Domain Generalization using Causal Matching

Divyat Mahajan[1] **Shruti Tople**[2] **Amit Sharma**[1]

Abstract

In the domain generalization literature, a common objective is to learn representations independent of the domain after conditioning on the class label. We show that this objective is not sufficient: there exist counter-examples where a model fails to generalize to unseen domains even after satisfying class-conditional domain invariance. We formalize this observation through a structural causal model and show the importance of modeling *within-class* variations for generalization. Specifically, classes contain *objects* that characterize specific causal features, and domains can be interpreted as interventions on these objects that change non-causal features. We highlight an alternative condition: inputs across domains should have the same representation if they are derived from the same object. Based on this objective, we propose matching-based algorithms when base objects are observed (e.g., through data augmentation) and approximate the objective when objects are not observed (MatchDG). Our simple matching-based algorithms are competitive to prior work on out-of-domain accuracy for rotated MNIST, Fashion-MNIST, PACS, and Chest-Xray datasets. Our method MatchDG also recovers ground-truth *object matches*: on MNIST and Fashion-MNIST, top-10 matches from MatchDG have over 50% overlap with ground-truth matches.

1. Introduction

Domain generalization is the task of learning a machine learning model that can generalize to unseen data distributions, after training on more than one data distributions. For example, a model trained on hospitals in one region may be deployed to another, or an image classifier may be deployed on slightly rotated images. Typically, it is assumed that the different domains share some "stable" features whose relationship with the output is invariant across domains (Piratla et al., 2020) and the goal is to learn those features. A popular class of methods aim to learn representations that are independent of domain *conditional on class* (Li et al., 2018c;d; Ghifary et al., 2016; Hu et al., 2019), based on evidence of their superiority (Zhao et al., 2019) to methods that learn representations that are marginally independent of domain (Muandet et al., 2013; Ganin et al., 2016).

In this work, we show that the class-conditional domain-invariant objective for representations is insufficient. We provide counter-examples where a feature representation satisfies the objective but still fails to generalize to new domains, both theoretically and empirically. Specifically, when the distribution of the stable features to be learnt varies across domains, class-conditional objective is insufficient to learn the stable features (they are optimal only when the distribution of stable features is the same across domains). Differing distributions of stable features within the same class label is common in real-world datasets, e.g., in digit recognition, the stable feature *shape* may differ based on people's handwriting, or medical images may differ based on variation in body characteristics across people. Our investigation reveals the importance of considering within-class variation in the stable features.

To derive a better objective for domain generalization, we represent the within-class variation in stable features using a structural causal model, building on prior work (Heinze-Deml & Meinshausen, 2019) from single-domain generalization. Specifically, we construct a model for the data generation process that assumes each input is constructed from a mix of stable (*causal*) and domain-dependent (*non-causal*) features, and only the stable features cause the output. We consider domain as a special intervention that changes the non-causal features of an input, and posit that an ideal classifier should be based only on the causal features. Using d-separation, we show that the correct objective is to build a representation that is invariant conditional on each *object*, where an object is defined as a set of inputs that share the same causal features (e.g., photos of the same person from different viewpoints or augmentations of an image in different rotations, color or background). When the object variable is observed (e.g., in self-collected data or by dataset augmentation), we propose a *perfect-match* regularizer for

[1] Microsoft Research, India [2] Microsoft Research, UK.. Correspondence to: Divyat Mahajan <divyatmahajan@gmail.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

domain generalization that minimizes the distance between representations of the same object across domains.

In practice, however, the underlying objects are not always known. We therefore propose an approximation that aims to learn which inputs share the same object, under the assumption that inputs from the same class have more similar causal features than those from different classes. Our algorithm, `MatchDG` is an iterative algorithm that starts with randomly matched inputs from the same class and builds a representation using contrastive learning such that inputs sharing the same causal features are closer to one another. While past work has used contrastive loss to regularize the empirical risk minimization (ERM) objective (Dou et al., 2019), we demonstrate the importance of a two-phase method that first learns a representation independent of the ERM loss, so that classification loss does not interfere with the learning of stable features. In datasets with data augmentations, we extend `MatchDG` to also use the perfect object matches obtained from pairs of original and augmented images (`MDGHybrid`).

We evaluate our matching-based methods on rotated MNIST and Fashion-MNIST, PACS and Chest X-ray datasets. On all datasets, the simple methods `MatchDG` and `MDGHybrid` are competitive to state-of-the-art methods for out-of-domain accuracy. On the rotated MNIST and Fashion-MNIST datasets where the ground-truth objects are known, `MatchDG` learns to makes the representation more similar to their ground-truth matches (about 50% overlap for top-10 matches), even though the method does not have access to them. Our results with simple matching methods show the importance of enforcing the correct invariance condition.

Contributions. To summarize, our contributions include:
1). An object-invariant condition for domain generalization that highlights a key limitation of previous approaches,
2). When object information is not available, a two-phase, iterative algorithm to approximate object-based matches.
Also, the code repository can be accessed at: `https://github.com/microsoft/robustdg`

2. Related Work

Learning common representation. To learn a generalizable classifier, several methods enforce the learnt representation $\Phi(\mathbf{x})$ to be independent of domain marginally or conditional on class label, using divergence measures such as maximum mean discrepancy (Muandet et al., 2013; Li et al., 2018b;c), adversarial training with a domain discriminator (Ganin et al., 2016; Li et al., 2018d; Albuquerque et al., 2020a), discriminant analysis (Ghifary et al., 2016; Hu et al., 2019), and other techniques (Ghifary et al., 2015).

Among them, several works (Zhao et al., 2019; Johansson et al., 2019; Akuzawa et al., 2019) show that the class-conditional methods (Li et al., 2018c;d; Ghifary et al., 2016; Hu et al., 2019) are better than those that enforce marginal domain-invariance of features (Muandet et al., 2013; Ganin et al., 2016; Li et al., 2018b; Albuquerque et al., 2020a), whenever there is a varying distribution of class labels across domains. We show that the class-conditional invariant is also not sufficient for generalizing to unseen domains.

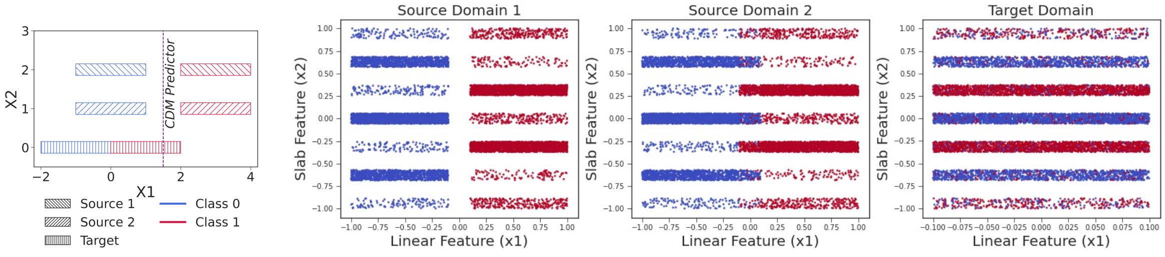
Causality and domain generalization. Past work has shown the connection between causality and generalizable predictors (Peters et al., 2016; Christiansen et al., 2020). There is work on use of causal reasoning for domain adaptation (Gong et al., 2016; Heinze-Deml & Meinshausen, 2019; Magliacane et al., 2018; Rojas-Carulla et al., 2018) that assumes $Y \to X$ direction and other work (Arjovsky et al., 2019; Peters et al., 2016) on connecting causality that assumes $X \to Y$. Our SCM model unites these streams by introducing Y_{true} and labelled Y and develop an invariance condition for domain generalization that is valid under both interpretations. Perhaps the closest to our work is by (Heinze-Deml & Meinshausen, 2019) who use the object concept in single-domain datasets for better generalization. We extend their SCM to the multi-domain setting and use it to show the inconsistency of prior methods. In addition, while (Heinze-Deml & Meinshausen, 2019) assume objects are always observed, we also provide an algorithm for the case when objects are unobserved.

Matching and Contrastive Loss. Regularizers based on matching have been proposed for domain generalization. (Motiian et al., 2017) proposed matching representations of inputs from the same class. (Dou et al., 2019) used a contrastive (triplet) loss to regularize the ERM objective. In contrast to regularizing based on contrastive loss, our algorithm `MatchDG` proceeds in two phases and learns a representation independent of the ERM objective. Such an iterative 2-phase algorithm has empirical benefits, as we will show in Suppl. D.4. Additionally, we propose an ideal object-based matching algorithm when objects are observed.

Other work. Others approaches to domain generalization include meta-learning (Li et al., 2018a; Balaji et al., 2018), dataset augmentation (Volpi et al., 2018; Shankar et al., 2018), parameter decomposition (Piratla et al., 2020; Li et al., 2017), and enforcing domain-invariance of the optimal $P(Y|\Phi(\mathbf{x}))$ (Arjovsky et al., 2019; Ahuja et al., 2020). We empirically compare our algorithm to some of them.

3. Insufficiency of class-conditional invariance

Consider a classification task where the learning algorithm has access to i.i.d. data from m domains, $\{(d_i, \mathbf{x}_i, y_i)\}_{i=1}^n \sim (D_m, \mathcal{X}, \mathcal{Y})^n$ where $d_i \in D_m$ and $D_m \subset \mathcal{D}$ is a set of m domains. Each training input $(d, \mathbf{x}, y)$ is sampled from an unknown distribution



(a) Simple Example

(b) Slab Dataset (*Slab (y-axis) is the stable feature)*

Figure 1. Two datasets showing the limitations of class-conditional domain-invariance objective. a) The CDM predictor is domain-invariant given the class label but does not generalize to the target domain; b) Colors denote the two ground-truth class labels. For class prediction, the linear feature exhibits varying level of noise across domains. The stable slab feature also has noise but it is invariant across domains.

$\mathcal{P}_m(D, X, Y)$. The domain generalization task is to learn a single classifier that generalizes well to unseen domains $d' \notin D_m$ and to new data from the same domains (Shankar et al., 2018). The optimum classifier can be written as: $f^* = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(d,\mathbf{x},y)\sim\mathcal{P}}[l(y^{(d)}, f(\mathbf{x}^{(d)}))]$, where $(d, \mathbf{x}, y) \sim \mathcal{P}$ over $(\mathcal{D}, \mathcal{X}, \mathcal{Y})$.

As mentioned above, a popular line of work enforces that the learnt representation $\Phi(\mathbf{x})$ be independent of domain conditional on the class (Li et al., 2018c;d; Ghifary et al., 2016; Hu et al., 2019), $\Phi(\mathbf{x}) \perp\!\!\!\perp D|Y$. Below we present two counter-examples showing that the class-conditional objective is not sufficient.

3.1. A simple counter-example

We construct an example where $\Phi(\mathbf{x}) \perp\!\!\!\perp D|Y$, but still the classifier does not generalize to new domains. Consider a two dimensional problem where $x_1 = x_c + \alpha_d; x_2 = \alpha_d$ where x_c and α_d are unobserved variables, and α_d varies with domain (Figure 1(a)). The true function depends only on the stable feature x_c, $y = f(x_c) = I(x_c \geq 0)$. Suppose there are two training domains with $\alpha_1 = 1$ for domain 1 and $\alpha_2 = 2$ for domain 2, and the test domain has $\alpha_3 = 0$ (see Figure 1(a)). Suppose further that the conditional distribution of X_C given Y is a uniform distribution that changes across domains: for domain 1, $X_c|Y = 1 \sim \mathcal{U}(1, 3); X_c|Y = 0 \sim \mathcal{U}(-2, 0)$; and for domain 2, $X_c|Y = 1 \sim \mathcal{U}(0, 2); X_c|Y = 0 \sim \mathcal{U}(-3, -1)$. Note that the distributions are picked such that $\phi(x_1, x_2) = x_1$ satisfies the conditional distribution invariant, $\phi(x) \perp\!\!\!\perp D|Y$. The optimal ERM classifier based on this representation, $(I(x_1 \geq 1.5)$ has 100% train accuracy on both domains. But for the test domain with $\alpha_d = 0; X_c|Y = 1 \sim \mathcal{U}(0, 2); X_c|Y = 0 \sim U(-2, 0)$, the classifier fails to generalize. It obtains 62.5% test accuracy (and 25% accuracy on the positive class), even though its representation satisfies class-conditional domain invariance. In comparison, the ideal representation is $x_1 - x_2$ which attains 100% train accuracy and 100% test domain accuracy, and does not satisfy the class-conditional invariant.

The above counter-example is due to the changing distribution of x_c across domains. If $P(X_c|Y)$ stays the same across domains, then class-conditional methods would not incorrectly pick x_1 as the representation. Following (Akuzawa et al., 2019), we claim the following (proof in Suppl. B.3).

Proposition 1. *Under the domain generalization setup as above, if $P(X_c|Y)$ remains the same across domains where x_c is the stable feature, then the class-conditional domain-invariant objective for learning representations yields a generalizable classifier such that the learnt representation $\Phi(\mathbf{x})$ is independent of the domain given x_c. Specifically, the entropy $H(d|x_c) = H(d|\Phi, x_c)$.*

However, if $P(X_C|Y)$ changes across domains, then we cannot guarantee the same: $H(d|x_c)$ and $H(d|\Phi, x_c)$ may not be equal. For building generalizable classifiers in such cases, this example shows that we need an additional constraint on Φ, $H(d|x_c) = H(d|\Phi, x_c)$; i.e. domain and representation should be independent conditioned on x_c.

3.2. An empirical study of class-conditional methods

As a more realistic example, consider the slab dataset introduced for detecting simplicity bias in neural networks (Shah et al., 2020) that contains a feature with spurious correlation. It comprises of two features and a binary label; (x_1) has a linear relationship with the label and the other feature (x_2) has a piece-wise linear relationship with the label which is a stable relationship. The relationship of the linear feature with the label changes with domains (A.1); we do so by adding noise with probability $\epsilon = 0$ for domain 1 and $\epsilon = 0.1$ for domain 2. On the third (test) domain, we add noise with probability 1 (see Figure 1(b)). We expect that methods that rely on the spurious feature x_1 would not be able to perform well on the out-of-domain data.

The results in Table 1 (implementation details in Appendix A.1) show that ERM is unable to learn the slab feature, as evident by poor generalization to the target domain, despite very good performance on the source domains. We also show that methods based on learning invariant representations by unconditional (DANN, MMD, CORAL) and conditional distribution matching (CDANN, C-MMD, C-CORAL), and matching same-class inputs (RandomMatch) (Motiian et al., 2017) fail to learn the stable slab feature. Note that Proposition 1 suggested the failure of conditional distribution matching (CDM) algorithms when the distribution of stable feature (slab feature) is different across the source domains. However, the slab dataset has similar distribution of stable feature (slabs) across the source domains, yet the CDM algorithms fail to generalize to the target domain. It can be explained by considering the spurious linear feature, which can also satisfy the CDM constraint by "shifting" the y-conditional distributions along the linear feature. We conjecture that the model may first learn the linear feature due to its simplicity (Shah et al., 2020), and then retain the spurious linear feature upon further optimization since it satisfies the CDM constraint. This shows that the CDM methods can empirically fail even when there is an equal distribution of stable features across domains.

How can we ensure that a model learns the stable, generalizable feature x_2? We turn to our example above, where the required invariant was that the representation $\Phi(\mathbf{x})$ should be independent of domain given the stable feature. We apply this intuition and construct a model that enforces that the learnt representation be independent of domain given x_2. We do so by minimizing the ℓ_2-norm of the representations for data points from different domains that share the same slab value (details of the *PerfectMatch* method in Section 4.3). The results improve substantially: out-of-domain accuracy is now 78%.

In the next section, we formalize the intuition of conditioning on stable features x_c using a causal graph, and introduce the concept of *objects* that act as proxies of stable features.

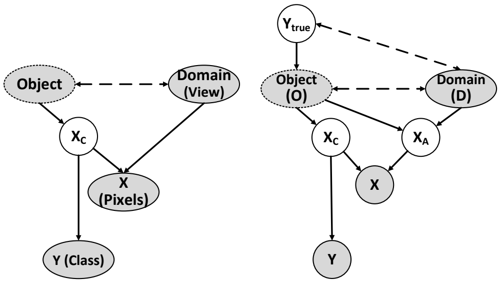
4. A Causal View of Domain Generalization

4.1. Data-generating process

Figure 2(a) shows a structural causal model (SCM) that describes the data-generating process for the domain generalization task. For intuition, consider a task of classifying the type of item or screening an image for a medical condition. Due to human variability or by design (using data augmentation), the data generation process yields variety of images for each class, sometimes multiple views for the *same object*. Here each view can be considered as a different *domain* D, the label for item type or medical condition as the class Y, and the image pixels as the features X. Photos of the same item or the same person correspond to a common *object* variable, denoted by O. To create an image, the data-generating process first samples an object and view (domain) that may be correlated to each other (shown with dashed arrows). The pixels in the photo are caused by both the object and the view, as shown by the two incoming arrows to X. The object also corresponds to high-level *causal* features X_C that are common to any image of the same object, which in turn are used by humans to label the class Y. We call X_C as causal features because they directly cause the class Y.

Table 1. Slab Dataset: Source domains with noisy linear component with probability 0.0 and 0.1, target domain with noise 1.0. Mean and standard deviation over 10 different seed values for each method. The results for DANN (Ganin et al., 2016), CDANN (Li et al., 2018d), MMD, C-MMD (Li et al., 2018b), CORAL, C-CORAL (Sun & Saenko, 2016) were computed by using their implementations in DomainBed (Gulrajani & Lopez-Paz, 2020).

Method	Source 1	Source 2	Target
ERM	**100.0** (0.0)	96.0 (0.25)	57.6 (6.58)
DANN	99.9 (0.07)	94.8 (0.25)	53.0 (1.41)
MMD	99.9 (0.01)	95.9 (0.27)	62.9 (5.01)
CORAL	99.9 (0.01)	96.0 (0.27)	63.1 (5.86)
RandMatch	**100.0** (0.0)	96.1 (0.22)	59.5 (3.50)
CDANN	99.9 (0.01)	96.0 (0.27)	55.9 (2.47)
C-MMD	99.9 (0.01)	96.0 (0.27)	58.9 (3.43)
C-CORAL	99.9 (0.01)	96.0 (0.27)	64.7 (4.69)
PerfMatch	99.9 (0.05)	**97.8** (0.28)	**77.8** (6.01)



(a) Image classification. (b) General SCM.

Figure 2. Structural causal models for the data-generating process. Observed variables are shaded; dashed arrows denote correlated nodes. *Object* may not be observed.

The above example is typical of a domain generalization problem; a general SCM is shown in Figure 2(b), similar to the graph in (Heinze-Deml & Meinshausen, 2019). In general, the underlying *object* for each input $\mathbf{x}_i^{(d)}$ may not be observed. Analogous to the object-dependent (*causal*) features X_C, we introduce a node for domain-dependent high-level features of the object X_A. Changing the domain can be seen as an intervention: for each observed $\mathbf{x}_i^{(d)}$, there are a set of (possibly unobserved) counterfactual inputs $\mathbf{x}_j^{(d')}$ where $d \neq d'$, such that all correspond to the same object (and thus share the same X_C). For completeness, we also show the true unobserved label of the object which led to its generation as Y_{true} (additional motivation for the causal graph is in Suppl. B.1). Like the object O, Y may be correlated with the domain D. Extending the model in (Heinze-Deml & Meinshausen, 2019), we allow that objects can be correlated with the domain conditioned on Y_{true}. As we shall see, considering the relationship of the *object* node becomes the key piece for developing the invariant condition. The SCM corresponds to the following non-parametric equations.

$$o := g_o(y_{true}, \epsilon_o, \epsilon_{od}) \qquad \mathbf{x}_c = g_{xc}(o)$$
$$\mathbf{x}_a := g_{xa}(d, o, \epsilon_{xa}) \qquad \mathbf{x} := g_x(\mathbf{x}_c, \mathbf{x}_a, \epsilon_x) y := h(\mathbf{x}_c, \epsilon_y)$$

where g_o, g_{xc}, g_{xa}, g_x and h are general non-parametric functions. The error ϵ_{od} is correlated with domain d whereas ϵ_o, ϵ_{xa}, ϵ_x and ϵ_y are mutually independent error terms that are independent of all other variables. Thus, noise in the class label is independent of domain. Since x_c is common to all inputs of the same object, g_{xc} is a deterministic function of o. In addition, the SCM provides conditional-independence conditions that all data distributions $\mathcal{P}$ must satisfy, through the concept of d-separation (Suppl. B.2) and the perfect map assumption (Pearl, 2009).

4.2. Identifying the invariance condition

From Figure 2(b), X_C is the node that causes Y. Further, by d-separation, the class label is independent of domain conditioned on X_C, $Y \perp\!\!\!\perp D | X_C$. Thus our goal is to learn y as $h(\mathbf{x}_c)$ where $h : \mathcal{C} \to \mathcal{Y}$. The ideal loss-minimizing function f^* can be rewritten as (assuming $\mathbf{x}_c$ is known):

$$\arg\min_f \mathbb{E}_{(d,\mathbf{x},y)} l(y, f(\mathbf{x})) = \arg\min_h \mathbb{E}[l(y, h(\mathbf{x}_c))] \qquad (1)$$

Since X_C is unobserved, this implies that we need to learn it through a representation function $\Phi : \mathcal{X} \to \mathcal{C}$. Together, $h(\Phi(x))$ leads to the desired classifer $f : \mathcal{X} \to \mathcal{Y}$.

Negative result on identification. Identification of causal features is a non-trivial problem (Magliacane et al., 2018). We first show that x_C is unidentifiable given observed data $P(X, Y, D, O)$ over multiple domains. Given the same probability distribution $P(X, Y, D, O)$, multiple values of X_C are possible. Substituting for o in the SCM equations, we obtain, $y = h(g_{xc}(o), \epsilon_y); \mathbf{x} = g_x(g_{xc}(o), g_{xa}(d, o, \epsilon_{xa}), \epsilon_x)$. By choosing g_x and h appropriately, different values of g_{xc} (that determine x_c from o) can lead to the same observed values for (y, d, o, x). The proof for the following proposition is in Supp. B.4.

Proposition 2. *Given observed data distribution $P(Y, X, D, O)$ that may also include data obtained from interventions on domain D, multiple values of X_C yield exactly the same observational and interventional distributions and hence X_c is unidentifiable.*

4.3. A "perfect-match" invariant

In the absence of identifiability, we proceed to find an invariant that can characterize X_c. By the d-separation criterion, we see that X_C satisfies two conditions: **1)** $X_C \perp\!\!\!\perp D | O$, **2)** $X_C \not\perp\!\!\!\perp O$; where O refers to the object variable and D refers to a domain. The first is an invariance condition: X_C does not change with different domains for the same object. To enforce this, we stipulate that the average pairwise distance between $\Phi(x)$ for inputs across domains for the same object is 0, $\sum_{\Omega(j,k)=1; d \neq d'} \text{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) = 0$. Here $\Omega : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ is a *matching* function that is 1 for pairs of inputs across domains corresponding to the same object, and 0 otherwise.

However, just the above invariance will not work: we need the representation to be informative of the object O (otherwise even a constant Φ minimizes the above loss). Therefore, the second condition stipulates that X_C should be informative of the object, and hence about Y. We add the standard classification loss, leading to constrained optimization,

$$f_{\texttt{perfectmatch}} = \arg\min_{h,\Phi} \sum_{d=1}^{m} L_d(h(\Phi(X)), Y)$$
$$\texttt{s.t.} \sum_{\Omega(j,k)=1; d \neq d'} \text{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) = 0 \qquad (2)$$

where $L_d(h(\Phi(X), Y)) = \sum_{i=1}^{n_d} l(h(\Phi(\mathbf{x}_i^{(d)}), y_i^{(d)})$. Here f represents the composition $h \circ \Phi$. E.g., a neural network with $\Phi(x)$ as its rth layer, and h being the rest of the layers.

Note that there can be multiple $\Phi(\mathbf{x})$ (e.g., linear transformations) that are equally good for the prediction task. Since x_c is unidentifiable, we focus on the set of *stable* representations that are d-separated from D given O. Being independent of domain given the object, they cannot have any association with X_a, the high-level features that directly depend on domain (Figure 2b). The proof for the next theorem is in Suppl. B.5.

Theorem 1. *For a finite number of domains m, as the number of examples in each domain $n_d \to \infty$,*
1. The set of representations that satisfy the condition $\sum_{\Omega(j,k)=1; d \neq d'} \text{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) = 0$ contains the optimal $\Phi(\mathbf{x}) = X_C$ that minimizes the domain generalization loss in (1).
2. Assuming that $P(X_a | O, D) < 1$ for every high-level feature X_a that is directly caused by domain, and for P-admissible loss functions (Miller et al., 1993) whose minimization is conditional expectation (e.g., ℓ_2 or cross-entropy), a loss-minimizing classifier for the following loss is the true function f^*, for some value of λ.

$$f_{\texttt{perfectmatch}} = \arg\min_{h,\Phi} \sum_{d=1}^{m} L_d(h(\Phi(X)), Y) +$$
$$\lambda \sum_{\Omega(j,k)=1; d \neq d'} \text{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) \qquad (3)$$

4.4. Past work: Learning common representation

Using the SCM, we now compare the proposed invariance condition to domain-invariant and class-conditional domain-invariant objectives. d-separation results show that both these objectives are incorrect: in particular, the class-conditional objective $\Phi(\mathbf{x}) \perp\!\!\!\perp D|Y$ is not satisfied by X_C, ($X_C \not\perp\!\!\!\perp D|Y_{true}$) due to a path through O. Even with infinite data across domains, they will not learn the true X_C. The proof is in Suppl. B.6.

Proposition 3. *The conditions enforced by domain-invariant ($\Phi(x) \perp\!\!\!\perp D$) or class-conditional domain-invariant ($\Phi(x) \perp\!\!\!\perp D|Y$) methods are not satisfied by the causal representation X_C. Thus, without additional assumptions, the set of representations that satisfy any of these conditions does not contain X_C, even as $n \rightarrow \infty$.*

5. MatchDG: Matching without objects

When object information is available, Eq. (3) provides a loss objective to build a classifer using causal features. However, object information is not always available, and in many datasets there may not be a perfect "counterfactual" match based on same object across domains. Therefore, we propose a two-phase, iterative contrastive learning method to approximate object matches.

The object-invariant condition from Section 4.2 can be interpreted as matching pairs of inputs from different domains that share the same X_C. To approximate it, our goal is to learn a matching $\Omega : \mathcal{X} \times \mathcal{X} \rightarrow \{0, 1\}$ such that pairs having $\Omega(\mathbf{x}, \mathbf{x}') = 1$ have low difference in $\mathbf{x}_c$ and $\mathbf{x}'_c$. We make the following assumption.

Assumption 1. *Let $(\mathbf{x}_i^{(d)}, y)$, $(\mathbf{x}_j^{(d')}, y)$ be any two points that belong to the same class, and let $(\mathbf{x}_k^{(d)} y')$ be any other point that has a different class label. Then the distance in causal features between $\mathbf{x}_i$ and $\mathbf{x}_j$ is smaller than that between $\mathbf{x}_i$ and $\mathbf{x}_k$ or $\mathbf{x}_j$ and $\mathbf{x}_k$: $\text{dist}(x_{c,i}^{(d)}, x_{c,j}^{(d')}) \leq \text{dist}(x_{c,i}^{(d)}, x_{c,k}^{(d')})$ and $\text{dist}(x_{c,j}^{(d)}, x_{c,i}^{(d')}) \leq \text{dist}(x_{c,j}^{(d)}, x_{c,k}^{(d')})$.*

5.1. Two-phase method with iterative matches

To learn a matching function Ω, we use unsupervised contrastive learning from (Chen et al., 2020; He et al., 2019) and adapt it to construct an iterative `MatchDG` algorithm that updates the both the representation and matches after each epoch. The algorithm relies on the property that two inputs from the same class have more similar causal features than inputs from different classes.

Contrastive Loss. To find matches, we optimize a contrastive representation learning loss that minimizes distance between same-class inputs from different domains in comparison to inputs from different classes across domains. Adapting the contrastive loss for a single domain (Chen et al., 2020), we consider *positive* matches as two inputs with the same class but different domains, and *negative* matches as pairs with different classes. For every positive match pair $(\mathbf{x}_j, \mathbf{x}_k)$, we propose a loss where τ is a hyperparameter, B is the batch size, and $\text{sim}(\mathbf{a}, \mathbf{b}) = \Phi(\mathbf{x}_a)^T \Phi(\mathbf{x}_b) / \|\Phi(\mathbf{x}_a)\| \, \|\Phi(\mathbf{x}_b)\|$ is the cosine similarity.

$$l(\mathbf{x}_j, \mathbf{x}_k) = -\log \frac{e^{\text{sim}(j,k)/\tau}}{e^{\text{sim}(j,k)/\tau} + \sum_{i=0, y_i \neq y_j}^{B} e^{\text{sim}(j,i)/\tau}} \tag{4}$$

Algorithm 1 MatchDG

```
In: Dataset $(d_i, x_i, y_i)_{i=1}^n$ from m domains, $\tau$, t
Out: Function $f : \mathcal{X} \rightarrow \mathcal{Y}$
Create random match pairs $\Omega_Y$.
Build a $p * q$ data matrix $\mathcal{M}$.
Phase I
while notconverged do
  for $batch \sim \mathcal{M}$ do
    Minimize contrastive loss (4).
  end for
  if epoch % t == 0 then
    Update match pairs using $\Phi_{epoch}$.
  end if
end while
Phase II
Compute matching based on $\Phi$.
Minimize the loss (3) with learnt match function $\Phi$ to obtain $f$.
```

Iterative matching. Our key insight is to update the positive matches during training. We start training with a random set of positive matches based on the classes, but after every t epochs, we update the positive matches based on the nearest same-class pairs in representation space and iterate until convergence. Hence for each anchor point, starting with an initial set of positive matches, in each epoch a representation is learnt using contrastive learning; after which the positive matches are themselves updated based on the closest same-class data points across domains in the representation. As a result, the method differentiates between data points of the same class instead of treating all of them as a single unit. With iterative updates to the positive matches, the aim is to account for intra-class variance across domains and match data points across domains that are more likely to share the same base object. In Suppl. D.6, we compare the gains due to the proposed iterative matching versus standard contrastive training.

Obtaining the final representation completes Phase I of the algorithm. In Phase II, we use this representation to compute a new match function based on closest same-class pairs and apply Eq. (3) to obtain a classifier regularized on those matches.

The importance of using two phases. We implement `MatchDG` as a 2-phase method, unlike previous methods (Motiian et al., 2017; Dou et al., 2019) that employed class-based contrastive loss as a regularizer with ERM. This is to avoid the classification loss interfering with the goal of learning an invariant representation across domains (e.g., in datasets where one of the domains has many more samples than others). Therefore, we first learn the match function using only the contrastive loss. Our results in Suppl. D.4 show that the two-phase method provides better overlap with ground-truth perfect matches than optimizing classification and matching simultaneously.

To implement MatchDG we build a $p \times q$ data matrix containing $q-1$ positive matches for each input and then sample mini-batches from this matrix. The last layer of the contrastive loss network is considered as the learnt representation (see Algorithm 1; details are in Suppl. C.1).

5.2. MDG Hybrid

While MatchDG assumes no information about objects, it can be easily augmented to incorporate information about known objects. For example, in computer vision, a standard practice is to augment data by performing rotations, horizontal flips, color jitter, etc. These self-augmentations provide us with access to known objects, which can included as perfect-matches in MatchDG Phase-II by adding another regularizer to the loss from Eq 3. We name this method MDGHybrid and evaluate it alongside MatchDG for datasets where we can perform self augmentations.

6. Evaluation

We evaluate out-of-domain accuracy of MatchDG on two simulated benchmarks by Piratla et al. (2020), Rotated MNIST and Fashion-MNIST, on PACS dataset (Li et al., 2017), and on a novel Chest X-rays dataset. In addition, using the simulated datasets, we inspect the quality of matches learnt by MatchDG by comparing them to ground-truth object-based matches. For PACS and Chest X-rays, we also implement MDGHybrid that uses augmentations commonly done while training neural networks. We compare to 1) ERM: Standard empirical risk minimization, 2) ERM-RandMatch that implements the loss from Eq. (3) but with randomly selected matches from the same class (Motiian et al., 2017), 3) other state-of-the-art methods for each dataset. For all matching-based methods, we use the cross-entropy loss for L_d and ℓ_2 distance for dist in Eq.(3). Details of implementation and the datasets are in Suppl. C.1. All the numbers are averaged over 3 runs with standard deviation in brackets.

Rotated MNIST & Fashion-MNIST. The datasets contain rotations of grayscale MNIST handwritten digits and fashion article images from 0° to 90° with an interval of 15° (Ghifary et al., 2015), where each rotation angle represents a domain and the task is to predict the class label. Since different domains' images are generated from the same base image (object), there exist perfect matches across domains. Following CSD, we report accuracy on 0° and 90° together as the test domain and the rest as the train domains; since these test angles, being extreme, are the hardest to generalize to (standard setting results are in Suppl. D.1, D.2).

PACS. This dataset contains total 9991 images from four domains: Photos (P), Art painting (A), Cartoon (C) and Sketch (S). The task is to classify objects over 7 classes. Following (Dou et al., 2019), we train 4 models with each domain as the target using Resnet-18, Resnet-50 and Alexnet.

Chest X-rays. We introduce a harder real-world dataset based on Chest X-ray images from three different sources: NIH (Wang et al., 2017), ChexPert (Irvin et al., 2019) and RSNA (rsn, 2018). The task is to detect whether the image corresponds to a patient with Pneumonia (1) or not (0). To create spurious correlation, all images of class 0 in the training domains are translated vertically downwards; while no such translation is done for the test domain.

Model Selection. While using a validation set from the test domain may improve classification accuracy, it goes against the problem motivation of generalization to unseen domains. Hence, we use only data from source domains to construct a validation set (except when explicitly mentioned in Table 4, to compare to past methods that use test domain validation).

6.1. Rotated MNIST and Fashion MNIST

Table 2 shows classification accuracy on rotMNIST and rotFashionMNIST for test domains 0° & 90° using Resnet-18 model. On both datasets, MatchDG *outperforms* all baselines. The last column shows the accuracy for an oracle method, ERM-PerfMatch that has access to ground-truth perfect matches across domains. MatchDG's accuracy lies between ERM-RandMatch and ERM-PerfMatch, indicating the benefit of learning a matching function. As the number of training domains decrease, the gap between MatchDG and baselines is highlighted: with 3 source domains for rotFashionMNIST, MatchDG achieves accuracy of 43.8% whereas the next best method ERM-RandMatch achieves 38.4%.

We also evaluate on a simpler 2-layer LeNet (Motiian et al., 2017), and the model from (Gulrajani & Lopez-Paz, 2020) to compare MatchDG to prior works (Ilse et al., 2020; Ganin et al., 2016; Shankar et al., 2018; Goodfellow et al., 2014); the results are in Suppl. D.1, D.2.

Why MatchDG works? We compare the matches returned by MatchDG Phase I (on Resnet-18 network) to the ground-truth perfect matches and find that it has significantly higher overlap than matching based on ERM loss (Table 3). We report three metrics on the representation learnt: percentage of MatchDG matches that are perfect matches, %-age of inputs for which the perfect match is within the top-10 ranked MatchDG matches, and mean rank of perfect matches measured by distance over the MatchDG representation.

On all three metrics, MatchDG finds a representation whose

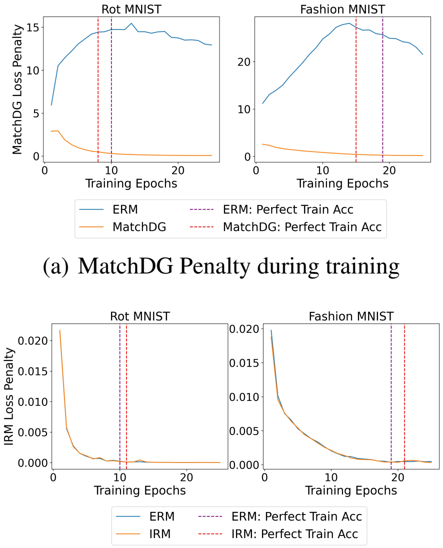
Table 2. Accuracy for Rotated MNIST & Fashion-MNIST datasets on target domains of 0° and 90°. Accuracy for CSD (Piratla et al., 2020), MASF (Dou et al., 2019), IRM (Arjovsky et al., 2019) are reproduced from their code. Results for the other versions of Rotated MNIST with all test angles (LetNet (Motiian et al., 2017), DomainBed (Gulrajani & Lopez-Paz, 2020)) are in Suppl. D.1, D.2.

Dataset	Source	ERM	MASF	CSD	IRM	`RandMatch`	`MatchDG`	`PerfMatch (Oracle)`
Rotated MNIST	15, 30, 45, 60, 75	93.0 (0.11)	93.2 (0.2)	94.5 (0.35)	92.8 (0.53)	93.4 (0.26)	**95.1** (0.25)	96.0 (0.41)
	30, 45, 60	76.2 (1.27)	69.4 (1.32)	77.7 (1.88)	75.7 (1.11)	78.3 (0.55)	**83.6** (1.44)	89.7 (1.68)
	30, 45	59.7 (1.75)	60.8 (1.53)	62.0 (1.31)	59.5 (2.61)	63.8 (3.92)	**69.7** (1.30)	80.4 (1.79)
Rotated Fashion MNIST	15, 30, 45, 60, 75	77.9 (0.13)	72.4 (2.9)	78.7 (0.38)	77.8 (0.02)	77.0 (0.42)	**80.9** (0.26)	81.6 (0.46)
	30, 45, 60	36.1 (1.91)	29.7 (1.73)	36.3 (2.65)	37.8 (1.85)	38.4 (2.73)	**43.8** (1.33)	54.0 (2.79)
	30, 45	26.1 (1.10)	22.8 (1.26)	24.2 (1.69)	26.6 (1.06)	26.9 (0.34)	**33.0** (0.72)	41.8 (1.78)

Table 3. Overlap with perfect matches. top-10 overlap and the mean rank for perfect matches for `MatchDG` and ERM over all training domains. Lower is better for mean rank.

Dataset	Method	Overlap (%)	Top 10 Overlap (%)	Mean Rank
MNIST	ERM	15.8 (0.42)	48.8 (0.78)	27.4 (0.89)
	`MatchDG` (Default)	**28.9** (1.24)	**64.2** (2.42)	**18.6** (1.59)
	`MatchDG` (PerfMatch)	47.4 (2.25)	83.8 (1.46)	6.19 (0.61)
Fashion MNIST	ERM	2.1 (0.12)	11.1 (0.63)	224.3 (8.73)
	`MatchDG` (Default)	**17.9** (0.62)	**43.1** (0.83)	**89.0** (3.15)
	`MatchDG` (PerfMatch)	56.2 (1.79)	87.2 (1.48)	7.3 (1.18)

matches are more consistent with ground-truth perfect matches. For both `rotMNIST` and `rotFashionMNIST` datasets, about 50% of the inputs have their perfect match within top-10 ranked matches based on the representation learnt by `MatchDG` Phase I. About 25% of all matches learnt by `MatchDG` are perfect matches. For comparison, we also show metrics for an (oracle) MatchDG method that is initialized with perfect matches: it achieves better overall and Top-10 values. Similar results for `MatchDG` Phase 2 are in Suppl. D.4. Mean rank for `rotFashionMNIST` may be higher because of the larger sample size $10,000$ per domain; metrics for training with 2000 samples are in Suppl. D.5. To see how the overlap with perfect matches affects accuracy, we simulate random matches with 25%, 50% and 75% overlap with perfect matches (Suppl. Tbl. D.3). Accuracy increases with the fraction of perfect matches, indicating the importance of capturing good matches.



(a) MatchDG Penalty during training

(b) IRM Penalty during training

Figure 3. MatchDG regularization penalty is not trivially minimized even as the training error goes to zero.

MatchDG vs. IRM on zero training error. Since neural networks often achieve zero training error, we also evaluate the effectiveness of the `MatchDG` regularization under this regime. Fig. 3 shows the matching loss term as training proceeds for `rotMNIST` and `rotFashionMNIST`. Even after the model achieves zero training error, we see that plain ERM objective is unable to minimize the matching loss (and thus MatchDG penalty is needed). This is because MatchDG regularization depends on comparing the (last layer) representations, and zero training error does not mean that the representations within each class are the same. In contrast, regularizations that are based on comparing loss between training domains such as the IRM penalty can be satisfied by plain ERM as the training error goes to zero (Fig. 3(b)); similar to Fig. (5) from (Krueger et al., 2020) where ERM can minimize IRM penalty on Colored MNIST.

6.2. PACS dataset

ResNet-18. On the PACS dataset with ResNet-18 architecture (Table 4), our methods are competitive to state-of-the-

Table 4. Accuracy on PACS with ResNet 18 (default), and Resnet 18 with test domain validation. The results for JiGen (Carlucci et al., 2019), DDAIG (Zhou et al., 2020), SagNet (Nam et al., 2019), DDEC (Asadi et al., 2019), were taken from the DomainBed (Gulrajani & Lopez-Paz, 2020) paper. For G2DM (Albuquerque et al., 2020a), CSD (Piratla et al., 2020), RSC (Huang et al., 2020) it was taken from the respective paper. Extensive comparison with other works and std. dev. in results is in Supp E.1.

	P	A	C	S	Average.
ERM	95.38	77.68	78.98	74.75	81.70
JiGen	96.0	79.42	75.25	71.35	80.41
G2DM	93.75	77.78	75.54	77.58	81.16
CSD	94.1	78.9	75.8	76.7	81.4
DDAIG	95.30	**84.20**	78.10	74.70	83.10
SagNet	95.47	83.58	77.66	76.30	83.25
DDEC	**96.93**	83.01	79.39	78.62	84.46
RSC	95.99	83.43	80.31	**80.85**	**85.15**
RandMatch	95.37	78.16	78.83	75.13	81.87
MatchDG	95.93	79.77	80.03	77.11	83.21
MDGHybrid	96.15	81.71	**80.75**	78.79	84.35
G2DM (Test)	94.63	81.44	79.35	79.52	83.34
RandMatch (Test)	95.57	79.09	79.37	77.60	82.91
MatchDG (Test)	96.53	81.32	80.70	79.72	84.56
MDGHybrid (Test)	**96.67**	**82.80**	**81.61**	**81.05**	**85.53**

Table 5. Accuracy on PACS with architecture ResNet 50. The results for IRM (Arjovsky et al., 2019), CORAL (Sun & Saenko, 2016), were taken from the DomainBed (Gulrajani & Lopez-Paz, 2020) paper. The result for RSC (Huang et al., 2020) was taken from their paper. Comparison with other works in Supp E.1.

	P	A	C	S	Average.
DomainBed (ResNet50)	97.8	**88.1**	77.9	79.1	85.7
IRM (ResNet50)	96.7	85.0	77.6	78.5	84.4
CORAL (ResNet50)	97.6	87.7	79.2	79.4	86.0
RSC (ResNet50)	97.92	87.89	82.16	**83.35**	**87.83**
RandMatch (ResNet50)	97.89	82.16	81.68	80.45	85.54
MatchDG (ResNet50)	97.94	85.61	82.12	78.76	86.11
MDGHybrid (ResNet50)	**98.36**	86.74	**82.32**	82.66	87.52

art results averaged over all domains. The MDGHybrid has the highest average accuracy across domains, except compared to DDEC and RSC. These works do not disclose their model selection strategy (whether the results are using source or test domain validation). Therefore, we also report results of MatchDG and MDGHybrid using test domain validation, where MDGHybrid obtains comparable results to the best-performing method. In addition, with DDEC (Asadi et al., 2019), it is not a fair comparison since they use additional style transfer data from Behance BAM! dataset during training.

ResNet-50. We implement MatchDG on Resnet50 model (Table 5) used by the ERM in DomainBed. Adding MatchDG loss regularization improves the accuracy of DomainBed, from 85.7 to 87.5 with MDGHybrid. Also, MDGHybrid performs better than the prior approaches using Resnet50 architecture, except RSC (Huang et al., 2020), whose results (87.83) are close to ours (87.52). Note that we chose a subset of the best-performing baselines for Table 4, 5; an extensive comparison with other works is in Suppl. E.1. Suppl. E.2 gives the results using AlexNet network, and a t-SNE plot (Figure 5) to show the quality of representation learnt by MatchDG.

Table 6. Chest X-Rays data. As an upper bound, training ERM on the target domain itself yields 73.8%, 66.5%, and 59.9% accuracy for RSNA, ChexPert, and NIH respectively.

	RSNA	ChexPert	NIH
ERM	55.1 (2.93)	60.9 (0.51)	53.4 (1.36)
IRM	57.0 (0.75)	63.3 (0.25)	54.6 (0.88)
CSD	58.6 (1.63)	**64.4 (0.88)**	54.7 (0.13)
RandMatch	56.3 (3.38)	55.3 (2.25)	53.1 (0.13)
MatchDG	58.2 (1.25)	59.0 (0.25)	53.2 (0.65)
MDGHybrid	**64.3** (0.75)	60.6 (0.25)	**57.6** (0.13)

6.3. Chest X-rays dataset

Table 6 provides results for the Chest X-rays dataset, where the spurious correlation of vertical translation with the class label in source domains may lead the models to learn an unstable relationship. With RSNA as the target domain, ERM obtains 79.8%, 81.8% accuracy on the source domains while its accuracy drops to 55.1% for the target domain. In contrast, MDGHybrid obtain the highest classification accuracy (8 % above ERM), followed by CSD and MatchDG; while methods like ERM and IRM are more susceptible to spurious correlation. However, on ChexPert as the target domain, CSD and IRM do better than ERM while matching-based methods are not effective. We conjecture these varying trends might be due to the inherent variability in images in the source domains, indicating the challenges of building domain generalization methods for real-world datasets.

7. Conclusion

We presented a causal view of domain generalization that provides an object-conditional objective. Simple matching-based methods perform competitively to state-of-the-art methods on PACS, indicating the importance of choosing the right invariance. The proposed MatchDG uses certain assumptions when objects are unknown. More work needs to be done to develop better matching methods, as indicated by the mixed results on the Chest-Xrays dataset.

Acknowledgements. We would like to thank Adith Swaminathan, Aditya Nori, Emre Kiciman, Praneeth Netrapalli, Tobias Schnabel, Vineeth Balasubramanian and the reviewers who provided us valuable feedback on this work. We also thank Vihari Piratla who helped us with reproducing the CSD method and other baselines.

References

Kaggle: Rsna pneumonia detection challenge, 2018. URL https://www.kaggle.com/c/rsna-pneumonia-detection-challenge.

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

Akuzawa, K., Iwasawa, Y., and Matsuo, Y. Adversarial invariant feature learning with accuracy constraint for domain generalization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 315–331. Springer, 2019.

Albuquerque, I., Monteiro, J., Darvishi, M., Falk, T. H., and Mitliagkas, I. Generalizing to unseen domains via distribution matching, 2020a.

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

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

Asadi, N., Sarfi, A. M., Hosseinzadeh, M., Karimpour, Z., and Eftekhari, M. Towards shape biased unsupervised representation learning for domain generalization. *arXiv preprint arXiv:1909.08245*, 2019.

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

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

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Christiansen, R., Pfister, N., Jakobsen, M. E., Gnecco, N., and Peters, J. A causal framework for distribution generalization. *arXiv e-prints*, pp. arXiv–2006, 2020.

Cohen, J. P., Hashir, M., Brooks, R., and Bertrand, H. On the limits of cross-domain generalization in automated x-ray prediction. *arXiv preprint arXiv:2002.02497*, 2020.

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

D'Innocente, A. and Caputo, B. Domain generalization with domain-specific aggregation modules. In *German Conference on Pattern Recognition*, pp. 187–198. Springer, 2018.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Ghifary, M., Bastiaan Kleijn, W., Zhang, M., and Balduzzi, D. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

Ghifary, M., Balduzzi, D., Kleijn, W. B., and Zhang, M. Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430, 2016.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pp. 2839–2848, 2016.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gulrajani, I. and Lopez-Paz, D. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Heinze-Deml, C. and Meinshausen, N. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2019.

Hu, S., Zhang, K., Chen, Z., and Chan, L. Domain generalization via multidomain discriminant analysis. In *Uncertainty in artificial intelligence: proceedings of the... conference. Conference on Uncertainty in Artificial Intelligence*, volume 35. NIH Public Access, 2019.

Huang, Z., Wang, H., Xing, E. P., and Huang, D. Self-challenging improves cross-domain generalization. *arXiv preprint arXiv:2007.02454*, 2, 2020.

Ilse, M., Tomczak, J. M., Louizos, C., and Welling, M. Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*, pp. 322–348. PMLR, 2020.

Irvin, J., Rajpurkar, P., Ko, M., Yu, Y., Ciurea-Ilcus, S., Chute, C., Marklund, H., Haghgoo, B., Ball, R., Shpanskaya, K., et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 590–597, 2019.

Johansson, F. D., Sontag, D., and Ranganath, R. Support and invertibility in domain-invariant representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 527–536, 2019.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Priol, R. L., and Courville, A. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

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

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

Li, D., Zhang, J., Yang, Y., Liu, C., Song, Y.-Z., and Hospedales, T. M. Episodic training for domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1446–1455, 2019a.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Sequential learning for domain generalization. *arXiv preprint arXiv:2004.01377*, 2020.

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

Li, Y., Gong, M., Tian, X., Liu, T., and Tao, D. Domain generalization via conditional invariant representations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018c.

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

Li, Y., Yang, Y., Zhou, W., and Hospedales, T. M. Feature-critic networks for heterogeneous domain generalization. *arXiv preprint arXiv:1901.11448*, 2019b.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., and Mooij, J. M. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, pp. 10846–10856, 2018.

Matsuura, T. and Harada, T. Domain generalization using a mixture of multiple latent domains. In *AAAI*, pp. 11749–11756, 2020.

Miller, J. W., Goodman, R., and Smyth, P. On loss functions which minimize to conditional expected values and posterior probabilities. *IEEE Transactions on Information Theory*, 39(4):1404–1408, 1993.

Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5715–5725, 2017.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18, 2013.

Nam, H., Lee, H., Park, J., Yoon, W., and Yoo, D. Reducing domain gap via style-agnostic networks. *arXiv preprint arXiv:1910.11645*, 2019.

Pearl, J. *Causality*. Cambridge university press, 2009.

Peters, J., Bühlmann, P., and Meinshausen, N. 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.

Piratla, V., Netrapalli, P., and Sarawagi, S. Efficient domain generalization via common-specific low-rank decomposition. *Proceedings of the International Conference of Machine Learning (ICML) 2020*, 2020.

Rahman, M. M., Fookes, C., Baktashmotlagh, M., and Sridharan, S. Correlation-aware adversarial domain adaptation and generalization. *Pattern Recognition*, 100:107124, 2020.

Rojas-Carulla, M., Schölkopf, B., Turner, R., and Peters, J. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

Shankar, S., Piratla, V., Chakrabarti, S., Chaudhuri, S., Jyothi, P., and Sarawagi, S. Generalizing across domains via cross-gradient training. In *International Conference on Learning Representations*, 2018.

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

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

Wang, H., He, Z., Lipton, Z. C., and Xing, E. P. Learning robust representations by projecting superficial statistics out. *arXiv preprint arXiv:1903.06256*, 2019.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

Wang, Y., Li, H., and Kot, A. C. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3622–3626. IEEE, 2020.

Xu, M., Zhang, J., Ni, B., Li, T., Wang, C., Tian, Q., and Zhang, W. Adversarial domain adaptation with domain mixup. *arXiv preprint arXiv:1912.01805*, 2019.

Yan, S., Song, H., Li, N., Zou, L., and Ren, L. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

Zhao, H., Combes, R. T. d., Zhang, K., and Gordon, G. J. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.

Zhao, S., Gong, M., Liu, T., Fu, H., and Tao, D. Domain generalization via entropy regularization. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhou, K., Yang, Y., Hospedales, T. M., and Xiang, T. Deep domain-adversarial image generation for domain generalisation. In *AAAI*, pp. 13025–13032, 2020.