GATED DOMAIN UNITS FOR MULTI-SOURCE DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Distribution shift (DS) is a common problem that deteriorates the performance of learning machines. To tackle this problem, we postulate that real-world distributions are composed of elementary distributions that remain invariant across different environments. We call this an *invariant elementary distribution* (I.E.D.) assumption. The I.E.D. assumption implies an invariant structure in the solution space that enables knowledge transfer to unseen domains. To exploit this property in domain generalization (DG), we developed a modular neural network layer that consists of Gated Domain Units (GDUs). Each GDU learns an embedding of an individual elementary distribution that allows us to encode the domain similarities during the training. During inference, the GDUs compute similarities between an observation and each of the corresponding elementary distributions which are then used to form a weighted ensemble of learning machines. Because our layer is trained with backpropagation, it can naturally be integrated into existing deep learning frameworks. Our evaluation on image, text, graph, and time-series data shows a significant improvement in the performance on out-of-training target domains without domain information and any access to data from the target domains. This finding supports the practicality of the I.E.D. assumption and demonstrates that our GDUs can learn to represent these elementary distributions.

1 INTRODUCTION

A fundamental assumption in machine learning is that training and test data are independently and identically distributed (I.I.D.). This assumption ensures consistency-results from statistical learning theory, meaning that the learning machine obtained from an empirical risk minimization (ERM) attains the lowest achievable risk as sample size grows (Vapnik, 1998; Schölkopf, 2019). Unfortunately, a considerable amount of research and real-world applications in the past decades has provided a staggering evidence against this assumption (Zhao et al., 2018; 2020; Ren et al., 2019; Taori et al., 2020) (see D'Amour et al. (2020) for case studies). The violation of the I.I.D. assumption is usually caused by a distribution shift (DS) and can result in inconsistent learning machines (Sugiyama & Kawanabe, 2012), implying the loss of performance guarantee of machine learning models in the real world. Therefore, to tackle DS, recent work advocates for domain generalization (DG) (Blanchard et al., 2011; Muandet et al., 2013; Li et al., 2017; 2018b; Zhou et al., 2021a). This generalization to utterly unseen domains is crucial for robust deployment of the models in practice, especially when new, unforeseeable domains emerge after model deployment. However, the most important question that DG seeks to answer is how to identify the right *invariance* that allows for generalization.

The contribution of this work is twofold. First, we advocate that real-world distributions are composed of smaller "units" called *invariant elementary distributions* that remain invariant across different domains; see Section 2.1. Second, we propose to implement this hypothesis through so-called gated domain units (GDUs). Specifically, we developed a modular neural network layer that consists of GDUs. Each GDU learns an embedding of an individual elementary domain that allows us to express the domain similarities during training. For this purpose, we adopt the theoretical framework of reproducing kernel Hilbert space (RKHS) to retrieve a geometrical representation of each distribution in the form of a kernel mean embedding (KME) without information loss (Berlinet & Thomas-Agnan, 2004; Smola et al., 2007; Sriperumbudur et al., 2010; Muandet et al., 2017). This representation accommodates methods based on analytical geometry to measure similarities between distributions.

We show that these similarity measures can be learned and utilized to improve the generalization capability of deep learning models to previously unseen domains.

The remainder of this paper is organized as follows: Our theoretical framework is laid out in Section 2 with our modular DG layer implementation shown in Section 3. In Section 4, we outline related work. Our experimental evaluations are presented in Section 5. Finally, we discuss potential limitations of our approach and future work in Section 6.

2 Domain Generalization with Invariant Elementary Distributions

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output space, with a joint distribution $\mathbb{P}$. In the multi-source DG setting, we are given a set of D labeled source datasets $\{\mathcal{D}_i^s\}_{i=1}^D$ with $\mathcal{D}_i^s \subseteq \mathcal{X} \times \mathcal{Y}$. Each of the source datasets is assumed to be I.I.D. generated by a joint distribution $\mathbb{P}_i^s$ with support on $\mathcal{X} \times \mathcal{Y}$, henceforth denoted *domain*. The set of probability measures with support on $\mathcal{X} \times \mathcal{Y}$ is denoted by $\mathcal{P}$. In general, we aim to minimize the empirical risk, see Section 3.3 for details. Important notation is summarized in Table 1.

2.1 Invariant Elementary Distributions

A multi-source dataset $\mathcal{D}^s$ comprises the merged individual source datasets $\{\mathcal{D}_j^s\}_{j=1}^D$. Similar to Mansour et al. (2012), Albuquerque et al. (2019), and Hoffman et al. (2018a), we assume that the distribution of the source dataset can be described as a convex combination $\mathbb{P}^s = \sum_{j=1}^D \alpha_j^s \mathbb{P}_j^s$ where $\alpha^s = (\alpha_1^s, \ldots, \alpha_D^s)$ is an element of the probability simplex, i.e., $\alpha^s \in \Delta^D := \{\alpha \in \mathbb{R}^D \mid \alpha_j \geq 0 \wedge \sum_{j=1}^D \alpha_j = 1\}$. In other words, α_j quantifies the contribution of each individual source domain to the combined source domain.

Table 1: Important notation

K	number of elementary distributions
M	number of elementary domain bases
N	number of basis vectors
$\mathbb{P}^s$	combined multi-source distribution
$\mathbb{P}_j^s$	j-th single-source distribution
$\mathbb{P}_j$	j-th elementary distribution
V_j	j-th domain basis
v_k^j	k-th vector in V_j
α_j^s	coefficient for $\mathbb{P}_j^s$
α_j	coefficient for $\mathbb{P}_j$
β_{ij}	coefficient for sample x_i and μ_{V_j}

In contrast to Mansour et al. (2009; 2012); Hoffman et al. (2018a), we generalize their problem descriptions: We express the distribution of each domain as a convex combination of K elementary distributions $\{\mathbb{P}_j\}_{j=1}^K \subset \mathcal{P}$, meaning that $\mathbb{P}^s = \sum_{j=1}^K \alpha_j \mathbb{P}_j$ where $\alpha \in \Delta^K$. Our main assumption is that *these elementary distributions remain invariant across the domains*. The advantage is that we can find an invariant subspace at a more elementary level, as opposed to when we consider the source domains as some sort of basis for all unseen domain. Figure 1 illustrates this idea.

Theoretically speaking, the I.E.D assumption is appealing because it implies the invariant structure in the solution space, as shown in the following lemma. The proof is given in Appendix A.1.

Lemma 1. *Let $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ be a non-negative loss function, $\mathcal{F}$ a hypothesis space of functions $f : \mathcal{X} \to \mathcal{Y}$, and $\mathbb{P}^s(X, Y)$ a data distribution. Suppose that the I.E.D assumption holds, i.e., there exist K elementary distributions $\mathbb{P}_1, \ldots, \mathbb{P}_K$ such that any data distribution can be expressed as $\mathbb{P}^s(X, Y) = \sum_{j=1}^K \alpha_j \mathbb{P}_j(X, Y)$ for some $\boldsymbol{\alpha} \in \Delta^K$. Then, the corresponding Bayes predictor $f^* \in \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(X,Y)\sim\mathbb{P}}[\mathcal{L}(Y, f(X))]$ is Pareto-optimal with respect to a vector of elementary risk functionals $(R_1, \ldots, R_K) : \mathcal{F} \to \mathbb{R}_+^K$ where $R_j(f) := \mathbb{E}_{(X,Y)\sim\mathbb{P}_j}[\mathcal{L}(Y, f(X))]$.*

Lemma 1 implies that, under the I.E.D assumption, Bayes predictors must belong to a subspace of $\mathcal{F}$ called the Pareto set $\mathcal{F}_{\text{Pareto}} \subset \mathcal{F}$ which consists of Pareto-optimal models. The model f is said to be Pareto-optimal if there exists no $g \in \mathcal{F}$ such that $R_j(g) \geq R_j(f)$ for all $j \in \{1, \ldots, K\}$ with $R_j(g) > R_j(f)$ for some j; see, e.g., Sener & Koltun (2018, Definition 1). In other words, the I.E.D assumption allows us to translate the invariance property of data distributions to the solution space. Unfortunately, neither the elementary distributions nor the weights $\boldsymbol{\alpha}$ are known in practice. Motivated by this theoretical insight, our DG layer presented in Section 3 is designed to uncover them from a multi-source training dataset $\mathcal{D}^s$. While Lemma 1 shows the theoretical appeal of the

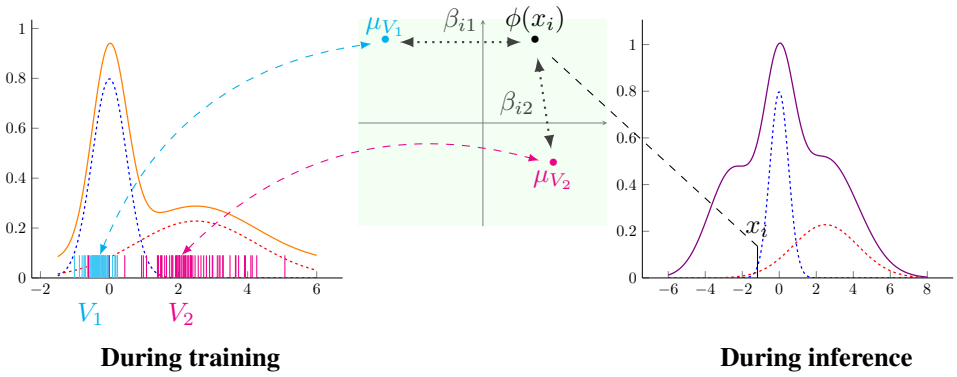

Figure 1: A visualization of an "invariant elementary distribution (I.E.D.)" assumption for domain generalization (DG): the observed data distributions (orange and violet) are composed of the same set of *unobserved* elementary distributions (blue and red) that remain invariant across different domains. Hence, the first challenge during the training phase (left panel) is to extract these elementary distributions from the observed data (orange). The unobserved elementary distributions are represented by the elementary bases V_1 and V_2 (cyan and pink). The second challenge during the inference phase (right panel) is to create a weighted ensemble of learning machines that utilize the similarities between the embedding of the unseen observation $\phi(x_i)$ and the embeddings of these distributions μ_{V_1} and μ_{V_2} in the RKHS $\mathcal{H}$ (green rectangle) as weights β_{i1} and β_{i2}.

I.E.D. assumption, we discuss below a situation in which it might hold in practice. The limitations will be discussed later in Section 6.

Real-world example. In this work, we postulate that the elementary domain bases are the invariant subspaces that allow us to generalize to unseen domains. In practice, the question arises if and when elementary domains evolve. Consider that we aim to learn to predict the risk of developing Diabetes from laboratory data from Europe and then infer the risk from data from the United States of America. Naturally, factors influencing the data-generating process may change, such as the level of physical activity and nutritional habits. While, to a certain degree, these common factors remain invariant across continents, each of these factors' contributions may differ. In terms of our assumptions, we model each of these factors with a corresponding elementary distribution $\mathbb{P}_j$. For a previously unseen individual, we can then determine the coefficients α_j^s and quantify each factor's contribution without any information about the individual's origin.

2.2 Kernel Mean Embedding of Distributions

We leverage the KME of distributions (Berlinet & Thomas-Agnan, 2004; Smola et al., 2007; Muandet et al., 2017) to discover the elementary distributions and evaluate similarities between them. Let $\mathcal{H}$ be a reproducing kernel Hilbert space (RKHS) of real-valued functions on $\mathcal{X}$ with a reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (Schölkopf et al., 2001). The KME of a probability measure $\mathbb{P} \in \mathcal{P}$ in the RKHS $\mathcal{H}$ is defined by a mapping $\phi(\mathbb{P}) = \mu_{\mathbb{P}} := \int_{\mathcal{X}} k(\mathbf{x}, \cdot)\, d\mathbb{P}(\mathbf{x})$. We assume that the kernel k is characteristic, i.e., the mapping $\mu_{\mathbb{P}}$ is injective (Fukumizu et al., 2004; Sriperumbudur et al., 2008). Theoretically, this essential assumption ensures that there is no information loss when mapping the distribution into $\mathcal{H}$. Given the samples $\{x_1, \ldots, x_n\}$ generated I.I.D. from $\mathbb{P}$, $\mu_{\mathbb{P}}$ can be approximated by the empirical KME $\hat{\mu}_{\mathbb{P}} = (1/n) \sum_{i=1}^{n} k(x_i, \cdot) = (1/n) \sum_{i=1}^{n} \phi(x_i)$. We refer non-expert readers to Muandet et al. (2017) for a thorough review on this topic.

Challenges. Figure 1 depicts two challenges that come with our assumption of elementary distributions. First, since we do not have access to the samples from the hidden elementary distributions, the elementary KME cannot be estimated directly from the samples at hand. To overcome this challenge, we instead seek a proxy KME $\mu_{V_j} := (1/N) \sum_{k=1}^{N} \phi(v_k^j) = (1/N) \sum_{k=1}^{N} k(v_k^j, \cdot)$ for each elementary KME $\mu_{\mathbb{P}_j}$ from a domain basis V_j, where $V_j = \{v_1^j, \ldots, v_N^j\} \subseteq \mathcal{X}$ for all $j \in \{1, \ldots, M\}$. Hence, the KME μ_{V_j} can be interpreted as the KME of the empirical probability measure $\hat{\mathbb{P}}_{V_j} = (1/N) \sum_{k=1}^{N} \delta_{v_k^j}$. Here, we assume that $M = K$. The sets V_j are referred to

as *elementary domain basis*. Intuitively, the elementary domain basis $V_1, \dots, V_M$ represents each elementary distribution by a set of vectors that mimic samples generated from the corresponding distribution. In Figure 1, V_1 and V_2 as well as their mapping in $\mathcal{H}$ visualize this first challenge.

The second challenge is the objective of learning the unknown similarity between a single sample x_i and an elementary domain V_j, which we denote by β_{ij}. Considering the advantage of KMEs, that is to tackle this challenge from a geometrical viewpoint, we quantify similarities between KMEs. For example, in Figure 1, the similarity between $\phi(x_i)$ and μ_{V_1} (β_{i1}) and μ_{V_2} (β_{i2}) could be quantified as their distance or angle. These similarity coefficients enable our *Domain Generalization Layer* to represent a convex combination of elementary domain-specific learning machines, commonly known as ensembles. We introduce our layer in the following Section 3

3 Domain Generalization Layer

This section aims to transfer the theoretical ideas presented in Section 2 into a deep learning framework. For the purpose of implementation, let $x \in \mathbb{R}^{h \times w}$ denote the input data point and $h_\xi : \mathbb{R}^{h \times w} \to \mathbb{R}^e$ the feature extractor (FE) that maps the input into a low-dimensional representation $\tilde{x} \in \mathbb{R}^e$. Then the prediction layer $g_\theta : \mathbb{R}^e \to \mathcal{Y}$ infers the label y. To tackle the DG problem, we introduce a layer module called the gated domain unit (GDU). A GDU consists of three main components: (1) a similarity function $\gamma : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ that is the same for all elementary domains, (2) an elementary basis V_j and (3) a learning machine $f(\tilde{x}, \theta_j)$ for each elementary domain $j \in \{1, \dots, M\}$. The architecture of the layer proposed herein is depicted in Figure 2.

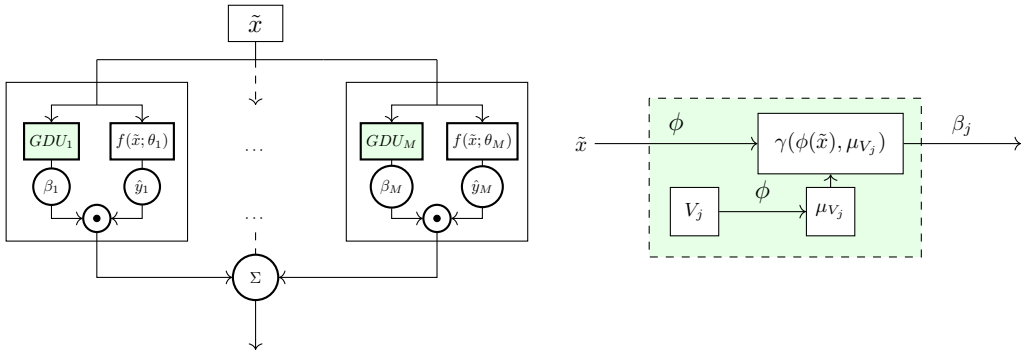

Figure 2: Visualization of the DG layer (left panel) and its main component, the GDU (right panel). The DG layer consists of several GDUs that represent the elementary distributions. During training, these GDUs learn the elementary domain bases $V_1, \dots, V_M$ that approximate these distributions.

Essentially, the process is as follows: First, the j-th GDU takes $\tilde{x}_i$ as an input and yields β_{ij} as an output. The KME of each domain basis V_j is required in order to apply γ to compute similarity between $\tilde{x}_i$ and V_j. These KMEs are obtained by $\phi(V_j) := \mu_{V_j} = (1/N) \sum_{k=1}^{N} \phi(v_k^j) = (1/N) \sum_{k=1}^{N} k(v_k^j, \cdot)$. The GDU, therefore, has the task to allocate coefficients β_{ij} for each elementary domain based on a similarity function γ. The function γ outputs the $\beta_{ij} = \gamma(\phi(\tilde{x}_i), \mu_{V_j})$ coefficients that in turn represent similarities between the KME of both, the corresponding domain basis V_j and the input $\tilde{x}_i$. Theoretically speaking, μ_{V_j} and the feature mapping $\phi(\tilde{x}_i)$ are elements of the associated RKHS $\mathcal{H}$, which allow us to evaluate similarities of non-linear features in a higher dimensional feature space. Each GDU is then connected to a learning machine $f(\tilde{x}_i, \theta_j)$ that yields an elementary domain-specific inference. The final prediction of the layer is then an ensemble of these learning machines $g_\theta(\tilde{x}_i) = \sum_{j=1}^{M} \beta_{ij} f(\tilde{x}_i, \theta_j)$ where $\theta = (\theta_1, \dots, \theta_M)$. In Figure 2, we give an overview of how data is processed and information is stored in the GDU.

In summary, GDUs leverage the invariant elementary distribution (I.E.D.) assumption and represent our algorithmic contribution: The elementary domain bases are stored as weights in the layer. Storing information as a weight matrix (i.e., domain memory) allows to learn the elementary domain bases efficiently using backpropagation. Hence, we avoid the dependency on problem-adaptive methods (e.g., domain-adversarial training) and domain information (e.g., domain labels).

3.1 DOMAIN SIMILARITY MEASURES

For the similarity function γ, we consider two similarity measures $H(\phi(\tilde{x}), \mu_{V_j})$, namely the cosine similarity (CS) (Kim et al., 2019) and maximum mean discrepancy (MMD) (Borgwardt et al., 2006; Gretton et al., 2012). To ensure that the resulting coefficients β_i lie on the probability simplex, we apply the kernel softmax function (Gao et al., 2019) and interpret its output as the similarity between an observation $\tilde{x}$ and an elementary domain basis V_i. We get

$$\beta_{ij} = \gamma(\phi(\tilde{x}_i), \mu_{V_j}) = \frac{\exp\big(\kappa H(\phi(\tilde{x}_i), \mu_{V_j})\big)}{\sum_{k=1}^{M} \exp\big(\kappa H(\phi(\tilde{x}_i), \mu_{V_k})\big)}, \tag{1}$$

where $\kappa > 0$ is a positive softness parameter for the kernel softmax. Geometrically speaking, these similarities correspond to the angle and distance of two KMEs in the RKHS $\mathcal{H}$. The function ϕ maps the observation $\tilde{x}$ and domain basis V_j into $\mathcal{H}$ meaning that $\phi(\tilde{x}) = \mu_{\delta_{\tilde{x}}} = k(\tilde{x}, \cdot)$ is the KME of a Dirac measure $\delta_{\tilde{x}}$ and $\phi(V_j) = \mu_{V_j} = (1/N) \sum_{k=1}^{N} k(v_k^j, \cdot)$.

CS. The CS function $H(\phi(\tilde{x}_i), \mu_{V_j}) = \frac{\langle \phi(\tilde{x}_i), \mu_{V_j} \rangle_{\mathcal{H}}}{\|\phi(\tilde{x}_i)\|_{\mathcal{H}} \|\mu_{V_j}\|_{\mathcal{H}}}$ is used as an angle-based similarity.

MMD. We consider the MMD for calculating a distance-based similarity measure. The distance is then given as $\|\phi(\tilde{x}_i) - \mu_{V_j}\|_{\mathcal{H}}$. Subsequently, the similarity function H is the negative MMD: $H(\phi(\tilde{x}_i), \mu_{V_j}) = -\|\phi(\tilde{x}_i) - \mu_{V_j}\|_{\mathcal{H}}$. The intuition behind the negative MMD is to put higher weights on samples that are closer to the KME of an elementary domain basis.

3.2 PROJECTION-BASED GENERALIZATION

For classification tasks, we introduce an alternative approach to infer the β_i coefficients that is based on the idea of kernel sparse coding (Gao et al., 2010; 2013). Herein the goal is to find an approximated representation of each feature mapping $\phi(\tilde{x}_i)$ using the elements of a dictionary $\{\mu_{V_j}\}_{j=1}^{M}$. This approach allows us to approximate the feature mapping with these elements by $\phi(\tilde{x}_i) \approx \sum_{j=1}^{M} \beta_{ij} \mu_{V_j}$. In contrast to the aforementioned approaches, an elementary domain KME μ_{V_j} does not necessarily represent the KME of an elementary distribution $\mu_{\mathbb{P}_j}$. Therefore, we present another approach that aims to find a set $\{\mu_{V_j}\}_{j=1}^{M}$ that permits $\mu_{\mathbb{P}^s}$ to be represented as a linear combination.

Since $\mathbb{P}$ is assumed to be a convex combination of elementary distributions, we can find a linear combination to represent $\mu_{\mathbb{P}^s}$ by the domain KMEs μ_{V_j}, as long as $\mu_{\mathbb{P}^s} \in \mathcal{H}_M := \text{span}\{\mu_{V_j} \,|\, j = 1, \ldots, M\}$. The RKHS $\mathcal{H}_M$ is a subspace of the actual RKHS $\mathcal{H}$, which allows us to represent elements of $\mathcal{H}$ at least approximately in the subspace $\mathcal{H}_M$. By keeping the $\mathcal{H}_M$ large, we gain more representative power. To make $\mathcal{H}_M$ as large as possible, we have to ensure its spanning elements are linearly independent or, even better, orthogonal. Orthogonal KMEs ensure two desirable properties. First, pairwise orthogonal elements in $\mathcal{H}_M$ guarantee no redundancy. Second, having orthogonal elements allows us to make use of the orthogonal projection. This projection geometrically yields the best approximation of $\phi(\tilde{x})$ in $\mathcal{H}_M$. In other words, we can achieve the best possible approximation of the feature mapping by using its orthogonal components (see Proposition 3.1). The orthogonal projection is given by

$$\Pi_{\mathcal{H}_M} : \mathcal{H} \to \mathcal{H}_M, \quad \phi(\tilde{x}) \mapsto \sum_{i=1}^{M} \frac{\langle \phi(\tilde{x}), \mu_{V_j} \rangle_{\mathcal{H}}}{\|\mu_{V_j}\|_{\mathcal{H}}^2} \mu_{V_j}. \tag{2}$$

Proposition 3.1. *For a KME $\mu_{\mathbb{P}}$ of a given mixture distribution $\mathbb{P}$ the following holds $\mu_{\mathbb{P}} \in$ span$\{\mu_{V_j} \,|\, V_j, \forall j = 1, \ldots, M\}$, where $\langle \mu_{V_i}, \mu_{V_j} \rangle_{\mathcal{H}} = 0, \forall i \neq j$ (i.e., the KME of the elementary domains basis are pairwise orthogonal). The value of the function $\sum_{j=1}^{M} \|\mu_{\mathbb{P}} - \beta_j \mu_{V_j}\|_{\mathcal{H}_k}^2$ is minimal if the coefficients are set as $\beta_j^* = \langle \mu_{\mathbb{P}}, \mu_{V_j} \rangle_{\mathcal{H}} / \|\mu_{V_j}\|_{\mathcal{H}}^2$.*

The Proposition 3.1 can be used to give an approximation of $\mu_{\mathbb{P}}$ by projecting it into $\mathcal{H}_M$, i.e., $\mu_{\mathbb{P}} \approx \sum_{j=1}^{M} \beta_j \mu_{V_j}$, where $\beta_j = \langle \mu_{\mathbb{P}}, \mu_{V_j} \rangle_{\mathcal{H}} / \|\mu_{V_j}\|_{\mathcal{H}}^2$. This best approximation property is the main advantage of our assumption in Proposition 3.1 (i.e., having orthogonal KME) and thus a potential advantage of projection-based DG. Appendix A.2 provides the proof of Proposition 3.1.

3.3 MODEL TRAINING

For model training, we adapt the domain adaptation (DA) framework from Zhuang et al. (2021). Thus, our learning objective function is formalized as $\mathcal{L}(g) + \lambda_D \Omega_D(\|g\|_{\mathcal{H}})$. The goal of the training can be described in terms of the two components of this function. Consider a batch of training data $\{x_1, \dots, x_b\}$, where b is the batch size. During training, we minimize the loss function $\mathcal{L}(g) = \frac{1}{b}\sum_{i=1}^{b} \mathcal{L}(\hat{y}_i, y_i) = \frac{1}{b}\sum_{i=1}^{b} \mathcal{L}(\sum_{j=1}^{M} \gamma(\phi(\tilde{x}_i), \mu_{V_j}) f_j(\tilde{x}_i), y_i)$ for an underlying task and the respective batch size. In addition, our objective is that the model learns to distinguish between different domains. Thus, the regularization Ω_D is introduced to control the domain basis. In our case, we require the regularization Ω_D to ensure that the KMEs of the elementary domain basis are able to represent the KMEs of the elementary domains. Therefore, we minimize the MMD between the feature mappings $\phi(\tilde{x}_i)$ and the associated representation $\sum_{j=1}^{M} \beta_{ij} \mu_{V_j}$. Note that $\beta_{ij} = \gamma(\phi(\tilde{x}_i), \mu_{V_j})$. Hence, the regularization $\Omega_D = \Omega_D^{OLS}$ is defined as $\Omega_D^{OLS}(\|g\|_{\mathcal{H}}) = \frac{1}{b}\sum_{i=1}^{b} \|\phi(\tilde{x}_i) - \sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2$ (see Appendix A.3 for details). The intuition is the objective to represent each feature mapping $\phi(\tilde{x}_i)$ by the domain KMEs μ_{V_j}. Thus, we try to minimize the MMD between the feature map and a combination of μ_{V_j}. The minimum of the stated regularization can be interpreted as the ordinary least square-solution of a regression-problem of $\phi(\tilde{x}_i)$ by the components of $\mathcal{H}_M$. In other words, we want to ensure that the basis V_j is contained in feature mappings $\phi(\tilde{x}_i)$.

In the particular case of projection, we want the KME of the elementary domain to be orthogonal to ensure high expressive power. For this purpose, the additional term $\Omega_D^{\perp}$ will be introduced to ensure the desired orthogonality. Considering a kernel function with $k(x, x) = 1$, orthogonality would require the Gram matrix $K_{ij} = \langle \mu_{V_i}, \mu_{V_j} \rangle_{\mathcal{H}}$ to be close to the identity matrix I. There are a variety of methods for regularizing matrices available (Xie et al., 2017; Bansal et al., 2018). A well-known method to ensure orthogonality is the soft orthogonality (SO) regularization $\Omega_D^{\perp} = \lambda \|K - I\|_F^2$ (Bansal et al., 2018). As pointed out by Bansal et al. (2018), the spectral restricted isometry property (SRIP) and mutual coherence (MC) regularization can be a promising alternative for SO and thus are additionally implemented in the DG layer. Hence, in the case of projection, the regularization is given by $\Omega_D(\|g\|_{\mathcal{H}}) = \lambda_{OLS}\Omega_D^{OLS}(\|g\|_{\mathcal{H}}) + \lambda_{ORTH}\Omega_D^{\perp}(\|g\|_{\mathcal{H}}),\ \lambda_{OLS}, \lambda_{ORTH} \geq 0$.

Lastly, sparse coding is an efficient technique to find the least possible basis to recover the data subject to a reconstruction error (Olshausen & Field, 1997). Several such applications yield strong performances, for example in the field of computer vision (Lee et al., 2007; Yang et al., 2009). Kernel sparse coding transfers the reconstruction problem of sparse coding into $\mathcal{H}$ by using the mapping ϕ, and, by applying a kernel function, the reconstruction error is quantified as the inner product (Gao et al., 2010; 2013). To ensure sparsity, we apply the L_1-norm on the coefficients β and add $\Omega_D^{L_1}(\|\gamma\|) := \|\gamma(\phi(\tilde{x}_i), \mu_{V_j})\|_1$ to the regularization term Ω_D with the corresponding coefficient λ_{L_1}. Appendix A.4 gives a visual overview of the model training.

4 RELATED WORK

DG, also known as out-of-distribution (OOD) generalization, is among the hardest problems in machine learning (Blanchard et al., 2011; Muandet et al., 2013; Arjovsky et al., 2019). In contrast, DA, which predates DG and OOD problems, deals with a slightly simpler scenario in which some data from the test distribution are available (Ganin et al., 2015). Hence, based on the available data, the task is to develop learning machines that transfer knowledge learned in a source domain specifically to the target domain. Approaches pursued in DA can be grouped primarily into (1) discrepancy-based DA (Sun et al., 2016; Peng & Saenko, 2018; Ben-David et al., 2010; Fang et al., 2020; Tzeng et al., 2014; Long et al., 2015; Baktashmotlagh et al., 2016) (2) adversary-based DA (Tzeng et al., 2017; Liu & Tuzel, 2016; Ganin et al., 2015; Long et al., 2018), and (3) reconstruction-based DA (Bousmalis et al., 2016; Hoffman et al., 2018b; Kim et al., 2017; Yi et al., 2017; Zhu et al., 2017; Ghifary et al., 2014). In DA, learning the domain-invariant components requires access to unlabeled data from the target domain. Unlike problems in DA, where the observed data from the test domains can be used to find the most appropriate invariant structures (Ben-David et al., 2010), the lack thereof in DG calls for a postulation of invariant structure that will enable the OOD generalization.

To enable generalization to unseen domains without any access to data from them, researchers have made significant progress in the past decade and developed a broad spectrum of methodologies (Zhou

et al., 2021a;c; Li et al., 2019; Blanchard et al., 2011). For thorough review see, e.g., Zhou et al. (2021a); Wang et al. (2021). Existing works can be categorized into methods based on domain-invariant representation learning (Muandet et al., 2013; Li et al., 2018b;d), meta-learning (Li et al., 2018a; Balaji et al., 2018), data augmentation (Zhou et al., 2020), to name a few. Another recent stream of research from a causal perspective includes invariant risk minimization (Arjovsky et al., 2019), invariant causal prediction (Peters et al., 2016), and causal representation learning (Schölkopf et al., 2021). The overall motivation here is to learn the representation that is robust to domain-specific spurious correlations. In other words, it is postulated that "causal" features are the right kind of invariance that will enable OOD generalization. Despite the successful applications, DG remains a challenging research gap.

We differentiate our work from existing ones as follow. First, we postulate the existence of domain-invariant structure at the distributional level rather than at the data representation, which is a common assumption in DG. This is motivated by theoretical results (Mansour et al., 2009; Hoffman et al., 2018a), stating that a distribution-weighted combination of source hypotheses represents the ideal hypothesis. Furthermore, our distributional assumption, as we argued in Section 2, generalizes previous work that proposes to use domain-specific knowledge to tackle the problem of DG from a more elementary setting. For example, approaches such as Piratla et al. (2020); Monteiro et al. (2021) can be compared to our GDUs as domain-specific predictors, in the special case, where each elementary domain represents a single source domain. However, GDUs do not assume the existence of a single common classifier for all the domains, providing a combination of multiple common classifiers shared between different source domains.

Second, we incorporate the I.E.D. assumption directly into our model's architecture, as shown in Figure 2. Designing effective architectures for DG has been largely neglected (Zhou et al., 2020, Sec. 4.1). Last, we do not assume access to domain information. Although obtaining such information is difficult in practice (Niu et al., 2017), DG methods that can deal with their absence (e.g., Huang et al. (2020); Carlucci et al. (2019); Li et al. (2018c)) are yet scarce (Zhou et al., 2020, Sec. 4.2).

5 Experiments

Since ERM is one of the strongest baselines in DG (Gulrajani & Lopez-Paz, 2020; Koh et al., 2021), we, first, compare our approach compared to ERM and ensemble learning (Table 2 and Appendix A.5.1). Second, we benchmark our approach to state-of-the-art DG (e.g., CORAL, LISA, IRM, FISH, Group DRO) methods focusing on image, graph, and text data (Table 3 and Appendix A.5.5). Third, we analyse the GDUs robustness gainst DS that occurs in daily clinical practice (Table 12 and Appendix A.5.4). Finally, in Appendix A.5.2, we conduct an ablation study focusing on the representation learned during training (Appendix A.5.3). In our experiments, we distinguish two modes of training the DG layer: fine tuning (FT), where we extract features using a pre-trained model, and end-to-end training (E2E), where the FE and the DG layer are jointly trained[1].

5.1 Proof-of-concept based on digits classification

Following Feng et al. (2020) among others, we create a multi-source dataset by combining five publicly available digits image datasets, namely MNIST (Lecun et al., 1998), MNIST-M (Ganin & Lempitsky, 2015), SVHN (Netzer et al., 2011), USPS, and Synthetic Digits (SYN) (Ganin & Lempitsky, 2015). The task is to classify digits between zero and nine. Each of these datasets is considered an out-of-training target domain which is inaccessible during training, and the remaining four are the source domains. Details are given in Appendix A.5.1. Table 2 summarizes the results for the most challenging out-of-training target domain, namely MNIST-M. In Appendix A.5.1, we provide the results on the remaining target domains in Ta-

Table 2: Results Digits experiment. The mean (standard deviation) accuracy for ten runs is reported. Best results are **bold**.

		MNIST-M
ERM	*Single*	*63.00 (3.20)*
	Ensemble	*62.87 (1.50)*
FT	CS	68.55 (0.80)
	MMD	68.62 (0.70)
	Projection	68.56 (0.91)
E2E	CS	**69.25 (0.61)**
	MMD	69.04 (0.83)
	Projection	68.67 (0.98)

[1]All source code will be made available on GitHub `https://github.com` upon acceptance.

ble 7 and a discussion heuristics for choosing hyperparameters for our GDUs. Our method noticeably improves for all datasets mean accuracy and decreases the standard deviation in comparison to the ERM and ensemble baselines, making the results more stable across the ten iterations reported. We also compare our methods with related work that uses domain information and data augmentation, based on the results of Li et al. (2021) (Table 9 in Appendix A.5.1). Although data augmentation in DG is a comparatively strong approach and, at the same time, we do not use domain information; we obtain comparable results to the baselines reported by Li et al. (2021).

5.2 WILDS BENCHMARK

To challenge the I.E.D. assumption and the OOD generalization capabilities of the GDUs, we use WILDS, a curated set of real-world experiments for benchmarking DG methods (Koh et al., 2021). We consider the following eight datasets: *Camelyon17*, *FMoW*, *FMoW*, *Amazon*, *iWildCam*, and *RxRx1*, *OGB-MolPCBA*, *Civil-Comments*, and *PovertyMap*, which represent the task of real-world DG. We closely follow Koh et al. (2021) for the experiments. Details on datasets and benchmark methods are given in Appendix A.5.5. We present our benchmarking in Table 3. Our results are achieved *out-of-the-box* (i.e., default parameters) since hyperparameter optimization has a substantial impact on the generalization performance (Gulrajani & Lopez-Paz, 2020), and we aim to highlight the improvements solely attributable to our GDUs.

First, we observe the strengths and weaknesses of the benchmarks in the different data sets, all of which are lower than ERM at least once. In contrast, although GDUs show similar behavior across the datasets, performing very well for some datasets (e.g., FMoW, Poverty Map), they, however, do not fall below ERM across all GDU experiments conducted. In addition, the baselines require domain information. Our approach requires less information, yet, achieving comparable results to the benchmarks.

Table 3: Results on WILDS benchmarking tasks. Our results are achieved *out-of-the-box* (i.e., default parameters) without hyperparameter optimization. We use a grey background to highlight methods using no domain information for DG. We compute the metrics following Koh et al. (2021) and report the mean (standard deviation). Best benchmark and GDU results are **bold**.

		CAMELYON17	FMOW	AMAZON	IWILDCAM	RXRX1	OGB-MOLPCBA	CIVIL COMMENTS	POVERTY MAP
		AVG ACC	WORST-REG ACC	10% ACC	MACRO F1	AVG ACC	AVG WORST-REGION ACC	WORST-GROUP ACC	WORST-U/R R
ERM		70.3 (6.4)	31.3 (0.17)	53.8 (0.8)	31.0 (1.3)	29.9 (0.4)	**27.2 (0.3)**	56.0 (3.6)	**0.45 (0.06)**
CORAL		59.3 (7.7)	32.8 (0.66)	52.9 (0.8)	**32.8 (0.1)**	28.4 (0.3)	17.9 (0.5)	65.6 (1.3)	0.44 (0.07)
FISH		74.7 (7.1)	**34.6 (0.18)**	53.3 (0.0)	22.0 (1.8)	-	-	**75.3 (0.6)**	-
IRM		64.2 (8.1)	32.8 (2.09)	52.4 (0.8)	15.1 (4.9)	8.2 (1.1)	15.6 (0.3)	64.2 (8.1)	0.43 (0.07)
GROUP DRO		68.4 (7.3)	31.1 (1.66)	53.3 (0.0)	23.9 (2.1)	22.5 (0.3)	22.4 (0.6)	68.4 (7.3)	0.39 (0.06)
LISA		**77.1 (6.9)**	35.5 (0.81)	**54.7 (0.0)**	-	**31.9 (1.0)**	-	72.9 (1.0)	-
CGD		69.4 (7.9)	32.0 (2.26)	-	-	-	-	69.1 (1.9)	0.43 (0.04)
ARM-BN		-	24.4 (0.54)	-	23.3 (2.8)	31.2 (0.1)	-	-	-
FT	CS	68.5 (8.3)	31.8 (1.2)	**54.2 (0.8)**	**31.2 (0.8)**	**29.9 (0.3)**	**27.5 (0.2)**	56.0 (3.7)	0.46 (0.07)
	MMD	67.9 (8.0)	31.9 (1.2)	**54.2 (0.8)**	**31.2 (0.8)**	**29.9 (0.3)**	**27.5 (0.2)**	55.9 (3.7)	**0.50 (0.06)**
	PRO	66.7 (8.9)	31.8 (1.0)	**54.2 (0.8)**	30.4 (1.0)	29.8 (0.3)	27.4 (0.2)	56.5 (2.9)	0.49 (0.07)
E2E	CS	66.7 (8.9)	34.0 (1.9)	53.9 (0.7)	27.8 (2.1)	29.7 (0.4)	26.9 (0.1)	55.9 (0.8)	0.46 (0.07)
	MMD	65.7 (6.7)	**34.4 (0.7)**	**54.2 (0.8)**	27.4 (1.6)	29.6 (0.2)	27.0 (0.5)	55.8 (0.7)	**0.50 (0.06)**
	PRO	**72.3 (9.5)**	32.9 (0.8)	53.8 (0.8)	30.1 (1.2)	29.0 (0.2)	26.6 (0.3)	**56.4 (2.1)**	0.49 (0.07)

5.3 ECG EXPERIMENT

The PhysioNet/Computing in Cardiology Challenge 2020 (Perez Alday et al., 2021; Goldberger et al., 2000; Perez Alday et al., 2020) aims to identify clinical diagnoses from 12-lead ECG recordings from 6 different databases. This publicly available pooled dataset contains 43,101 recordings sampled with various sampling frequencies and lengths. Each recording is labeled as having one or more of 24 cardiac abnormalities; hence, the task is to perform a multi-label binary classification. For our experiment, we iterate over the databases, taking one at a time as the test domain while utilizing the remaining five databases for training. The performance was measured according to the original PhysioNet challenge score. This generalized intersection-over-union score assigns partial credit to misdiagnoses that result in similar treatments or outcomes. The score is then adjusted for a solution

that always selects the normal/majority class and normalized for the perfect solution. Therefore, the score can have negative values and a best possible score of 1.

Table 4 reports results for the ECG experiments (see Appendix A.5.4 for details). For this clinical time-series data, we observe an improvement in mean score and a reduction in standard deviation over the ERM and ERM ensemble baselines across all DG tasks. We attribute poorer performance for the *PTB* dataset to the fact that it contains considerably longer recordings than other datasets (except for *INCART* which, however, contains only 75 samples) and a higher sampling rate (1000Hz vs. 500Hz and 257Hz). The negative challenge score for the *PTB-XL* dataset is due to the presence of previously unobserved labels in other datasets as well as a considerably smaller amount of data for training since the *PTB-XL* dataset comprises the majority of all samples (21,837 out of 43,101).

Table 4: Results ECG experiment. All experiments were repeated five times and the mean (standard deviation) challenge metric is reported. Higher is better. Best overall results are highlighted in **bold**.

		CPSC	CPSC-EXTRA	INCART	PTB	PTB-XL	G12EC
ERM	*Single*	*0.0840 (0.0220)*	*0.2715 (0.0270)*	*0.2290 (0.0059)*	*-8.8206 (0.3908)*	*-0.3373 (0.0403)*	*0.2011 (0.0015)*
	Ensemble	*0.1699 (0.0346)*	*0.2488 (0.0079)*	*0.2456 (0.0109)*	*-8.9115 (0.1023)*	*-0.4136 (0.0780)*	*0.2079 (0.0161)*
FT	CS	0.1830 (0.0061)	0.2950 (0.0035)	0.1595 (0.0313)	-8.8802 (0.1069)	-0.1932 (0.0168)	0.1853 (0.0036)
	MMD	0.1877 (0.0077)	0.3011 (0.0035)	0.2100 (0.0413)	-8.8082 (0.1458)	-0.1567 (0.0211)	0.1919 (0.0036)
	PROJECTION	**0.1941 (0.0050)**	**0.3135 (0.0015)**	-0.1041 (0.0015)	-8.8817 (0.0478)	-0.2166 (0.0191)	0.2409 (0.0042)
E2E	CS	0.1067 (0.0170)	0.2866 (0.0146)	**0.2539 (0.0289)**	-9.2947 (0.3004)	-0.1651 (0.0494)	0.1927 (0.0080)
	MMD	0.1034 (0.0143)	0.2834 (0.0228)	0.2398 (0.0257)	-9.0600 (0.3100)	-0.1433 (0.0293)	0.1925 (0.0067)
	PROJECTION	0.1411 (0.0269)	0.2962 (0.0065)	-0.1467 (0.0513)	**-8.5904 (0.3310)**	**-0.0178 (0.0291)**	**0.2947 (0.0117)**

6 CONCLUSION AND DISCUSSIONS

We introduced the I.E.D. assumption, postulating that real-world distributions are composed of elementary distributions that remain invariant across different domains and showed that it implies an invariant structure in the solution space that enables knowledge transfer to unseen domains. Empirical results based on real-world data support the practicality of the I.E.D. assumption and that we can learn such a representation. Further, we presented a modular neural network layer consisting of Gated Domain Units (GDUs) that leverage the I.E.D. assumption. Our GDUs can substantially improve the downstream performance of learning machines in real-world DG tasks. Across our experiments, we observed that for some datasets FT is better than E2E and vice versa. In E2E training, the feature extractor (encoder) is jointly trained with GDUs. Hence, the latent representation is stochastic during training, meaning that we have variability in the representation fed into GDUs between epochs. In contrast, in FT, the feature extractor is pretrained and always produces the same embedding. Especially with large feature extractors such as ResNet-50, learning the elementary domains can be more effective when we avoid any stochasticity in the latent representation.

Limitations. A major limitation of our I.E.D. assumption is to provide theoretical evidence that this assumption holds in practice. We aim to expand the scope of the theoretical understanding of the I.E.D. assumption and the GDUs. In addition, the particular theoretical setting of Albuquerque et al. (2019) (i.e., each elementary domain represents a source domain) seems promising to extend their generalization guarantee to cases where our I.E.D. assumption holds. Second, our GDU layer induces additional computational overhead due to the regularization and model size that increases as a function of the number of elementary domains. Noteworthy, our improvement is achieved with a relatively small number of elementary domains indicating that the increased complexity is not a coercive consequence of applying the DG layer. Also, the results achieved are not a consequence of increased complexity, as the ensemble baseline shows.

Future work We expect the I.E.D. assumption and GDUs to be adapted, yielding novel applications that tackle DG. For example, we suggest dynamically increasing the number of elementary domains during learning until their distributional variance reaches a plateau as a measure of their heterogeneity. Hence, one would learn the number of elementary domains instead of fixing the number of elementary domains prior to training.

REPRODUCIBILITY STATEMENT

As for our *theoretical results*, which is Lemma 1 in Section 2.1 and Proposition 3.1 in Section 3.2, we provide the proofs in Appendix A.1 and A.2, respectively. Regarding the *empirical evaluations*, we dedicated Appendix A.5 to outline all details to reproduce our results. In addition, we provide our method for two deep learning frameworks, namely TensorFlow and PyTorch. Specifically, we implemented our method in PyTorch to integrate our method in the standardized and public evaluation pipeline of the benchmarking presented in Section 5.2. For the submission, we uploaded our anonymized repository to OpenReview. Of course, we will make our code publicly available on GitHub upon acceptance.

REFERENCES

Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H. Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *ArXiv Prepr. ArXiv191100804*, 2019.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. 2019. URL `http://arxiv.org/abs/1907.02893`. arxiv:1907.02893.

Mahsa Baktashmotlagh, Mehrtash Harandi, and Mathieu Salzmann. Distribution-matching embedding for visual domain adaptation. *J. Mach. Learn. Res.*, 17:1–30, 2016. ISSN 15337928.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. MetaReg: Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 31. Curran Associates, Inc., 2018.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 31. Curran Associates, Inc., 2018.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010. ISSN 15730565. doi: 10/c5bhdk.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from Several Related Classification Tasks to a New Unlabeled Sample. In *Advances in Neural Information Processing Systems*, volume 24, pp. 9, 2011.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):e49–e57, July 2006. ISSN 1367-4803. doi: 10/czwvqs.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain Separation Networks. In D Lee, M Sugiyama, U Luxburg, I Guyon, and R Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 29. Curran Associates, Inc., 2016.

Fabio M. Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain Generalization by Solving Jigsaw Puzzles. In *2019 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR*, pp. 2224–2233, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72813-293-8. doi: 10/gh7pq4.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov,

Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. Underspecification Presents Challenges for Credibility in Modern Machine Learning. November 2020. doi: 2011.03395v2.

Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Adv. Neural Inf. Process. Syst.*, volume 33, pp. 11996–12007. Curran Associates, Inc., 2020.

Hao-Zhe Feng, Zhaoyang You, Minghao Chen, Tianye Zhang, Minfeng Zhu, Fei Wu, Chao Wu, and Chen Wei. KD3A: Unsupervised Multi-Source Decentralized Domain Adaptation via Knowledge Distillation. *ArXiv Prepr. ArXiv201109757*, 2020. doi: 2011.09757.

Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

Yaroslav Ganin and Victor Lempitsky. Unsupervised Domain Adaptation by Backpropagation. In Francis Bach and David Blei (eds.), *Proc. 32nd Int. Conf. Mach. Learn.*, volume 37, pp. 1180–1189. PMLR, June 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *Adv. Comput. Vis. Pattern Recognit.*, 17(9783319583464):189–209, May 2015. ISSN 21916594. doi: 10/gjgc2j.

S. Gao, I. W. Tsang, and L. Chia. Sparse Representation With Kernels. *IEEE Transactions on Image Processing*, 22(2):423–434, February 2013. ISSN 1941-0042. doi: 10/f4nhfq.

Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Kernel Sparse Representation for Image Classification and Face Recognition. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios (eds.), *Comput. Vis. – ECCV 2010*, pp. 1–14, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15561-1.

Yingbo Gao, Christian Herold, Weiyue Wang, and Hermann Ney. Exploring Kernel functions in the softmax layer for contextual word classification. *arXiv*, 2019. ISSN 23318422.

Muhammad Ghifary, W. Bastiaan Kleijn, and Mengjie Zhang. Domain adaptive neural networks for object recognition. *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.*, 8862:898–904, 2014. ISSN 16113349. doi: 10/gjgc2m.

A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, June 13 2000. doi: 10.1161/01.CIR.101.23.e215. URL `http://circ.ahajournals.org/content/101/23/e215.full`. Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218;.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, 2012. ISSN 15324435.

Ishaan Gulrajani and David Lopez-Paz. In Search of Lost Domain Generalization. *ArXiv200701434 Cs Stat*, July 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, pp. 770–778, June 2016. ISBN 1063-6919. doi: 10/gdcfkn.

C. Hillermeier. Generalized homotopy approach to multiobjective optimization. *Journal of Optimization Theory and Applications*, 110(3):557–583, 2001a.

C. Hillermeier. *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. International Series of Numerical Mathematics. Springer Basel AG, 2001b.

Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 31. Curran Associates, Inc., 2018a.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In Jennifer Dy and Andreas Krause (eds.), *Proc. 35th Int. Conf. Mach. Learn.*, volume 80, pp. 1989–1998, Stockholmsmässan, Stockholm Sweden, 2018b. PMLR.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In Jennifer Dy and Andreas Krause (eds.), *Proc. 35th Int. Conf. Mach. Learn.*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2029–2037. PMLR, July 2018.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, pp. 2261–2269, 2017. doi: 10/gfhw3n.

Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging Improves Cross-Domain Generalization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Comput. Vis. – ECCV 2020*, pp. 124–140, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58536-5.

Geewook Kim, Akifumi Okuno, Kazuki Fukui, and Hidetoshi Shimodaira. Representation learning with weighted inner product for universal approximation of general similarities. In *Proc. Twenty-Eighth Int. Jt. Conf. Artif. Intell. IJCAI-19*, pp. 5031–5038. International Joint Conferences on Artificial Intelligence Organization, July 2019. doi: 10/gjgdb3.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In Doina Precup and Yee Whye Teh (eds.), *Proc. 34th Int. Conf. Mach. Learn.*, volume 70, pp. 1857–1865, International Convention Centre, Sydney, Australia, 2017. PMLR.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In Marina Meila and Tong Zhang (eds.), *Proc. 38th Int. Conf. Mach. Learn.*, volume 139, pp. 5637–5664, Proceedings of Machine Learning Research, 2021. PMLR.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256. doi: 10/d89c25.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Adv. Neural Inf. Process. Syst.*, volume 19. MIT Press, 2007.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, Broader and Artier Domain Generalization. In *2017 IEEE Int. Conf. Comput. Vis. ICCV*, pp. 5543–5551, Venice, October 2017. IEEE. ISBN 978-1-5386-1032-9. doi: 10/gfx4ss.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain Generalization with Adversarial Feature Learning. In *2018 IEEECVF Conf. Comput. Vis. Pattern Recognit.*, pp. 5400–5409, Salt Lake City, UT, June 2018b. IEEE. ISBN 978-1-5386-6420-9. doi: 10/ghv245.

Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M. Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8886–8895, October 2021.

W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool. Domain Generalization and Adaptation Using Low Rank Exemplar SVMs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5): 1114–1127, May 2018c. ISSN 1939-3539. doi: 10/gdc5f7.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep Domain Generalization via Conditional Invariant Adversarial Networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Comput. Vis. – ECCV 2018*, pp. 647–663, Cham, 2018d. Springer International Publishing. ISBN 978-3-030-01267-0.

Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proc. 36th Int. Conf. Mach. Learn.*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3915–3924. PMLR, June 2019.

Ming-Yu Liu and Oncel Tuzel. Coupled Generative Adversarial Networks. In D Lee, M Sugiyama, U Luxburg, I Guyon, and R Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 29. Curran Associates, Inc., 2016.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. *32nd Int. Conf. Mach. Learn. ICML 2015*, 1:97–105, 2015.

Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 31. Curran Associates, Inc., 2018.

Pingchuan Ma, Tao Du, and Wojciech Matusik. Efficient continuous pareto exploration in multi-task learning. In *Proceedings of the 37th International Conference on Machine Learning*. JMLR.org, 2020.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (eds.), *Adv. Neural Inf. Process. Syst.*, volume 21. Curran Associates, Inc., 2009.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple Source Adaptation and the Renyi Divergence. *Proc. 25th Conf. Uncertain. Artif. Intell. UAI 2009*, pp. 367–374, May 2012.

Joao Monteiro, Xavier Gibert, Jianqiao Feng, Vincent Dumoulin, and Dar-Shyang Lee. Domain Conditional Predictors for Domain Adaptation. In Luca Bertinetto, João F. Henriques, Samuel Albanie, Michela Paganini, and Gül Varol (eds.), *NeurIPS 2020 Workshop Pre-Regist. Mach. Learn.*, volume 148, pp. 193–220, Proceedings of Machine Learning Research, 2021. PMLR.

Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. In Sanjoy Dasgupta and David McAllester (eds.), *Proc. 30th Int. Conf. Mach. Learn.*, pp. 10–18. PMLR, February 2013.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. *Kernel Mean Embedding of Distributions: A Review and Beyond*, volume 10. May 2016.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning*, 10(1-2):1–141, 2017.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop Deep Learn. Unsupervised Feature Learn. 2011*, 2011.

L. Niu, W. Li, D. Xu, and J. Cai. Visual Recognition by Learning From Web Data via Weakly Supervised Domain Generalization. *IEEE Transactions on Neural Networks and Learning Systems*, 28(9):1985–1999, September 2017. ISSN 2162-2388. doi: 10/gbtxs5.

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.*, 37(23):3311–3325, 1997. ISSN 0042-6989. doi: 10/dn4knd.

X. Peng and K. Saenko. Synthetic to Real Adaptation with Generative Correlation Alignment Networks. In *2018 IEEE Winter Conf. Appl. Comput. Vis. WACV*, pp. 1982–1991, March 2018. doi: 10/gjgc2k.

X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment Matching for Multi-Source Domain Adaptation. In *2019 IEEECVF Int. Conf. Comput. Vis. ICCV*, pp. 1406–1415, 2019. ISBN 2380-7504. doi: 10/ghfhmt.

E. A. Perez Alday, A. Gu, A. J. Shah, C. Robichaux, A. I. Wong, C. Liu, F. Liu, A. B. Rad, A. Elola, S. Seyedi, Q. Li, A. Sharma, G. D. Clifford, and M. A. Reyna. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. *Physiol Meas*, 41(12):124003, 01 2021.

Erick Andres Perez Alday, Annie Gu, Amit Shah, Chengyu Liu, Ashish Sharma, Salman Seyedi, Ali Bahrami Rad, Matthew Reyna, and Gari D. Clifford. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020 (version 1.0.1). *PhysioNet*, 2020. doi: 10.13026/f4ab-0814.

Jonas Peters, Peter BÃŒhlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient Domain Generalization via Common-Specific Low-Rank Decomposition. In Hal Daumé III and Aarti Singh (eds.), *Proc. 37th Int. Conf. Mach. Learn.*, volume 119, pp. 7728–7738, Proceedings of Machine Learning Research, 2020. PMLR.

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Focus on the common good: Group distributional robustness follows, 2021. URL `https://arxiv.org/abs/2110.02619`.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 32. Curran Associates, Inc., 2019.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. In *International Conference on Learning Representations 2020*, pp. 19, 2020.

Bernhard Schölkopf. Causality for Machine Learning. *arXiv*, pp. 1–20, 2019. ISSN 23318422.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A Generalized Representer Theorem. In G. Goos, J. Hartmanis, J. van Leeuwen, David Helmbold, and Bob Williamson (eds.), *Computational Learning Theory*, volume 2111, pp. 416–426. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-42343-0 978-3-540-44581-4. doi: 10.1007/3-540-44581-1_27.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 525–536, Red Hook, NY, USA, 2018. Curran Associates Inc.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing Across Domains via Cross-Gradient Training. In *International Conference on Learning Representations*, pp. 12, 2018.

Yuge Shi, Jeffrey Seely, Philip H. S. Torr, N. Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient Matching for Domain Generalization. *arXiv*, 2021. doi: 2104.09937.

Alexander J. Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A Hilbert space embedding for distributions. In *Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT)*, pp. 13–31. Springer-Verlag, 2007.

Bharath Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Gert Lanckriet, and Bernhard Schölkopf. Injective Hilbert space embeddings of probability measures. In *The 21st Annual Conference on Learning Theory (COLT)*, pp. 111–122. Omnipress, 2008.

Bharath K Sriperumbudur, Gretton Arthur, Fukumizu Kenji, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.*, 11:1517–1561, 2010.

Masashi Sugiyama and Motoaki Kawanabe. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press, March 2012. ISBN 978-0-262-01709-1.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *30th AAAI Conf. Artif. Intell. AAAI 2016*, (Figure 1):2058–2065, 2016.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring Robustness to Natural Distribution Shifts in Image Classification. *arXiv:2007.00644*, September 2020.

E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial Discriminative Domain Adaptation. In *2017 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, pp. 2962–2971, July 2017. ISBN 1063-6919. doi: 10/gf5br3.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep Domain Confusion: Maximizing for Domain Invariance. January 2014.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Tomas Vicar, Jakub Hejc, Petra Novotna, Marina Ronzhina, and Oto Janousek. Ecg abnormalities recognition using convolutional network with global skip connections and custom loss function. In *2020 Computing in Cardiology*, pp. 1–4, 2020. doi: 10.22489/CinC.2020.189.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. *CoRR*, abs/2103.03097, 2021. URL `https://arxiv.org/abs/2103.03097`.

Pengtao Xie, B. Póczos, and E. Xing. Near-orthogonality regularization in kernel methods. In *UAI*, 2017.

Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1794–1801, 2009. doi: 10/d52z9q.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving Out-of-Distribution Robustness via Selective Augmentation, June 2022.

Z. Yi, H. Zhang, P. Tan, and M. Gong. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. In *2017 IEEE Int. Conf. Comput. Vis. ICCV*, pp. 2868–2876, 2017. ISBN 2380-7504. doi: 10/ggv876.

Kun Zhang, Mingming Gong, Petar Stojanov, Biwei Huang, QINGSONG LIU, and Clark Glymour. Domain adaptation as a problem of inference on graphical models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Adv. Neural Inf. Process. Syst.*, volume 33, pp. 4965–4976. Curran Associates, Inc., 2020.

Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: Learning to adapt to domain shift. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Adv. Neural Inf. Process. Syst.*, volume 34, pp. 23664–23678. Curran Associates, Inc., 2021.

Han Zhao, Shanghang Zhang, Guanhang Wu, José M F Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial Multiple Source Domain Adaptation. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett (eds.), *Adv. Neural Inf. Process. Syst.*, volume 31. Curran Associates, Inc., 2018.

S. Zhao, X. Yue, S. Zhang, B. Li, H. Zhao, B. Wu, R. Krishna, J. E. Gonzalez, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and K. Keutzer. A Review of Single-Source Deep Unsupervised Visual Domain Adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020. ISSN 2162-2388. doi: 10/ghtz3h.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep Domain-Adversarial Image Generation for Domain Generalisation. *AAAI*, 34(07):13025–13032, April 2020. doi: 10/gj4mr2.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain Generalization: A Survey. *ArXiv210302503 Cs*, March 2021a.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to Generate Novel Domains for Domain Generalization, March 2021b.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain Generalization with MixStyle. In *International Conference on Learning Representations*, pp. 15, 2021c.

J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE Int. Conf. Comput. Vis. ICCV*, pp. 2242–2251, 2017. ISBN 2380-7504. doi: 10/gfhw33.

F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1):43–76, January 2021. ISSN 1558-2256. doi: 10/ghcsqr.

A APPENDIX

A.1 PROOF OF LEMMA 1

Proof. The result holds trivially for $K = 1$. For $K \geq 2$ and by the I.E.D assumption, $\mathbb{P}^s(X,Y) = \sum_{j=1}^K \alpha_j \mathbb{P}_j(X,Y)$ for some $\boldsymbol{\alpha} \in \Delta^K$. Then, we can write the risk functional for each $f \in \mathcal{F}$ as $R(f) = \int \mathcal{L}(y, f(x))\, d\mathbb{P}^s(x,y) = \int \mathcal{L}(y, f(x))\, d(\sum_{j=1}^K \alpha_j \mathbb{P}_j(x,y)) = \sum_{j=1}^K \alpha_j \int \mathcal{L}(y, f(x))\, d\mathbb{P}_j(x,y) = \sum_{j=1}^K \alpha_j R_j(f)$ where $R_j : \mathcal{F} \to \mathbb{R}_+$ is the elementary risk functional associated with the elementary distribution $\mathbb{P}_j(X,Y)$. Hence, the Bayes predictors satisfy

$$f^* \in \arg\min_{f \in \mathcal{F}} \; R(f) = \arg\min_{f \in \mathcal{F}} \sum_{j=1}^K \alpha_j R_j(f). \tag{A.3}$$

Since the rhs of equation A.3 corresponds to the linear scalarization of a multi-objective function $(R_1, \ldots, R_K)$, its solution (i.e., a stationary point) is Pareto-optimal with respect to these objective functions (Ma et al., 2020, Definition 3.1); see, also, (Hillermeier, 2001a;b). That is, the Bayes predictors for the data distribution that satisfies the I.E.D assumption must belong to the Pareto set $\mathcal{F}_{\text{Pareto}} := \{f^* \,:\, f^* = \arg\min_{f \in \mathcal{F}} \sum_{j=1}^K \alpha_j R_j(f),\; \boldsymbol{\alpha} \in \Delta^K\} \subset \mathcal{F}$. □

A.2 PROOF OF PROPOSITION 3.1

Proof. Suppose we have a representation,

$$\mu_{\mathbb{P}} = \sum_{j=1}^M \beta_j \mu_{V_j} \qquad \langle \mu_{V_i}, \mu_{V_i} \rangle_{\mathcal{H}} = 0 \; \forall i \neq j, \tag{A.1}$$

i.e. $\{\mu_{V_1}, \ldots, \mu_{V_m}\}$ are pairwise orthogonal. We want to minimize the MMD by minimizing

$$\|\mu_{\mathbb{P}} - \sum_{j=1}^M \beta_j \mu_{V_j}\|_{\mathcal{H}}^2 = \underbrace{\langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}} \rangle_{\mathcal{H}}}_{\|\mu_{\mathbb{P}}\|_{\mathcal{H}}^2 =} - 2\langle \mu_{\mathbb{P}}, \sum_{j=1}^M \beta_j \mu_{V_j} \rangle_{\mathcal{H}} + \langle \sum_{i=1}^M \beta_i \mu_{V_i}, \sum_{j=1}^M \beta_j \mu_{V_j} \rangle_{\mathcal{H}} \tag{A.2}$$

$$= \|\mu_{\mathbb{P}}\|_{\mathcal{H}}^2 - 2\sum_{j=1}^M \beta_j \langle \mu_{\mathbb{P}}, \mu_{V_j} \rangle_{\mathcal{H}} + \sum_{i=1}^M \sum_{j=1}^M \beta_i \beta_j \underbrace{\langle \mu_{V_i}, \mu_{V_j} \rangle_{\mathcal{H}}}_{\delta_{ij} \langle \mu_{V_i}, \mu_{V_j} \rangle_{\mathcal{H}} =} \tag{A.3}$$

$$= \|\mu_{\mathbb{P}}\|_{\mathcal{H}}^2 - 2\sum_{j=1}^M \beta_j \langle \mu_{\mathbb{P}}, \mu_{V_j} \rangle_{\mathcal{H}} + \sum_{j=1}^M \beta_j^2 \|\mu_{V_j}\|_{\mathcal{H}}^2 \,. \tag{A.4}$$

By defining

$$\Phi(\beta) := \|\mu_{\mathbb{P}} - \sum_{j=1}^M \beta_j \mu_{V_j}\|_{\mathcal{H}}^2 \,, \tag{A.5}$$

we can simply find the optimal β_j by using the partial derivative

$$\frac{\partial \Phi}{\partial \beta_j} = -2\langle \mu_{\mathbb{P}}, \mu_{V_j} \rangle_{\mathcal{H}} + 2\beta_j \|\mu_{V_j}\|_{\mathcal{H}}^2 \stackrel{!}{=} 0 \tag{A.4}$$

$$\Leftrightarrow \beta_j \|\mu_{V_j}\|_{\mathcal{H}}^2 = \langle \mu_{\mathbb{P}}, \mu_{V_j} \rangle_{\mathcal{H}} \tag{A.5}$$

$$\Leftrightarrow \beta_j^* = \frac{\langle \mu_{\mathbb{P}}, \mu_{V_j} \rangle_{\mathcal{H}}}{\|\mu_{V_j}\|_{\mathcal{H}}^2} \,. \tag{A.6}$$

Please note that the function Φ is convex. □

A.3 DETAILED VIEW OF THE REGULARIZATION TERM Ω_D^{OLS}

First, consider the following single term $\|\phi(\tilde{x}_i) - \sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2$ that can be expressed as

$$\|\phi(\tilde{x}_i) - \sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2 = \underbrace{\|\phi(\tilde{x}_i)\|_{\mathcal{H}}^2}_{(1)} \underbrace{-2\,\langle \phi(\tilde{x}_i), \sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\rangle_{\mathcal{H}}}_{(2)} + \underbrace{\|\sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2}_{(3)} . \tag{A.7}$$

AD (1):

We begin with Term (1) and write $\|\phi(\tilde{x}_i)\|_{\mathcal{H}}^2$ as $\|\phi(\tilde{x}_i)\|_{\mathcal{H}}^2 = \langle \phi(\tilde{x}_i), \phi(\tilde{x}_i)\rangle_{\mathcal{H}} = k(\tilde{x}_i, \tilde{x}_i)$. We could evaluate this term using the kernel function k for each data point in the batch b. However, since this term does not depend on the the elementary domains $\{V_1, \ldots, V_M\}$, it is unnecessary to compute this value to minimize the penalty. Thus, we obtain a similar result by minimizing the penalty without considering $\|\phi(\tilde{x}_i)\|_{\mathcal{H}}^2$ in the regularization.

AD (2):

Term (2) can be expressed as

$$\langle \phi(\tilde{x}_i), \sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\rangle_{\mathcal{H}} = \sum_{j=1}^{M} \beta_{ij}\langle \phi(\tilde{x}_i), \mu_{V_j}\rangle_{\mathcal{H}} \tag{A.8}$$

Implementation-wise, the evaluation of this term requires the calculation of the inner product $\langle \phi(\tilde{x}_i), \mu_{V_j}\rangle_{\mathcal{H}}$. Since our CS and projection-based methods involve this inner product to determine the coefficients β_{ij}, we pre-compute the inner product $\langle \phi(\tilde{x}_i), \mu_{V_j}\rangle_{\mathcal{H}}$ once for a mini-batch and store these information during training to avoid multiple calculations of the same term.

Moreover, the projection-based method does not apply softmax and has a linear form. Therefore, the term (2) can be simplified even further:

$$\langle \phi(\tilde{x}_i), \sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\rangle_{\mathcal{H}} = \sum_{j=1}^{M} \beta_{ij}\langle \phi(\tilde{x}_i), \mu_{V_j}\rangle_{\mathcal{H}} \tag{A.9}$$

$$= \sum_{j=1}^{M} \frac{\langle \phi(\tilde{x}_i), \mu_{V_j}\rangle_{\mathcal{H}}}{\|\mu_{V_j}\|_{\mathcal{H}}^2}\langle \phi(\tilde{x}_i), \mu_{V_j}\rangle_{\mathcal{H}} \tag{A.10}$$

$$= \sum_{j=1}^{M} \frac{\langle \phi(\tilde{x}_i), \mu_{V_j}\rangle_{\mathcal{H}}^2}{\|\mu_{V_j}\|_{\mathcal{H}}^2}. \tag{A.11}$$

AD (3):

Last, we express the term (3) as follows

$$\|\sum_{j=1}^{M} \beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2 = \sum_{j=1}^{M}\sum_{k=1}^{M} \beta_{ij}\beta_{ik}\langle \mu_{V_j}, \mu_{V_k}\rangle_{\mathcal{H}}, \tag{A.12}$$

and calculate the inner product of the domains $\langle \mu_{V_j}, \mu_{V_k}\rangle_{\mathcal{H}}$ by

$$\langle \mu_{V_j}, \mu_{V_k}\rangle_{\mathcal{H}} = \frac{1}{N^2}\sum_{l=1}^{N}\sum_{m=1}^{N} \langle \phi(v_j^l), \phi(v_k^m)\rangle_{\mathcal{H}} \tag{A.13}$$

$$= \frac{1}{N^2}\sum_{l=1}^{N}\sum_{m=1}^{N} k\big(v_j^l, v_k^m\big) =: K_{jk}, \tag{A.14}$$

where N represents the number of vectors per domain basis. Note that this term does not depend on the input data x_i and, hence, matrix K_{jk} can be calculated once at the beginning of the optimization step and stored to be re-used for all the data point of a batch.

Combining Equation A.12 and Equation A.14 yields

$$\|\sum_{j=1}^{M}\beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2 = \sum_{j=1}^{M}\sum_{k=1}^{M}\beta_{ij}\beta_{ik}\langle\mu_{V_j},\mu_{V_k}\rangle_{\mathcal{H}} \tag{A.15}$$

$$= \frac{1}{N^2}\sum_{j=1}^{M}\sum_{k=1}^{M}\beta_{ij}\beta_{ik}\sum_{l=1}^{N}\sum_{m=1}^{N}k\big(v_j^l, v_k^m\big) \tag{A.16}$$

$$= \sum_{j=1}^{M}\sum_{k=1}^{M}\beta_{ij}\beta_{ik}K_{jk} \tag{A.17}$$

$$= \beta_i^T K_{jk}\beta_i\,. \tag{A.18}$$

As a final step, we use the results for Term (1), (2), and (3) to obtain the desired regularization term

$$\Omega_D^{OLS} = \frac{1}{b}\sum_{i=1}^{b}\left(\|\phi(\tilde{x}_i) - \sum_{j=1}^{M}\beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2\right) \tag{A.19}$$

$$= \frac{1}{b}\sum_{i=1}^{b}\left(\|\phi(\tilde{x}_i)\|_{\mathcal{H}}^2 - 2\langle\phi(\tilde{x}_i), \sum_{j=1}^{M}\beta_{ij}\mu_{V_j}\rangle_{\mathcal{H}} + \|\sum_{j=1}^{M}\beta_{ij}\mu_{V_j}\|_{\mathcal{H}}^2\right). \tag{A.20}$$

As mentioned above, $\|\phi(\tilde{x}_i)\|_{\mathcal{H}}^2$ is independent from the elementary domains, and, thus a constant in the regularization. Hence, we can exclude this term, which avoids additional computational effort.

A.4 Visualization of DG Layer

Figure 3 depicts the layout of our DG layer.

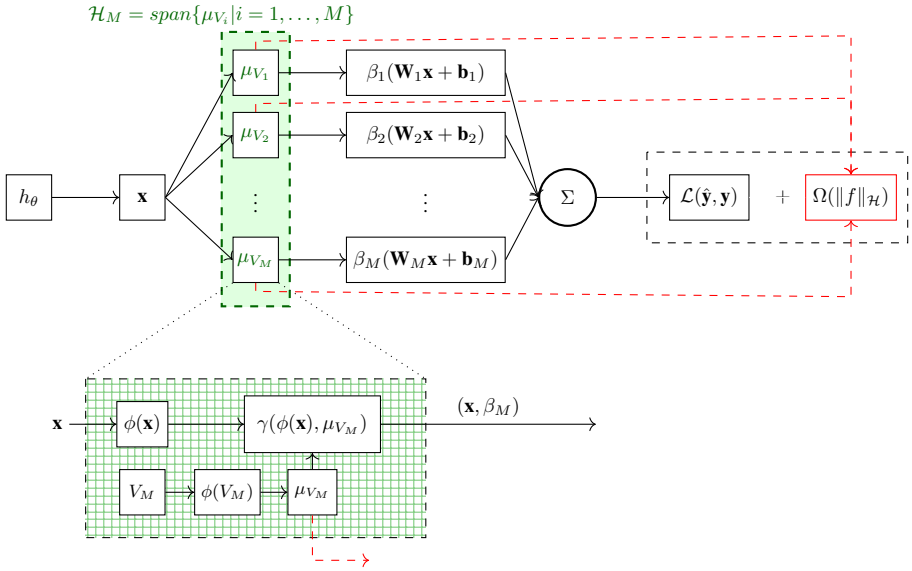

Figure 3: Domain generalization (DG) layer.

A.5 EXPERIMENTS

In this section, we provide a detailed description of the DG experiment presented in Section 5. Our Digits and ECG experiments are implemented using TensorFlow 2.4.1 and TensorFlow Probability 0.12.1. For the WILDS benchmarking we use our PyTorch (version 1.11.0). The code for both deep learning frameworks will be made publicly available on GitHub at `https://github.com` upon acceptance. Overall, our experiments aim to show the validity of the invariant elementary distribution (I.E.D.) assumption and the Gated Domain Units (GDUs).

For the DG layer, we considered two modes of model training: fine tuning (FT) and end-to-end training (E2E). In FT scenario, we first pre-train the FE in the ERM single fashion. Then, we extract features using the pre-trained model and pass them to the DG layer for training the latter. For the E2E training, however, the whole model including the FE and DG layer is trained jointly from the very beginning.

A.5.1 DIGITS EXPERIMENT

Our experiment setup is closely related to Peng et al. (2019); Feng et al. (2020); Zhang et al. (2020); Zhao et al. (2018). Each dataset, except USPS, is split into training and test sets of 25,000 and 9,000 images, respectively. For USPS, we take the whole dataset for the experiment since it contains only 9,298 images[2]. Our experimental setup regarding datasets, data loader, and FE are based on existing work (Feng et al., 2020; Peng et al., 2019). The structure of the FE is summarized in Table 5 and the subsequent learning machine is a dense layer.

In the Empirical Risk Minimization (ERM) single experiment, we add a dense layer with 10 outputs (activation=*tanh*) as a classifier to the FE. In the Empirical Risk Minimization (ERM) ensemble experiment, we add M classification heads (a dense layers with 10 outputs and *tanh* activation each) to the FE and average their output for the final prediction. This sets a baseline for our DG layer to show performance gain against the ERM model with the same number of learning machines.

For training, we resorted to the Adam optimizer with a learning rate of 0.001. We used early stopping and selected the best model weights according to the validation accuracy. For the validation data, we used the combined test splits only of the respective source datasets. The batch size was set to 512. Although the DG layer requires more computation resources than the ERM models, all digits experiments were conducted on a single GPU (NVIDIA GeForce RTX 3090).

Heuristics for main parameter of DG layer From a practical perspective, our layer requires choosing two main hyper-parameters: the number of elementary domains M and since we use the characteristics Gaussian kernel the corresponding parameter σ. The parameter M determines the size of the ensemble of learning machines and, thus, for deep learning models, their overall network size. As a heuristic to choose M, we suggest to cluster the output of a pre-trained FE. In the following, we provide an example. We pre-trained the FE for the test domain *MNIST-M* and pass the source data through this FE, which we cluster with the k-means algorithm. Subsequently, we analyse three different metrics (Calinski Harabasz score, Davies Bouldinn score, and Silhouette score) to select the optimal number of clusters as the basis to choose M. All scores yielded an accordance between four to five clusters. Therefore, we set M to five and observed in Table 2 in Section 5 strong results in the generalizing to the unseen test domain *MNIST-M*.

As for the parameter σ, we resort to the median heuristic proposed in (Muandet et al., 2016) that is $\sigma^2 = \text{median}\{ \parallel \tilde{x}_i - \tilde{x}_j \parallel^2 : i, j = 1, \ldots, n\}$. While both heuristics require a pre-trained FE, cross-validation can act as a reasonable alternative. The hyper-parameters relevant for the DG layer are summarized in Table 6. In the FT setting, we applied the median heuristics presented above to estimate σ of the Gaussian kernel function, where the estimator is denoted as $\hat{\sigma}$. Since median heuristic is not applicable for the E2E scenario, σ was fixed to 7.5 for E2E.

Note that our approach to choose the relevant parameters was kept very general to show the feasibility of the I.E.D. assumption and the generalization ability of GDUs and, most importantly, to provide easy-to-reproduce results. During training, additional epoch metrics can be subscribed using our custom *DG layer callback*, which may help to choose the model parameters. Furthermore, we

[2]We used the digits data from `https://github.com/FengHZ/KD3A` [last accessed on 2022-05-17, available under MIT License.] published in Feng et al. (2020).

Table 5: Feature Extractor used for the Digits Experiment

FEATURE EXTRACTOR	
LAYER TYPE	OUTPUT SHAPE
2D-CONVOLUTIONAL LAYER	(32, 32, 64)
BATCH NORMALIZATION	(32, 32, 64)
MAXPOOLING 2D	(16, 16, 64)
2D-CONVOLUTIONAL LAYER	(16, 16, 64)
BATCH NORMALIZATION	(16, 16, 64)
MAXPOOLING 2D	(8, 8, 64)
2D-CONVOLUTIONAL LAYER	(8, 8, 128)
BATCH NORMALIZATION	(8, 8, 128)
MAXPOOLING 2D	(4, 4, 128)
FLATTEN	(2048)
DENSE LAYER	(3072)
BATCH NORMALIZATION	(3072)
DROPOUT	(3072)
BATCH NORMALIZATION	(2048)
DENSE LAYER	(2048)

observed that the elementary domains become naturally orthogonal during the experiments, and thus, we set λ_{ORTH} relatively small. Since the orthogonal regularization puts additional computational burden, one could omit this term completely to speed up training.

Table 6: Parameters for DG Layer in Digits and Digit-DG Experiments for the Fine Tuning (FT) and End-to-end training (E2E) Settings. In case of Projection, we chose the spectral restricted isometry property (SRIP) as the orthogonal regularization $\Omega_D^{\perp}$.

EXPERIMENT		M	N	λ_{L_1}	λ_{OLS}	λ_{ORTH}	σ	κ
FT	CS	5	10	$1e^{-3}$	$1e^{-3}$	-	$\hat{\sigma}$	2
	MMD	5	10	$1e^{-3}$	$1e^{-3}$	-	$\hat{\sigma}$	2
	PROJECTION	5	10	$1e^{-3}$	$1e^{-3}$	$1e^{-8}$	$\hat{\sigma}$	-
E2E	CS	5	10	$1e^{-3}$	$1e^{-3}$	-	7.5	2
	MMD	5	10	$1e^{-3}$	$1e^{-3}$	-	7.5	2
	PROJECTION	5	10	$1e^{-3}$	$1e^{-3}$	$1e^{-8}$	7.5	-

Table 7: Results Digits experiment. All experiments were repeated ten times and the mean (standard deviation) accuracy is reported. Best results according to the mean accuracy are highlighted in **bold**.

		MNIST	**MNIST-M**	**SVHN**	**USPS**	**SYN**
ERM	*Single*	*97.98 (0.34)*	*63.00 (3.20)*	*70.18 (2.74)*	*93.70 (1.74)*	*83.62 (1.47)*
	Ensemble	*98.21 (0.39)*	*62.87 (1.50)*	*72.01 (3.59)*	*95.16 (0.89)*	*83.80 (1.22)*
FT	CS	98.53 (0.16)	68.55 (0.80)	78.90 (1.41)	95.83 (0.50)	88.39 (0.82)
	MMD	98.60 (0.08)	68.62 (0.70)	79.20 (2.01)	**96.24 (0.71)**	88.27 (0.41)
	PROJECTION	98.57 (0.17)	68.56 (0.91)	79.34 (0.72)	96.24 (0.71)	88.58 (0.53)
E2E	CS	98.62 (0.19)	**69.25 (0.61)**	**79.42 (1.27)**	96.17 (0.52)	87.92 (0.84)
	MMD	98.58 (0.16)	69.04 (0.83)	79.20 (0.90)	96.00 (0.44)	88.18 (0.86)
	PROJECTION	**98.67 (0.12)**	68.67 (0.98)	78.56 (1.68)	96.24 (0.77)	**88.77 (0.48)**

Digit-DG Benchmark In previous research, the aforementioned digits data is not only used for domain adaptation (DA), but also for domain generalization (DG) methods. For the latter, Zhou et al. (2021b) and Li et al. (2021) introduced Digit-DG dataset and the evaluation protocol to benchmark seven DG methods and ERM [3]. Unlike the Digits experiment described above, Digit-DG dataset from

[3]Results were reported by Zhou et al. (2021b) and Li et al. (2021). Of note, both authors did not report the standard deviation on their results.

Zhou et al. (2021b) and Li et al. (2021) consists of only four datasets (without USPS) and a different FE summarized in Table 8. Therefore, we follow their instructions to conduct a fair comparison and ensure reproducibility. For the hyper-parameters, however, we kept the same values that we used for the Digits experiment, see Table 6.

Table 8: Feature Extractor used for the Digit-DG Benchmark Experiment

FEATURE EXTRACTOR	
LAYER TYPE	OUTPUT SHAPE
2D-CONVOLUTIONAL LAYER	(32, 32, 64)
MAXPOOLING 2D	(16, 16, 64)
2D-CONVOLUTIONAL LAYER	(16, 16, 64)
MAXPOOLING 2D	(8, 8, 64)
2D-CONVOLUTIONAL LAYER	(8, 8, 64)
MAXPOOLING 2D	(4, 4, 128)
2D-CONVOLUTIONAL LAYER	(8, 8, 64)
MAXPOOLING 2D	(4, 4, 128)
2D-CONVOLUTIONAL LAYER	(4, 4, 64)
MAXPOOLING 2D	(2, 2, 64)
FLATTEN	(256)

As a first method, we consider the CCSA (Classification and Contrastive Semantic Alignment) method, which learns a domain-invariant representation by utilizing the CCSA loss (Motiian et al., 2017). Second, MMD-AAE (Maximum Mean Discrepancy-based Adverserial Autoencoders) extends adverserial autoencoders by a maximum mean discrepancy regularization to learn a domain-invariant feature representation (Li et al., 2018b). CrossGrad (Cross-Gradient) augments data by perturbating the input space using the cross-gradients of a label and domain predictor (Shankar et al., 2018). Another augmentation-based DG method is L2A-OT (Learning to Augment by Optimal Transport) (Zhou et al., 2021b). Specifically, a data generator trained to maximize the optimal transport distance between source and pseudo domains, is used to augment the source data. All aforementioned methods rely on the availability of domain information such as domain labels. To benchmark our layer to a method for DG without domain information, we resort to the JiGen (Jigsaw puzzle based Generalization) method (Carlucci et al., 2019). JiGen introduces an auxiliary loss for solving jigsaw task during training. Further, we use the adaptive and non-adaptive stochastic feature augmentation (SFA-S and SFA-A, respectively) method proposed by Li et al. (2021). In principle, both method augment the latent feature embedding of a FE using random noise.

Our results are summarized in Table 9. As noted by Li et al. (2021), it is challenging to outperform augmentation-based DG methods. In addition, SFA-A and SF-S are computationally light (i.e., only adding random noise to the feature embedding) and do not require domain information (Li et al., 2021). Nevertheless, our layer achieves competitive results even against the strongest baselines in all DG tasks without requiring domain information.

A.5.2 ABLATION STUDY

Brief Summary We chose the digits dataset (including USPS and same setting described in the first paragraph of Appendix A.5) to analyze each component of our DG layer. We (A) vary M, N on Figure 7, and the strength of the regularization terms on Figure 4, Figure 5, and Figure 6 to assess the sensitivity of the DG layer to the choice of hyper-parameters (Appendix A.5.2), (B) visualize the output of the FE (Figure 9). Our ablation study in (A) reveals stable results across different sets of hyper-parameters. While the layer is not sensitive to the choice of regularization strength, we recommend not to omit the regularization completely, although the computational expenses decrease without the orthogonal regularization. As an illustration in (B), we project the output of the FE trained with a dense layer (ERM) and with the DG layer by t-SNE (t-distributed stochastic neighbor embedding). The GDU-trained FE yields more concentrated and bounded clusters in comparison to the one trained by ERM. Hence, we observe a positive effect on the representation learned by the FE.

Detailed Analysis We chose the Digits dataset to conduct an ablation study, which is organized as follows: (1) ablation of the regularization terms presented in Section 3, (2) effect of the orthogonal regularization for projection-based generalization, and (3) affect on the FE's output.

Table 9: Results of the Digits-DG experiment. All experiments were repeated ten times. Methods are classified into augmentation-based (A) and non-augmentation-based (B) as well as DG with (✓) and without (✗) domain information according to Li et al. (2021). Best results according to the mean accuracy are highlighted in **bold**.

		MNIST	MNIST-M	SVHN	SYN	CATEGORIE	DOMAIN INFORMATION
		Results reported by (Li et al., 2021)					
ERM		*95.8*	*58.8*	*61.7*	*78.6*	B	✗
CCSA (MOTIIAN ET AL., 2017)		*95.2*	*58.2*	*65.5*	*79.1*	B	✓
MMD-AAE (LI ET AL., 2018B)		*96.5*	*58.4*	*65.0*	*78.4*	B	✓
CrossGrad(SHANKAR ET AL., 2018)		*96.7*	*61.1*	*65.3*	*80.2*	A	✓
L2A-OT (ZHOU ET AL., 2021B)		*96.7*	*63.9*	*68.6*	*83.2*	A	✓
SFA-S (LI ET AL., 2021)		*96.7*	*66.3*	*68.8*	*85.1*	A	✗
SFA-A (LI ET AL., 2021)		*96.5*	*66.5*	*70.3*	*85.0*	A	✗
JiGen (CARLUCCI ET AL., 2019)		*96.5*	*61.4*	*63.7*	*74.0*	B	✗
		Gated Domain Units (GDUs)					
FT	CS	97.5	68.9	74.0	85.5	B	✗
	MMD	97.6	69.0	74.2	84.3	B	✗
	PROJECTION	97.7	69.1	74.1	86.1	B	✗
E2E	CS	97.6	69.4	**75.9**	**86.5**	B	✗
	MMD	97.6	**69.5**	75.6	**86.5**	B	✗
	PROJECTION	**97.8**	66.7	73.4	84.0	B	✗

As a reminder, we introduced the regularization to be dependent on the form of generalization (i.e., domain similarity measures or projection-based generalization in Section 3). For the domain similarity measure case, the regularization is

$$\Omega_D(\|g\|_{\mathcal{H}}) = \lambda_{OLS}\Omega_D^{OLS}(\|g\|_{\mathcal{H}}) + \lambda_{L_1}\Omega_D^{L_1}(\|\gamma\|), \tag{A.21}$$

where $\lambda_{OLS}, \lambda_{L_1} \geq 0$. In the case of projection, the regularization is given by

$$\Omega_D(\|g\|_{\mathcal{H}}) = \lambda_{OLS}\Omega_D^{OLS}(\|g\|_{\mathcal{H}}) + \lambda_{ORTH}\Omega_D^{\perp}(\|g\|_{\mathcal{H}}) \tag{A.22}$$

with $\lambda_{OLS}, \lambda_{ORTH} \geq 0$. Although one can additionally choose the sparse regularization in projection-based generalization, we set the focus in the ablation study on the two main regularization terms that are the OLS and orthogonal regularization. For (1) we vary in Equation A.21 and Equation A.22 the corresponding weights λ_1 and λ_2 in the interval of $[0; 0.1]$ and display the mean classification accuracy for the most challenging classification task of MNSIT-M in the form of a heatmap. In Figures 4-6, we see that the classification accuracy remains on an overall similar level which indicates that the DG layer is not very sensitive to the hyper-parameter change for MNIST-M as the test domain. Nevertheless, we observe that ablating the regularization terms by setting the corresponding weights to zero decreases the classification results and the peaks in performance occur when the regularization is included during training of the DG layer.

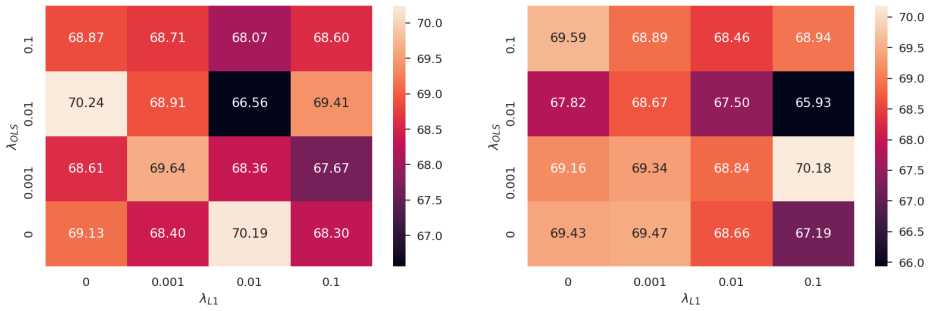

Figure 4: Classification results for varying λ_{L_1} and λ_{OLS} in the interval of $[0; 0.1]$ for FT (left) and E2E (right) CS on MNIST-M.

Applying the DG layer comes with additional overhead, especially the regularization that ensures the orthogonality of the elementary domain bases. This additional effort raises a question whether

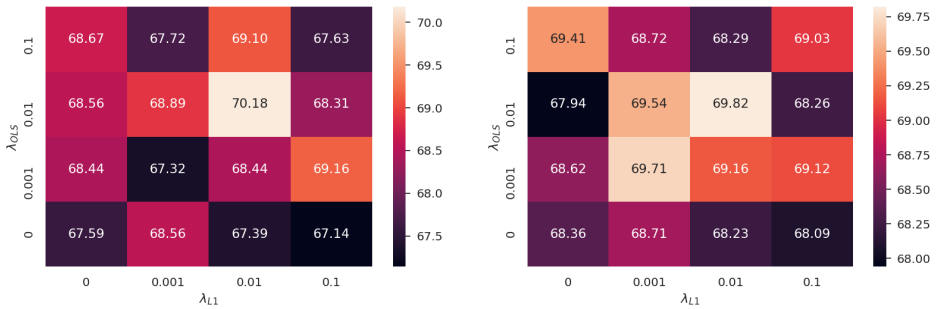

Figure 5: Classification results for varying λ_{L_1} and λ_{OLS} in the interval of $[0; 0.1]$ for FT (left) and E2E (right) MMD on MNIST-M.

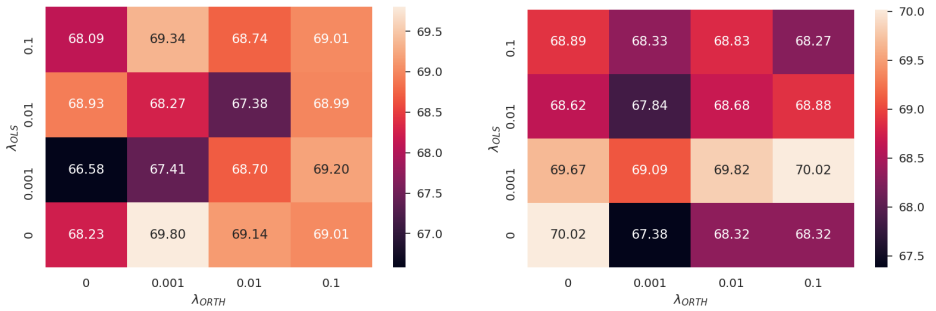

Figure 6: Classification results for varying λ_{ORTH} and λ_{OLS} in the interval of $[0; 0.1]$ for FT (left) and E2E (right) Projection on MNIST-M.

ensuring the theoretical assumptions outweigh the much higher computational effort. Thus, in a second step, we analyze how the orthogonal regularization affects the orthogonality of the elementary domain bases (i.e., spectral restricted isometry property (SRIP) value) and the loss function (i.e., categorical cross-entropy).

In Figure 8, we depict the mean and standard deviation of the SRIP value and loss over five runs for 40 epochs. The SRIP value can be tracked during training with the DG layer's callback functionalities. First, we observe that the elementary domains are almost orthogonal when initialized. Training the layer leads in the first epochs to a decrease in orthogonality. This initial decrease happens because cross-entropy has a stronger influence on the optimization than regularization in the first epochs. After five epochs, the cross-entropy decrease to a threshold when the regularization becomes more effective and the orthogonality of the elementary domain bases increases again. In Figure 8, we also observe that ablating the orthogonal regularization, while leading to better orthogonality of the domains, does not significantly affect the overall cross-entropy during training.

Finally, we project the output of the FE trained with a dense layer (ERM) and with the DG layer by t-SNE (t-distributed stochastic neighbor embedding) in Figure 9. The GDU-trained FE yields more concentrated and bounded clusters in comparison to the one trained by ERM. Hence, we observe a positive effect on the representation learned by the FE.

A.5.3 Interpretation of the Elementary Domains

We analyze the learned elementary domains in the digits experiment based on two visualizations, and choose the maximum mean discrepancy (MMD) as the similarity measure and *MNIST-M* as the test domain. The first visualization depicts the MMD between the datasets (i.e., *MNIST*, *MNIST-M*, *SVHN*, *USPS*, and *Synthetic Digits (SYN)*) and the learned elementary domains (i.e., $V_1 - V_5$) as a heatmap (see Figure 10 (left)). The heatmap indicates that the source and test domains are close to one another in terms of the MMD. Hence, we expect that their closeness reflects in the learning of the elementary domains. In other words, we expect that each elementary domains contributes similarly to

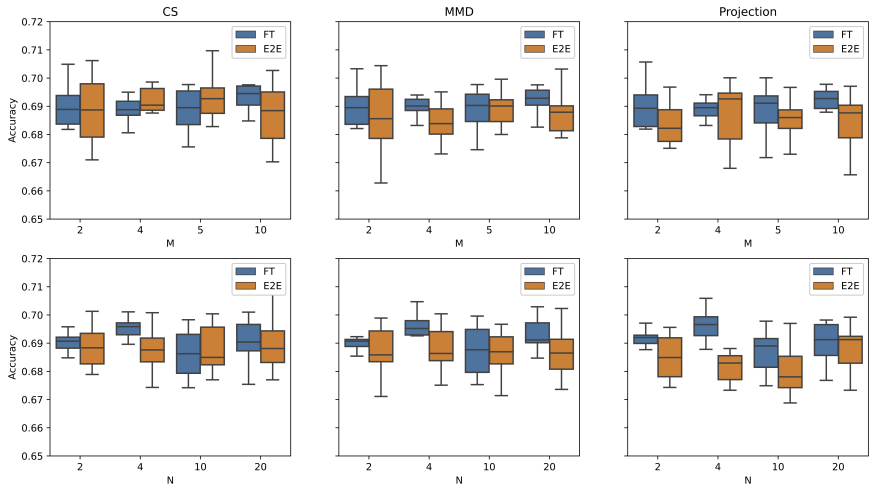

Figure 7: Mean and standard deviation of classification accuracy over 10 runs for varying number of elementary domains (M, upper panel) and varying number of vector for each domain basis (N, lower panel) for MNIST-M dataset.

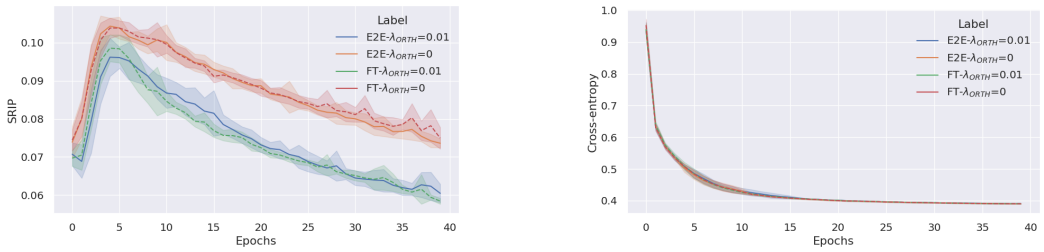

Figure 8: Effect of omitting the orthogonal regularization term $\Omega_D^{\perp}$. Spectral restricted isometry property (SRIP) (left) and categorical cross-entropy (right) with and without orthogonal regularization and their evolution during training for MNSIT-M dataset. The mean and standard deviation presented for End-to-end (E2E) and Fine-tuning (FT) training scenarios are calculated over 10 runs.

the source and test domains (i.e., the coefficients β are similar for each of these domains). In Section 3.1, we derive the coefficients by applying a kernel softmax function to the negative MMD distances. Since the MMD distances between the source / test domains and the elementary domains are similar, the coefficients will be similar too. We conclude that the learned elementary domains represent the same distributional characteristics that existed among the source and test domains.

In the second visualisation, we show the t-SNE (t-distributed stochastic neighbor embedding) of the feature extractor output for each source and test domain alongside the elementary domains in Figure 10 (right). First, we observe that the learned elementary domain bases form distinctive clusters. We see these clusters as a validation of our hypothesis that each GDU learns to mimic samples generated from a corresponding elementary distribution as pointed out in Section 2.2. However, we can not answer whether and where these elementary distributions occur in the real world. Moreover, these elementary distributions yet lack interpretability.

In summary, the MMD heatmap and t-SNE embeddings of the learned elementary and source domains on Figure 10 indicate that the GDUs learn to represent distributional structures in the dataset.

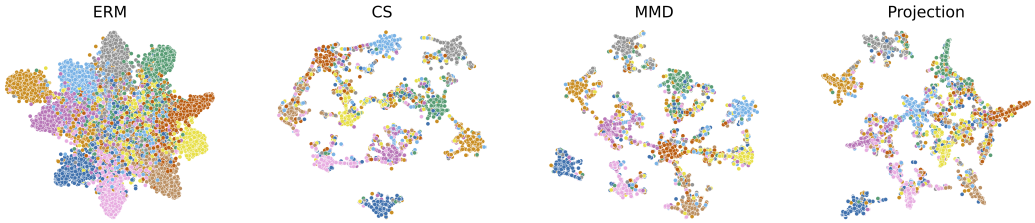

Figure 9: Visualization of t-SNE Embedding on unseen Synthetic Digits Dataset. Colors encode true label.

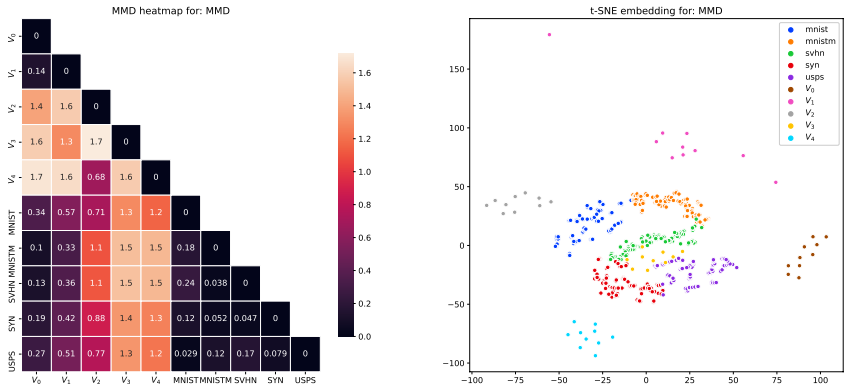

Figure 10: MMD heatmap (left) and t-SNE embedding (right) for the test domain MNIST-M.

A.5.4 ECG Experiment

We adopted the task of multi-label binary classification of 12-lead electrocardiogram (ECG) signals combined from 6 different sources introduced in the *PhysioNet/Computing in Cardiology Challenge 2020*[4] (Perez Alday et al., 2021; Goldberger et al., 2000; Perez Alday et al., 2020). Each ECG recordings is annotated with 24 binary labels indicating whether or not a certain cardiac abnormality is present. The data is aggregated from 6 different databases and contains 43,101 recordings sampled with various sampling frequencies, number of subjects, and lengths. Table 10 summarizes most important details about the data sources for this experiment.

According to the original challenge score, we measure the performance in terms of the generalized Intersection-over-Union (IoU) score where partial credit is assigned to misdiagnoses that result in similar treatments or outcomes. The score is defined as

$$score := \frac{y^T \cdot W \cdot \hat{y}}{y \cup \hat{y}}, \tag{A.23}$$

where $y, \hat{y} \in \{0,1\}^{24}$ represent actual labels and predicted labels and W stands for the partial credit-assignment matrix provided as a part of the challenge description. Note that in case of identity matrix W the score is exactly the Intersection-over-Union (IoU) score. The score is then adjusted for a solution $y_{majority}$, which always predicts the normal/majority class, and is moreover normalized for the perfect solution y. Therefore, the final score can have negative values and the best possible score of 1 and is formalized as

$$adjusted\ \ score := \frac{score(y, \hat{y}) - score(y, y_{majority})}{score(y, y) - score(y, y_{majority})}. \tag{A.24}$$

As a pre-processing step, we down-sampled all the signals to 125 Hz and applied Z-score, random amplification and random stretching according to Vicar et al. (2020). For that we partially adopted

[4] `https://physionetchallenges.org/2020/` [last accessed on 2021-03-10, available under Creative Commons Attribution 4.0 International Public License].

Table 10: ECG Data Sources Details.

DATASET	NUMBER OF SAMPLES	ECG LENGTH [SEC]	FREQUENCY	COUNTRY
CPSC	6,877	6 TO 60	500 HZ	CHINA
CPSC-EXTRA	3,453	6 TO 60	500 HZ	CHINA
INCART	75	1,800	257 HZ	RUSSIA
PTB	549	UP TO 120	1,000 HZ	GERMANY
PTB-XL	21,837	10	500 HZ	GERMANY
G12EC	10,344	10	500 HZ	USA

the code provided by the authors[5]. Additionally, we cropped each signal to its first 15,000 points if the signal was too long (mostly applied to *INCART* database). Each dataset was randomly split into train and validation parts with 3:1 ratio. During each experiment, we used the train splits of 5 databases for training and utilized the validation splits of the training databases for early stopping. The hold-out 6-th database was used for inference and testing only.

Table 11 describes the architecture of FE used for the task. Since the provided ECG recordings have different lengths, we used TensorFlow padded batching, which is padding all the recordings in a batch to the length of the longest sequence in the batch. Therefore, input from different batches can have different lengths so the spatial dimensions of the 1D-Convolutional layers are not predefined and are presented as *.

Table 11: Feature Extraction Architecture used for the ECG Experiment is an adapted Version of LeNet Architecture for 1D input Signals. Note that ECG recordings have variable lengths, therefore, the spatial dimension is not defined and denoted as *.

FEATURE EXTRACTOR	
LAYER TYPE	OUTPUT SHAPE
1D-CONVOLUTIONAL LAYER (KERNEL SIZE=*3*, ACTIVATION=*relu*)	(*, 32)
BATCH NORMALIZATION	(*, 32)
MAXPOOLING 1D (POOL SIZE=*2*, STRIDES=*2*)	(*, 32)
1D-CONVOLUTIONAL LAYER (KERNEL SIZE=*2*, ACTIVATION=*relu*)	(*, 64)
BATCH NORMALIZATION	(*, 64)
GLOBAL AVERAGE POOLING 1D	(64)
DENSE LAYER (ACTIVATION=*relu*)	(100)
DENSE LAYER (ACTIVATION=*relu*)	(100)

We used the Adam optimizer to optimize weighted binary cross-entropy loss defined as $-(w_{pos} \cdot y \cdot \log \hat{y}) + (1 - y) \cdot \log (1 - \hat{y})$. Positive weights w_{pos} are defined per class based on the training split data inversely proportional to the frequency of positive labels for each class. A learning rate was initially set to 0.001 and during the training reduced by the factor of 0.2 if the training loss was not improving for 10 epochs. We also applied early stopping and restored model weights to the best model according to the validation accuracy after the training end. Since each input samples for this experiment have a larger size than the previous one, we decreased the batch size to 64. Each ECG experiment was performed on a single GPU (Nvidia GTX 1080 Ti). The parameters relevant for the DG layer are summarized in Table 12. We have to emphasize that we did not perform extensive hyper-parameter tuning since our goal was to show the feasibility of the I.E.D. assumption and GDUs while keeping the experiments reproducible.

A.5.5 WILDS BENCHMARKING EXPERIMENTS

For comparison of our approach and benchmarking, we followed the standard procedure of WILDS experiments, described in Koh et al. (2021). As a technical note, all WILDS experiments have been implemented in Pytorch (version >= 1.7.0) based on the specifications made in Koh et al. (2021) and their code published on `https://github.com/p-lambda/wilds` [last accessed on 2022-05-17, available under MIT License]. The results for the benchmarks were retrieved from the official leaderboard `https://wilds.stanford.edu/leaderboard/` [last accessed on 2022-09-26].

[5] `https://github.com/tomasvicar/BUTTeam` [last accessed on 2022-05-17, available under BSD 2-Clause License].

Table 12: Parameters for DG Layer in ECG experiments for the Fine Tuning (FT) and End-to-end training (E2E) Settings.

Experiment		M	N	λ_{L_1}	λ_{OLS}	λ_{ORTH}	σ	κ
FT	CS	10	10	$1e^{-3}$	$1e^{-3}$	-	5.5	2
	MMD	10	10	$1e^{-3}$	$1e^{-3}$	-	5.5	2
	Projection	10	10	$1e^{-3}$	$1e^{-3}$	$1e^{-6}$	5.5	-
E2E	CS	10	10	$1e^{-3}$	$1e^{-3}$	-	5.5	2
	MMD	10	10	$1e^{-3}$	$1e^{-3}$	-	5.5	2
	Projection	10	10	$1e^{-3}$	$1e^{-3}$	$1e^{-6}$	5.5	-

Camelyon17 In medical applications, the goal is to apply models trained on a comparatively small set of hospitals to a larger number of hospitals. For this application, we study images of tissue slides under a microscope to determine whether a patient has cancer or not. Shifts in patient populations, slide staining, and image acquisition can impede model accuracy in previously unseen hospitals. Camelyon17 comprises images of tissue patches from five different hospitals. While the first three hospitals are the source domains (302,436 examples), the forth and fifth are the validation (34,904 examples) and test domain (85,054 examples), respectively.

We deviate from the specifications made in (Koh et al., 2021) in terms of the FE. We use the FE from Feng et al. (2020); Peng et al. (2019) since we observed a higher mean accuracy and faster training than with the by Koh et al. (2021) originally proposed DenseNet-121 FE (Huang et al., 2017). We trained the FE from scratch. Both, ERM and the DG were trained over 250 epochs with early stopping, a learning rate of 0.001, which is reduced by a factor of 0.2 if the cross-entropy loss has not improved after 10 epochs. All results were aggregated over ten runs.

FMoW Analyzing satellite images with machine learning (ML) models may enable novel possibilities in tackling global sustainability and economic challenges such as population density mapping and deforestation tracking. However, satellite imagery changes over time due to human behavior (e.g., infrastructure development), and the extent of change is different in each region. The Functional Map of the World (FMoW) dataset consists of satellite images from different continents and years: training (76,863 images; between 2002–2013), validation (19,915 images; between 2013 and 2016), and test (22,108 images, between 2016–2017). The objective is to determine one of 62 building types (e.g., shopping malls) and land-use.

As instructed in Koh et al. (2021), we used the DenseNet-121 pre-trained on ImageNet without L2-regularization. For the optimization, we use the Adam optimizer with a learning rate of 1e-4, which is decayed by a factor of 0.96 per epoch. The models were trained for 50 epochs with early stopping and a batch size of 64. Additionally, we report the worst-region accuracy, which is a specific metric used for FMoW. This worst-region accuracy reports the worst accuracy across the following regions: Asia, Europe, Africa, America, and Oceania (see Koh et al. (2021) for the details). Again, we report the results over three runs.

Amazon Recent research shows that consumer-facing machine learning application large performance disparities across different set of users. To study this performance disparities, WILDS (Koh et al., 2021) leverages a variant of the Amazon Review dataset. The Aamazon-WILDS dataset is composed of data from 3,920 domains (number of reviewers) and the task is a multi-class sentiment classification, where the model receives a review text and has to predict the rating from one to five. To split this dataset, a between training, validation, and test disjoint set of reviewers is used: training (245,502 reviews from 1,252 reviewers), validation (100,050 reviews from 1,334 reviewers), test (100,050 reviews from 1,334 reviewers).

For the experiments and baseline models, we use the specifications made in Koh et al. (2021). As for the FE, we used DistilBERT-base-uncased models. For ERM, we use a batch size of 8, learning rate 1e-5, L2 regularization of 0.01, 3 epochs with early stopping and 512 as the maximum length of tokens. For training the DG layer, we used the same specifications as made for ERM. The performance is measured in 10th percentile accuracy.

iWildsCam Wildlife camera traps offer an excellent possibility to understand and monitor biodiversity loss. However, images from different camera traps differ in illumination, color, camera angle, background, vegetation, and relative animal frequencies. We use the iWildsCam dataset consisting of 323 different camera traps positioned in different locations worldwide. In the dataset, we refer to different locations of camera traps as different domains, in particular 243 training traps (129,809 images), 32 validation traps (14,961 images), and 48 test traps (42,791 images). The objective is to classify one of 182 animal species.

Following the instructions by Koh et al. (2021), we used again the ResNet50 pre-trained on ImagNet (He et al., 2016). For ERM, we used a learning rate of 3e-5 and no L2-regularization. The models were trained for 12 epochs with a batch size of 16 with the Adam optimizer. In addition to the accuracy, we report the macro F1-score to evaluate the performance on rare species (see Koh et al. (2021) for details). All results were aggregated over three runs.

RxRx1 In biomedical research areas such as genomics or drug discovery, high-throughput screening techniques generate a vast amount of data in several batches. Because experimental designs cannot fully mitigate the effects of confounding variables like temperature, humidity, and measurements across batches, this creates heterogeneity in the observed datasets (commonly known as batch effect). The RxRx1 dataset comprises images obtained by fluorescent microscopy from 51 domains (disjoint experiments): training (40,612 images, 33 domains), validation (9,854 images, 4 domains), and test (34,432 images, 14 domains). The aim is to classify one of 1,139 genetic treatments. All results were aggregated over three runs.

We conducted the RxRx1 experiments in accordance with the specifications made in (Koh et al., 2021). As for the FE, we, thus, used the ResNet50 pre-trained on ImagNet (He et al., 2016). We trained the models using AdamW with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ using a learning rate of 1e-4 and a L2-regularization with strength 1e-5 for 90 epochs with a batch size of 75. We scheduled the learning rate to linearly increase in the first ten epochs and then decreased it following a cosine rate. For trainingthe DG layer, we chose the same parameters as for the ERM. All results were aggregated over three runs.

OBG-MolPCBA In biomedical research, machine learning has the potential to accelerate drug discovery while reducing the experimental overhead due to lowering the number of experiments required. However, to leverage the potential of machine learning, the models need to generalize to molecules structurally different from those seen during training. To study this OOD generalization across molecule scaffolds, we use the OGB-MolPCBA dataset. This dataset is split into the following subsets according to the scaffold structure: training (44,930 domains), validation (31,361 domains), and test (43,739 domains). The task is to classify the presence/absence of 128 biological activities based on a graph representation of a molecule.

In line with Koh et al. (2021), we use a Graph Isomorphism Network (GIN) combined with virtual nodes as the FE. For training ERM and our DG, we use the default parameters: five GNN layers with a dimensionality of 300 and a learning rate of 0.001. We train for 100 epochs using early stopping. As for the performance, we report the mean and standard deviation of the average precision across all scaffolds (domains) over three runs.

CivilComments In the last decades, users have generated a vast amount of text on the Internet, some of which contain toxic comments. Machine learning has been leveraged for automatic text review to flag toxic comments. However, the models are prone to learn spurious correlations between toxicity and information on demographics in the comment, which causes the model performance to drop in specific subpopulations. To study this OOD task, we leverage the modified CivilComment dataset from Koh et al. (2021). Based on text input, the task is to predict a binary label, toxic or non-toxic. The domains are defined according to eight demographic identities: male, female, LGBTQ, Christian, Muslim, other religions, Black, and White. All comments were randomly split into a disjoint training (269,038 comments), validation (45,180 comments), and test (133,782 comments) set.

Again, we follow Koh et al. (2021) and use a DistillBERT-base-uncased model with the following parameters: batch size = 16, learning rate = 1e-5, AdamW optimizer, number of epochs = 5, L2 regularization 0.01, and the maximum number of tokens of 300. We use these default parameters for

Table 13: Parameters for DG Layer in WILDS experiments for the Fine Tuning (FT) and End-to-end training (E2E) Settings.

	Experiment		M	N	λ_{L_1}	λ_{OLS}	λ_{ORTH}	σ	κ
WILDS Benchmark	FT and E2E	CS	5	10	$1e^{-3}$	$1e^{-3}$	-	4	2
		MMD	5	10	$1e^{-3}$	$1e^{-3}$	-	4	2
		Projection	5	10	-	$1e^{-3}$	$1e^{-3}$	16	-

training our DG layer. The performance is measured in the worst-group accuracy and we report mean and standard deviation across five runs.

PovertyMap As the FMoW example shows, satellite images in combination with machine learning models can been used to monitor sustainability and economic challenges on a global scale. Another application of these satellite images is poverty estimation across different spatial regions. However, there exists a lack of labels for developing countries since obtaining the ground truth is expensive, which makes this application attractive for machine learning models. To study the OOD generalization to unseen countries, we use a modified version of the poverty mapping dataset of WILDS (Koh et al., 2021). The task is to predict a real-valued aset wealth index between 1 and 5 based on a multi-spectral satellite image. The domain refers to the country and whether the the the image is from a rural or urban are. In contrast to the other datasets, this dataset is split in five different folds, whereby in each fold the the training, valdiation and test set contains a disjoint set of countries, however, data from both rural and urban regions. The avergae size of each set across the 5 folds is for the training ~10,000 images (13-14 countries), ~4,000 images (4-5 different countries), and for the test set ~4,000 images (13-14 countries).

We follow Koh et al. (2021) and use a pre-trained ResNet-18 model minimizing the sqarred error loss. For the optimization, we rely on the Adam optimizer with the following parameters: learning rate of 1e-3 with a decay of 0.96 per epoch, batch size of 64 and early stopping based on the OOD evaluation score. For evaluation, we report the Pearson correlation (r) between the predicted and actual asset wealth indices across the five different folds.

General benchmark methods Following the WILDS benchmarking procedure (Koh et al., 2021), we compare our proposed DG layer to the following baselines. First, empirical risk minimization (ERM), which minimizes the average training loss over the pooled dataset. Second, a group of DG algorithms provided by the WILDS benchmark, namely, Coral, Fish, IRM, and DRO. The Coral algorithm introduces a penalty for differences in means and covariances of the domains feature distributions. The Fish algorithm achieves DG by approximating an inter-domain gradient matching objective, i.e., maximizing the inner product between gradients from different domains (Shi et al., 2021). Conceptually, Fish learns feature representations that are invariant across domains. Invariant risk minimization (IRM) introduces a penalty for feature distributions with different optimal classifiers for each domain (Arjovsky et al., 2019). The idea is to enable OOD generalization by learning domain-invariant causal predictors. Lastly, group distributionally robust optimization (DRO) explicitly minimizes the training loss on the worst-case domain (Sagawa et al., 2020; Hu et al., 2018).

In addition to the baselines originally presented in Koh et al. (2021), we consider the following more recent DG baselines. First, we describe LISA, which instead of regularizing the internal representations for generalization, seeks to learn domain-invariant predictors with selective data augmentation Yao et al. (2022). Common Gradient Descent (CGD), introduced by Piratla et al. (2021), is based on Group-DRO. However, it proposes to focus not on groups with the worst regularization but on common groups that enable generalization. Last, Adaptive Risk Minimization using batch normalization (ARM-BN) by Zhang et al. (2021) is different from the methods presented since it adapts to previously unseen domains during test time using unlabeled observations from this test domain.