2DM and ERM. We consider each subject as a different domain and perform leave-one-subject-out evaluation. For each subject left out for testing, we use 10 out of remaining 14 domains for training and use the other 4 as validation data. Similarly to our previous experiments, for each test domain we perform 3 independent runs. We report in Table 4 the affective state prediction accuracy (%) averaged across all unseen subjects and runs. Under **source data validation**, the performance reported was computed on the epoch of highest accuracy on the source domains at the validation partition. The results under **semi-privileged** were obtained on the epoch of highest accuracy on the unseen subject. The comparison between G2DM and ERM shows that using G2DM to leverage domain information (which in this case comes with no additional effort at the data collection) yields an improvement in performance for both stopping criteria. 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 target domain (unknown in our case) is used to adapt representations to yield subject-specific models.

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

When comparing the DA strategies with our 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 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. Proposition 2).

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

Method	Criterion	P	A	C	S	Average
AlexNet						
CIDDG [9]	From [9]	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

Table 4: Average accuracy (%) on the SEED dataset across 15 subjects. Privileged baselines have access to unseen domain data.

Setting	Method	Average acc. (%)
DG	*Source data validation*	
	ERM	51.98
	G2DM	**55.77**
	Semi-privileged	
	ERM	56.82
	G2DM	**60.26**
	Privileged baselines	
DA	DAN [32, 30]	50.28
	DANN [3, 30]	55.87
	MDAN [12, 30]	56.65
	MDMN [30]	60.59

5 Related work

In [8], authors proposed to enforce generalization to unseen domains by adding a regularization term that depends on a self-supervised task. Other work proposed to enforce domain generalization with adversarial approaches. This is the case of CIDDG [9], 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, MMD-AAE [24], proposed an approach that relies on an adversarial autoencoder and a maximum mean discrepancy penalty to remove domain-specific information. Recent approaches also proposed to simulate domain-shifts at training time by splitting the source domains into meta-train and meta-test sets [7, 33, 34] or by proposing an episodic training approach as in Epi-FCR [23]. Previous work also included strategies based on learning domain-invariant representations [4, 22], data augmentation [6, 5], and on decomposing the model's parameters into domain-agnostic and domain-specific components [17]. 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 [35], a setting where multiple domains are available at training time and test data is drawn from the same distributions seen during training [36], also leveraged $\mathcal{H}$-divergence minimization to derive an adversarial approach.

6 Conclusion

We tackled the domain generalization setting and showed 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 that making use of domain information enables a boost in performance compared to standard settings relying on i.i.d. requirements. Moreover, the introduced approach outperformed recent methods which also leverage domain labels. We further showed such approach to yield strong results on a realistic setting, with performance comparable to privileged systems tailored to test distributions. In future work, we intend to investigate if the introduced assumptions on the data generating process can yield PAC-like results for domain complexity in a meta-distribution-agnostic fashion, i.e. we intend to assess questions such as: how many source domains are needed to guarantee low meta-risk with high probability?

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] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[27] 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.

[28] 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.

[29] 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.

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

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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 Proposition 1

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:

$$\begin{aligned}
d_{\mathcal{H}}[\mathcal{D}'_U, \mathcal{D}''_U] =& 2 \sup_{h \in \mathcal{H}} |\mathrm{Pr}_{x \sim \mathcal{D}'_U}[h(x) = 1] - \mathrm{Pr}_{x \sim \mathcal{D}''_U}[h(x) = 1]|, \\
=& 2 \sup_{h \in \mathcal{H}} |\mathbb{E}_{x \sim \mathcal{D}'_U}[\mathbf{I}(h(x))] - \mathbb{E}_{x \sim \mathcal{D}''_U}[\mathbf{I}(h(x))]|, \\
=& 2 \sup_{h \in \mathcal{H}} \left| \int_{\Omega} \mathcal{D}'_U(x)\mathbf{I}(h(x))dx - \int_{\Omega} \mathcal{D}''_U(x)\mathbf{I}(h(x))dx \right|, \\
=& 2 \sup_{h \in \mathcal{H}} \left| \int_{\Omega} \sum_{k=1}^{N_S} \pi_k \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_{\Omega} \sum_{l=1}^{N_S} \pi_l \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right|, \\
=& 2 \sup_{h \in \mathcal{H}} \left| \int_{\Omega} \sum_{l=1}^{N_S} \sum_{k=1}^{N_S} \pi_l \pi_k \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_{\Omega} \sum_{l=1}^{N_S} \sum_{k=1}^{N_S} \pi_l \pi_k \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right|, \\
=& 2 \sup_{h \in \mathcal{H}} \left| \sum_{l=1}^{N_S} \sum_{k=1}^{N_S} \pi_l \pi_k \left(\int_{\Omega} \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_{\Omega} \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right) \right|.
\end{aligned} \tag{9}$$

Using the triangle inequality, we can write:

$$d_{\mathcal{H}}[\mathcal{D}'_U, \mathcal{D}''_U] \leq 2 \sup_{h \in \mathcal{H}} \sum_{l=1}^{N_S} \sum_{k=1}^{N_S} \pi_l \pi_k \left| \int_{\Omega} \mathcal{D}_S^k(x)\mathbf{I}(h(x))dx - \int_{\Omega} \mathcal{D}_S^l(x)\mathbf{I}(h(x))dx \right|. \tag{10}$$

Finally, using the sub-additivity of the sup:

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

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 Proposition 2

Recall the result from [12] for the multi-source domain adaptation setting (for the sake of clarity, stated here replacing the target domain by an unseen domain according to our notation):

$$R_U[h] \leq \sum_{i=1}^{N_S} \alpha_i \left(R_S^i[h] + \frac{1}{2} d_{\mathcal{H}}[\mathcal{D}_U, \mathcal{D}_S^i] \right) + \lambda_\alpha. \tag{12}$$

Using the triangle inequality for the $\mathcal{H}$-divergence, we can bound the $\mathcal{H}$-divergence between an unseen domain $\mathcal{D}_U^j$ and a source domain $\mathcal{D}_S^i$, $d_{\mathcal{H}}[\mathcal{D}_U^j, \mathcal{D}_S^i]$ by:

$$\begin{aligned}
d_{\mathcal{H}}[\mathcal{D}_U^j, \mathcal{D}_S^i] &\leq d_{\mathcal{H}}[\mathcal{D}_U^j, \bar{\mathcal{D}}_U^j] + d_{\mathcal{H}}[\bar{\mathcal{D}}_U^j, \mathcal{D}_S^i] \\
&\leq \gamma + \epsilon,
\end{aligned} \tag{13}$$

where γ is the $\mathcal{H}$-divergence between $\mathcal{D}_U^j$ and the convex-hull of the sources, i.e. $\gamma = d_{\mathcal{H}}[\mathcal{D}_U^j, \bar{\mathcal{D}}_U^j]$ such that $\bar{\mathcal{D}}_U^j = \mathrm{argmin}_{\pi_1, \ldots, \pi_{N_s}} d_{\mathcal{H}} \left[\mathcal{D}_U^j, \sum_{i=1}^{N_S} \pi_{i,j} \mathcal{D}_S^i \right]$.

We can now choose each α_i's to correspond to the $\pi_{i,j}$'s and re-write Eq. 12 for an unseen domain $\mathcal{D}_U^j$ as

$$R_U^j[h] \leq \sum_{i=1}^{N_S} \pi_{i,j} \left(R_S^i[h] + \frac{1}{2} d_{\mathcal{H}}[\mathcal{D}_U^j, \mathcal{D}_S^i] \right) + \lambda_{\pi_j}. \tag{14}$$

Using Eq. 13, we can upper-bound $\sum_{i=1}^{N_S} \pi_{i,j} d_{\mathcal{H}}[\mathcal{D}_U, \mathcal{D}_S^i]$ by $\gamma + \epsilon$, which gives

$$R_U^j[h] \leq \sum_{i=1}^{N_S} \pi_{i,j} R_S^i[h] + \frac{\gamma + \epsilon}{2} + \lambda_{\pi_j}. \qquad \square \tag{15}$$

C 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:

$$\begin{aligned} \mathcal{L}_1 &= \frac{1}{3M} \sum_{i=1}^{3M} \ell(D_1(x_i), y_1), \\ &= \frac{1}{M} \sum_{i=1}^{M} \ell(D_1(x_i), y_1) + \frac{1}{M} \sum_{i=M+1}^{2M} \ell(D_1(x_i), y_1) + \frac{1}{M} \sum_{i=2M+1}^{3M} \ell(D_1(x_i), y_1), \end{aligned} \tag{16}$$

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:

$$\begin{aligned} \mathcal{L}_1 = \frac{1}{M} \sum_{i=1}^{M/2} \ell(D_1(x_i), 1) + \frac{1}{M} \sum_{i=M+1}^{2M} \ell(D_1(x_i), 0) \\ + \frac{1}{M} \sum_{i=\frac{M}{2}+1}^{M} \ell(D_1(x_i), 1) + \frac{1}{M} \sum_{i=2M+1}^{3M} \ell(D_1(x_i), 0). \end{aligned} \tag{17}$$

The first two terms from Eq.17 account for $d_{\mathcal{H}}[\mathcal{D}_1, \mathcal{D}_2]$ and the last two terms account for $d_{\mathcal{H}}[\mathcal{D}_1, \mathcal{D}_3]$.

D Illustration

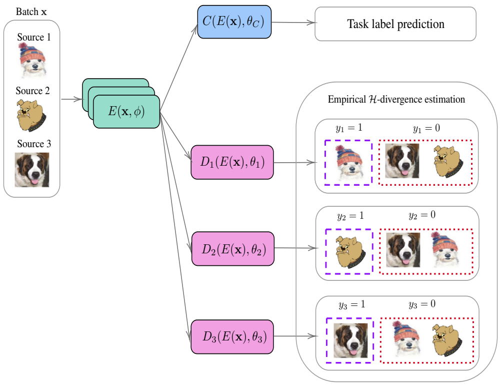

Batch x
Source 1
Source 2
Source 3
$E(\mathbf{x}, \phi)$
$C(E(\mathbf{x}), \theta_C)$
Task label prediction
$D_1(E(\mathbf{x}), \theta_1)$
$D_2(E(\mathbf{x}), \theta_2)$
$D_3(E(\mathbf{x}), \theta_3)$
Empirical $\mathcal{H}$-divergence estimation
$y_1 = 1$
$y_1 = 0$
$y_2 = 1$
$y_2 = 0$
$y_3 = 1$
$y_3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

Figure 3: Proposed approach illustration.

E Extra experiments

E.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.

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	-	-

E.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 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 4 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.

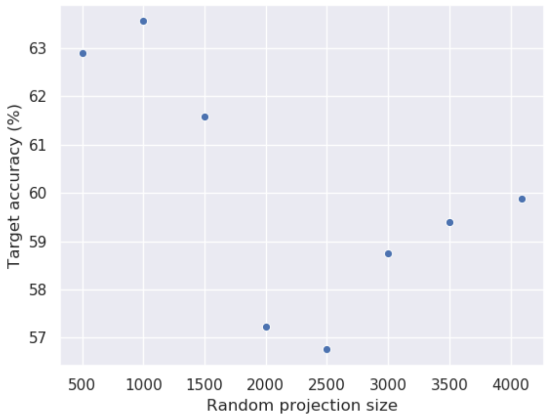


Figure 4: Accuracy obtained on the PACS benchmark using Sketch as target domain.

F 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.

G Implementation details

G.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 [25] and ResNet-18 [27] 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 [17, 23], we use models pre-trained on the ILSVRC dataset [37] as initialization. For fair comparison, all models we implemented were given a budget of 200 epochs. We use label smoothing [38] 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[7], 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 Appendix. 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 [15], 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: $\{0.0005^*, 0.001, 0.005^{\dagger,+}\}$;

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

- 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: $\{0.1^{+}, 0.3^{\dagger}, 0.5^{*}\}$.

G.2 Affective state prediction

We use SyncNet [31] 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 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 [31]). 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

G.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.