iDAG: Invariant DAG Searching for Domain Generalization

Zenan Huang Haobo Wang Junbo Zhao Nenggan Zheng*
Zhejiang University
{lccurious,wanghaobo,j.zhao,zng}@zju.edu.cn

Abstract

Existing machine learning (ML) models are often fragile in open environments because the data distribution frequently shifts. To address this problem, domain generalization (DG) aims to explore underlying invariant patterns for stable prediction across domains. In this work, we first characterize that this failure of conventional ML models in DG attributes to an inadequate identification of causal structures. We further propose a novel invariant Directed Acyclic Graph (dubbed iDAG) searching framework that attains an invariant graphical relation as the proxy to the causality structure from the intrinsic data-generating process. To enable tractable computation, iDAG solves a constrained optimization objective built on a set of representative class-conditional prototypes. Additionally, we integrate a hierarchical contrastive learning module, which poses a strong effect of clustering, for enhanced prototypes as well as stabler prediction. Extensive experiments on the synthetic and real-world benchmarks demonstrate that iDAG outperforms the state-of-the-art approaches, verifying the superiority of causal structure identification for DG. The code of iDAG is available at `https://github.com/lccurious/iDAG`.

1. Introduction

An imperative goal of deep learning is to learn representations that faithfully represent task-oriented semantics and also generalize to different domains. It is known, however, that the performance of current models trained by the Empirical Risk Minimization (ERM) paradigm relies heavily on the *i.i.d.* assumptions and suffers a dramatic performance drop when inferring on Out-Of-Distribution (OOD) datasets [50]. However, when we deploy our models in the real world, we have little control over the distribution we observe; for instance, variables may change in frequency or new feature combinations may emerge that were not included in the training set [21]. In response to this challenge, Domain Generalization (DG) is proposed to improve performance in OOD inference by learning invariant features over the source domains that are generalizable to distributions different from those observed during training [5, 35].

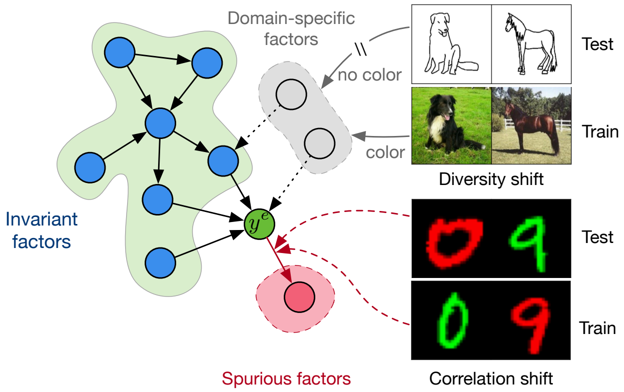

Figure 1. Multiple causally related factors may be present in visual data, and these factors are structurally organized together and present a higher level of semantics. However, due to domain diversity, the learned color factors in one domain can be useless in others; the spurious factors that dominate classification in one domain can be misleading in others. iDAG seeks to estimate the DAG which represents the directed causal relations between factors.

The most substantial challenge of DG is the spurious correlations may mislead the models to cheating classify the easier-to-fit features rather than true attributes [3]. As we exemplified in Figure 1, the reasons for such cheating are two-fold. First, some discriminative features appeared in training domains, but may disappear in test domains, which is known as *diveristy shift*. Second, there exists *correlation shift* that induces spurious features for predictions, *e.g.*, the background of an image can dominate the classification during training. To cope with these problems, a plethora of methods have been proposed to learn domain invariant features, including content-style disentanglement [61], constructing auxiliary task as penalty [7], force risks invariant cross-domains [3, 30]. However, most of them focus on relations from features to labels, and a unified consideration of modeling the global relationship between features and semantics remains underexplored.

*Corresponding author

With further scrutinizing the key causes above, we find that both challenges stem from the wrong identification of causal relations. Indeed, the most substantial concept for achieving DG [43, 41] is eliciting the causal structure across domains that invariantly controls the data-generating process. To this end, we propose to reformulate DG to a novel *invariant causal graph searching problem*. Once obtaining the directed acyclic graph (DAG) relations between latent factors, both the spurious correlation and diversity shift can be naturally recognized and removed by estimation of a global picture of causal graph among factors and label (see Figure 1). Despite the promise, it is non-trivial to identify the DAGs since the feature distribution constantly changes over the course of training. To date, few efforts have been made to resolve this.

In this study, we investigate a novel domain generalization framework by learning invariant Directed Acyclic Graph-structured feature relations (dubbed **iDAG**), which discovers the intrinsic feature organizations for stable label prediction. The core of iDAG is a constrained optimization problem that minimizes the trace of the adjacent matrix exponential, which guarantees the acyclicity and eliminates the spurious relations simultaneously. To prevent traveling the whole dataset in every training step, we design a prototype-based dataset proxy technique for efficient and stable optimization. Furthermore, iDAG incorporates a hierarchical contrastive learning module that aligns the latent causal factors to improve the representativeness of the prototypes and achieve stabler prediction. Comprehensive experiments show that iDAG accomplishes the *state-of-the-art* performance on various benchmark datasets, and is able to mitigate both diversity and correlation shifts in a unified framework.

2. Related Works

Domain Generalization. The goal of domain/out-of-distribution generalization is to explore the invariant pattern from multi-domain data to mitigate potential domain shifts when testing. To tackle this problem, a popular line of DG algorithms [1, 2, 26, 57] resorts to extract domain-invariant features from backbone, including adversarial learning-based algorithms [1, 2, 14, 26, 57], mixup-based methods [34, 57], data augmentation [42], feature alignment [38, 48], gradient alignment [39, 44], invariant risk minimization [3, 30], prototypical learning [13]. Another line of DG algorithms follows the idea of the style and content disentanglement [49, 17, 61]. Inspired by self-supervised learning, some methods construct auxiliary tasks for improving model generalization, such as solving jigsaw puzzles [7], self-challenging [20]. Recently, Ye *et al.* [59] indicate that there can be two forms of domain shifts, *i.e.*, *diversity shift* and the *correlation shift* in the DG problems, but most existing works tackle only one of them. In our work, we show the two shifts can be mitigated in a unified framework by reformulating the DG to an invariant causal structure searching problem.

Causality. Without prior knowledge of data-generating processes, conventional ML models tend to rely upon spurious associations in the training data for prediction. To address this issue, structural causal models (SCMs) [37, 3] have attracted great attention due to their specification of invariance under different environments. Given a set of causal factors, constrained-based methods [45, 46, 62] learn DAG that represents SCM by applying conditional independent tests to all predefined variables in the dataset; score-based methods [11, 15, 60] learn DAG by optimizing a certain score function. Recently, several studies [16, 33, 47] attempts to model conditional independence between features and labels as analogous to causal relations in domain adaptation [16, 33, 47, 63]. To date, few efforts have been made to resolve the standard causal graphs in DG. iCaRL [31] applies a post-hoc pruning strategy on fixed features, but it requires an invertible VAE architecture and the performance is far away from practical utilization. In contrast, our work discovers causal structures from an online graph-searching perspective that offers superior end-to-end performance.

Contrastive Learning (CL). CL [51, 19] is a representation learning framework by exploiting and enhancing the instance similarity and dissimilarity. It has shown promising results in many research fields like unsupervised learning [51, 19], weakly-supervised learning [54, 27], disentanglement learning [69]. A few works [67, 65] also employ CL to alleviate the domain shift problem as its superiority in feature alignment. Different from these studies, our CL module exploits the clustering effect of CL [54] to enhance the representativeness of prototypes as well as promote the invariant DAG identification.

3. Notation and Preliminary

We consider a domain generalization problem with E labeled domains $\{\mathcal{D}_L^e\}_{e=1}^E$ which together construct entire labeled training dataset $\mathcal{D}_{\text{tr}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Let $\mathcal{X}$ be the sample space, and $\mathcal{Y}$ be the label space. Our goal is to train a model $f : \mathcal{X} \mapsto \mathcal{Y}$ on $\mathcal{D}_{\text{tr}}$ which gives a fairly good performance on the inaccessible dataset $\mathcal{D}_{\text{te}}$ during the training phase. Define $\mathcal{Z} \in \mathbb{R}^d$ as the feature space. We decompose the f as $f = \omega \circ \phi$ that indicate classifier $\omega : \mathcal{Z} \mapsto \mathcal{Y}$ and convolutional backbone $\phi : \mathcal{X} \mapsto \mathcal{Z}$ respectively; see Appendix A.1 for a more detailed notation table.

3.1. Investigating the Out-of-Distribution

In this section, we first demonstrate the challenges of DG by emphasizing *diversity shift* and *correlation shift* [59]. To begin with, it is necessary to define the latent factors that control the data-generating process.

Data-generating process. Here, we generalize the previous setting [3, 31] to show the failure of ordinary ERM on the OOD problem by revisiting the data-generating process. Without loss of generality, assume there is a group of injective data-generating functions $g_y(\cdot), g_s^e(\cdot), g_r^e(\cdot)$ for each environment, the underlying factors of observational data follow the rule:

$$y^e = g_y(\boldsymbol{z}_y^e) + \epsilon_y, \quad \boldsymbol{z}_s^e = g_s^e(y^e) + \boldsymbol{\epsilon}_s^e, \quad \boldsymbol{z}_r^e = \boldsymbol{\epsilon}_r^e, \tag{1}$$

where $\boldsymbol{z}_y$ denotes invariant features, $\boldsymbol{z}_s^e$ denotes the easier-to-fit spurious features, $\boldsymbol{z}_r^e$ denotes the domain-private features, $\epsilon_y, \boldsymbol{\epsilon}_s^e, \boldsymbol{\epsilon}_r^e$ are mutually independent exogenous noises. Models heavily rely on $\boldsymbol{z}_s^e$ and $\boldsymbol{z}_r^e$ may lower their empirical risk in a particular domain, but become extremely fragile to open environments. It is a well-known domain shift problem that can be formally attributed to two types of shifts [59], *i.e.*, *diversity shifts* and *correlation shifts*.

Causal Structure Matters in DG. Hence, an ideal model robust to open environments should discard the tendency of learning cheating rules using $\boldsymbol{z}_r^e$ and $\boldsymbol{z}_s^e$. However, this can be an almost impossible task when using an ordinary supervised paradigm that only considers inferring y^e based on $\boldsymbol{z}^e = \{\boldsymbol{z}_y, \boldsymbol{z}_s^e, \boldsymbol{z}_r^e\}$. An intuitive case would be that ordinary ERM can only detect strong associations between grass and cow $\boldsymbol{z}_s^e \leftrightarrow y^e$, but never learn the concept that because the images are of cows so there are a lot of grass backgrounds $y^e \rightarrow \boldsymbol{z}_s^e$. In other words, the notion of ordinary ERM can only struggle on reducing the total risks by balancing the weights on naive directions $\{\boldsymbol{z}_y, \boldsymbol{z}_s^e, \boldsymbol{z}_r^e\} \rightarrow y^e$.

The above example illustrates a typical problem in DG, which is that fragile models are learned on the basis of incorrect causal relationships. As a consequence of causal language, a relation between factors is assigned in a particular direction. In a nutshell, our ultimate goal is to identify a graphical structure that reflects the spurious relations $y^e \rightarrow \boldsymbol{z}_s^e$ and prunes unstable relations $\boldsymbol{z}_r^e \rightarrow y^e$. Therefore, the domain invariant features $\phi(\boldsymbol{x}) \approx \boldsymbol{z}_y^e$ can be identified for stable learning.

4. Method

In this section, we describe our domain generalization by learning invariant DAG-structured feature/label relations. In a nutshell, iDAG comprises two key components. First, we present a DAG-based feature modeling framework, the optimal DAG naturally induces the invariant features for stable prediction (Section 4.1) and contrastive prototype update respectively (Section 4.4). These two components work in a reciprocal manner.

4.1. Stable DAG for Domains

To perform causal discovery, our iDAG framework resorts to manipulating the causal relations between the union set of latent factors and labels. For the sake of notational simplicity, we will use v_i to uniformly represent feature element z_i and label y. Thereafter, we can define a structural causal model (SCM) on the whole collection of causal factors $\mathcal{V} = \{v_i\}_{i=1}^d = \{z_i\}_{i=1}^d \cup \{y\}$.

Definition 1. *(Domain Invariant DAG). A domain-specific SCM $\mathcal{M}^e$ on a set of nodes $\mathcal{V}^e$ with joint distribution $p(\boldsymbol{v}^e)$, according to Markov condition, it can be factorized by:*

$$p_{\mathcal{M}^e}(\boldsymbol{v}^e) = \prod_i p_{\mathcal{M}^e}(v_i^e | \mathbf{Pa}_i^e), \tag{2}$$

where $\mathbf{Pa}_i^e$ indicates the set of parents (its direct causes*) for v_i^e in domain e. Each $\mathcal{M}^e$ specify a graphical representation (DAG) $\mathcal{G}^e = (\mathcal{V}^e, \mathcal{E}^e)$, where $\mathcal{E}^e = \{(v_i^e, v_j^e)\}$ concludes the causal edges such as $v_i^e \rightarrow v_j^e$. So the i-th factor v_i can be directly inferred by $v_i = g_i(\boldsymbol{v})$ leverage a generation function g_i. An invariant DAG $\mathcal{G}$ is defined as the common structure of all $\{\mathcal{G}^e\}_{e=1}^E$ across all domains.*

Theorem 1. *If $\mathcal{G}$ matches the common structures of all $\mathcal{G}^e$, then it discards the directed edges that start from domain-private factors $\boldsymbol{v}_r^e$ and identifies the association $v_i \leftrightarrow v_j$ into one correct causal direction.*

The proof can be found in Appendix C. Theorem 1 indicates that spurious relations can be identified by causal structure learning, and domain-private factors dependences can be eliminated via graph consolidation. Distinct from previous works [68, 3] simply finding the invariant factors, this framework brings two main advantages. First, the direct/indirect causes of y^e correspond to its ancestors on the DAG, indicating that invariant factors can be found by tracing back from y^e. Second, the strict acyclic constraint naturally avoids the spurious correlations $\boldsymbol{z}_s^e \rightarrow y^e$ while only maintaining $y^e \rightarrow \boldsymbol{z}_s^e$ when the global optimum of the invariant DAG is achieved. In other words, the invariant DAG identification procedure offers new possibilities for resolving the key challenges of DG.

4.2. Searching DAG from Features and Labels

To learn invariant DAG, an intuitive strategy is to learn DAGs for every domain and take their shared subgraph. In light of the fact that the same data can induce multiple valid DAGs, a.k.a. Markov equivalence class [32, 10], which are a set of graphs that satisfy the same conditional independence relations, it may prove difficult to extract the subgraph under these circumstances. In our iDAG method, we solve this problem by penalizing a single domain-invariant graph that explains causal relationships across all domains.

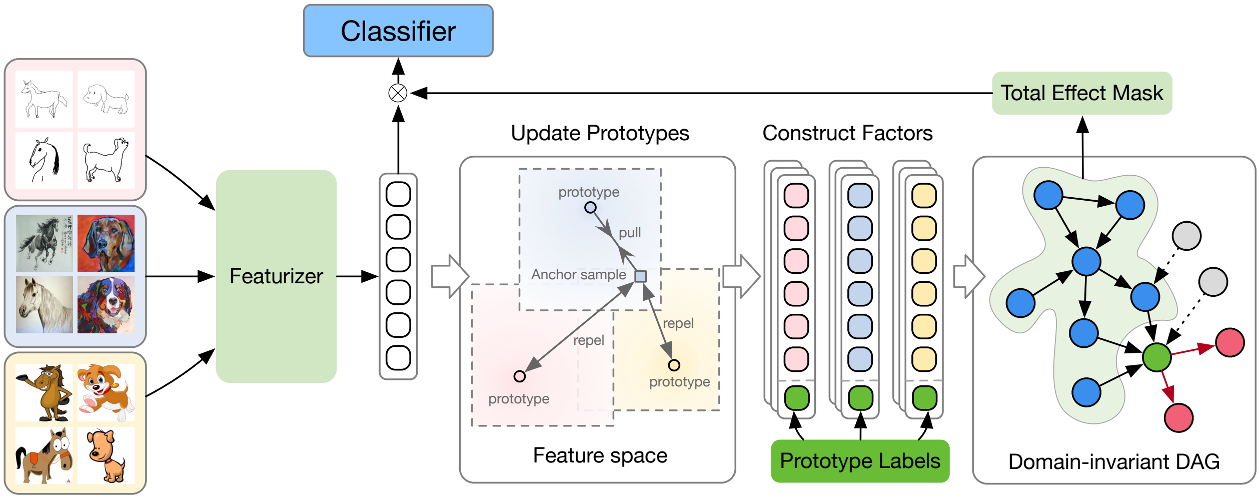

Figure 2. Illustration of iDAG. The features are used to update the domain-specific prototypes. The prototypes concatenated with labels are then used to optimize a Directed Acyclic Graph. The shaded green regime on DAG indicates the factors have total effects on label y, and the invariant features corresponding to these factors are used for final prediction.

In each training step, we would like to search for an invariant DAG represented by a learnable adjacent matrix $\boldsymbol{A} \in \mathbb{R}^{d+1\times d+1}$ from the currently learned features and labels. To achieve this, we draw inspiration from the graph learning literature [66, 60] by causal factor reconstruction. According to the data-generating process in Defintion 1, an arbitrary factor v_i can be inferred based on the entire factors set $\{v_i\}_{i=1}^{d+1}$. Let $\boldsymbol{v}$ be the concatenation $\boldsymbol{v} = [\boldsymbol{z}, y]$. For each factor, we first adopt a row $\boldsymbol{A}_i$ for masking out the non-parent elements and then map the parents to i-th factor by g_i. The reconstruction process is instantiated as follows,

$$g_i(\cdot) := \begin{cases} \boldsymbol{A}_i\boldsymbol{v} & \text{for numerical } v_i, \\ \boldsymbol{W}(\boldsymbol{A}_i^\top \odot \boldsymbol{v}) & \text{for categorical } v_i, \end{cases} \quad (3)$$

where $\odot$ denotes the element-wise product and $\boldsymbol{A}_i$ indicates the assignment of parents to child. Here $\boldsymbol{W} \in \mathbb{R}^{C\times d}$ is a weight matrix that maps the parents to the categorical logits, which is particularly designed for classification tasks. In practice, we can simply apply a unified numerical mapping function for regression tasks. But, since we can also discretize regression problems, our following discussion concentrates on the mixed-learning mode where the features are numerical and the labels are categorical.

Now, we can write down the overall score function according to Eq. (3). For general bottleneck feature element z_i, we use $L^2(\cdot,\cdot)$ norm metric as the score function for them all, but for the categorical label variable y, the cross-entropy loss $\ell_{ce}(\cdot,\cdot)$ is used. Put them together, the complete graph reconstruction loss is:

$$\mathcal{L}_{\text{rec}} = \sum_{i=1}^{d} L^2\left(g_i(\boldsymbol{z}, y), z_i\right) + \ell_{ce}\left(g_y(\boldsymbol{z}), y\right). \quad (4)$$

Yet, the graph is still not ensured to be acyclic. When there exist easier-to-fit spurious relations, the least squares style objective Eq. (4) tends to introduce the cycles in the estimated graph (see Appendix for more motivative analysis). To eliminate spurious relationships, we introduce the exponential trace constraint [66] for $\boldsymbol{A}$ to guarantee the acyclic property of DAGs,

$$\boldsymbol{A} \text{ is a DAG} \Leftrightarrow h(\boldsymbol{A}) = \text{Tr}(e^{\boldsymbol{A}\odot\boldsymbol{A}}) - (d+1) = 0. \quad (5)$$

In effect, this regularizer restricts a node from being not able to reach itself even after infinite steps in this directed graph. The acyclic constraints and the fact additive noise [37] only appeared in the true causal direction jointly ensuring the correct causal direction inference. With the correct causal directions between factors and labels acknowledged, we can identify spurious relationships and extract invariant features, thereby mitigating the correlation shift. The final graph learning objective is given by,

$$\mathcal{L}_{\mathcal{G}} = \mathcal{L}_{\text{rec}} + \lambda_{l1}|\text{vec}(\boldsymbol{A})|_1 \;\; s.t. \;\; h(\boldsymbol{A}) = 0, \quad (6)$$

where $\text{vec}(\cdot)$ is a vectorize operator for matrix, λ_{l1} is a weight parameter for enforcing the sparsity of the DAG. Note the $\mathcal{L}_{\text{rec}}$ corresponding to reconstruction loss which is flexible to use other distance functions.

With further scrutiny of the above objective function, this formulation complies with our ultimate goals: 1) for spurious features, the edge $y^e \rightarrow \boldsymbol{z}_s^e$ will be learned prior to its reversed version to resist mutually independent noise $\boldsymbol{\epsilon}_s^e$ (hence lower loss), and the acyclic constraint further discards the reverse edge completely; 2) for invariant features $\boldsymbol{z}_y$, the relations $\boldsymbol{z}_y \rightarrow y^e$ will be first learned for the same reason; 3) lastly, those domain-private features have less relations to other factors across domains, and thus no edge will not be assigned. Notably, we can draw the above arguments in a more rigorous way by the following theorem.

Theorem 2. *Under the assumption of the data-generating process with E environments and the Theorem 1 holds, if $E * |\mathcal{D}_L^e| > Q_1 + Q_2 \ln(d/\delta)$, then with probability at least $1-\delta$ the following inequality holds:*

$$\hat{\mathcal{L}}(\boldsymbol{A}_{\text{inv}}) < \hat{\mathcal{L}}(\boldsymbol{A}), \forall \boldsymbol{A} \neq \boldsymbol{A}_{\text{inv}} \wedge h(\boldsymbol{A}) = 0, \tag{7}$$

where Q_1 and Q_2 are constants specified in the appendix C.

The proof can be found in Appendix C. Theorem 2 indicates that under mild assumptions, iDAG can asymptotically find a more wildly reliable invariant causal graph $\boldsymbol{A}_{\text{inv}}$ as the number of environments increases. In conclusion, our learned DAG is ensured to be invariant for DG and also theoretically feasible.

Tractable and efficient optimization. To optimize our objective, while theoretically feasible, it is required to travel the entire training dataset to extract the latent factors of each example for optimization. Empirically, we find that such a solution is not only computationally expensive but is unstable since one mini-batch may not cover all the domains and categories. To this end, we devise a novel prototype-based algorithm for efficient optimization. For each domain e, we maintain C class-conditional prototypes $\mathcal{B}_\nu^e = \{\boldsymbol{\nu}_c^e\}_{c=1}^C$ embeddings that condense the dataset to several representative vectors.

In our implementation, we optimize the graph after each iteration of model training, while freezing all model parameters. Then, the augmented Lagrangian algorithm is used to resolve the constrained optimization problem in Eq. (6). After that, we follow [60] apply the alternative augmented Lagrangian optimization and employ the new features to update the prototype in a moving average style,

$$\boldsymbol{\nu}_c^e = \text{Normalize}(\gamma \boldsymbol{\nu}_c^e + (1-\gamma)\boldsymbol{z}_c^e), \tag{8}$$

where $\gamma \in (0,1)$ is a scalar controls the momentum of updating prototypes, $\boldsymbol{z} = \phi(\boldsymbol{x})$ is bottleneck feature, $\boldsymbol{z}_c^e$ is the features that corresponding to the label c and domain e. This moving-average updating procedure enables the prototypes to be relatively stable over the course of training, which also results in the stable calculation of the DAG.

4.3. Stable Prediction based on DAG

To extract the invariant features for stable prediction, a brute-force approach is adopting the same strategy of Eq. (3) which predicts y based on the direct causal parents $\mathbf{Pa}_y$. Nevertheless, the ordinary $\boldsymbol{z}^e$ general suffers from noise, and predictions may be too sensitive when based solely on parental factors. To obtain a more stable prediction, it is required to collect not only the parent factors but also the ancestral factors of y in the DAG. Thus, we define invariant features by including all direct and indirect causal factors. Follow the idea of the fact that the positivity of the (i,j) element of the k-th power of $\boldsymbol{A}$ indicates the existence of a length-k path $v_i \to \cdots \to v_j$, we derive the $\boldsymbol{P}^{\text{tol}}$ to analogy the directed pairwise total effects,

$$\boldsymbol{P}^{\text{tol}} = \left[\sum_{k=0}^{\infty} \frac{1}{k!}(\boldsymbol{A} \odot \boldsymbol{A})^k\right] = e^{\boldsymbol{A} \odot \boldsymbol{A}}, \tag{9}$$

where $P_{i,j}^{\text{tol}}$ analogy the total causal effect $v_i \to v_j$. Then, the invariant features $\boldsymbol{z}_y^e := \boldsymbol{z}^e \odot [\boldsymbol{P}^{\text{tol}}]_{d+1,1:d}^\top \in \mathbb{R}^d$ contains all the direct and indirect causal features of y. Finally, we optimize the stable classifier based on invariant features,

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{tr}}}\left[\ell_{ce}(\omega(\phi(\boldsymbol{x}) \odot [\boldsymbol{P}^{\text{tol}}]_{d+1,1:d}^\top), y)\right]. \tag{10}$$

4.4. Enhanced Prototypes by Contrastive Learning

In our iDAG framework, it is crucial to guarantee the representativeness of the prototypes. To achieve this, we additionally incorporate a hierarchical prototypical contrastive learning (PCL) module [28] for enhanced prototypes.

Contrastive loss. To begin with, we first present the classical formulation of prototypical contrastive learning. Given an anchor sample, the PCL attempts to optimize the following objective,

$$\mathcal{L}_{\text{CL}} = -\frac{1}{|\mathcal{P}(\boldsymbol{z})|}\sum_{\boldsymbol{k}_+ \in \mathcal{P}(\boldsymbol{z})} \log \frac{\exp(\boldsymbol{z}^\top \boldsymbol{k}_+/\tau)}{\sum_{\boldsymbol{k}' \in \mathcal{B}(\boldsymbol{z})} \exp(\boldsymbol{z}^\top \boldsymbol{k}'/\tau)}, \tag{11}$$

where $\tau \geq 0$ is the tunable temperature. We attempt to pull an anchor sample to its positive prototype set and repel it from all other samples and prototypes. Thus, the key step of PCL is to define the positive set $\mathcal{P}(\boldsymbol{z})$ and the complete set $\mathcal{B}(\boldsymbol{z})$. In our work, we employ the PCL algorithm for achieving two goals: (i)-improving the representativeness of in-domain prototypes; (ii)-promoting the identification of invariant DAG across domains.

In-domain PCL. First, we collect the features of each sample and the per-domain prototypes to construct the following in-domain embedding pool:

$$\mathcal{B}^e = \mathcal{B}_\nu^e \cup \mathcal{B}_z,\ \mathcal{B}_\nu^e = \{\boldsymbol{\nu}_c^e\}_{c=1}^C,\ \mathcal{B}_z = \{\boldsymbol{z}_i\}_{i=1}^B, \tag{12}$$

where $\mathcal{B}_\nu$ contains prototypes of all environments and classes, and $\mathcal{B}_z$ contains the domain-specific features from current mini-batch with size B. Then corresponding positive set $\mathcal{P}(\boldsymbol{z}^e)$ is defined by:

$$\mathcal{P}(\boldsymbol{z}^e) = \{\boldsymbol{k}' | \boldsymbol{k}' \in \mathcal{B}_\nu^e, e' = e \wedge c' = c\}. \tag{13}$$

Then, we follow Eq. (11) and calculate the in-domain contrastive loss $\mathcal{L}_{\text{CL-}\nu}$. It is worth noting that contrastive learning is known to pose a clustering effect in the embedding space [54]. Thus, the samples can be tightly aligned to their prototypes, making them more representative.

Cross-domain PCL. Recall that our ultimate goal of DG is to learn an invariant DAG from the prototypes. In effect, it suggests that the invariant factors of all the samples are requested to be aligned in a shared space. This motivates us to develop a novel PCL loss to directly enhance this property. Specifically, we extract the invariant features of each sample by total effect masking as in Eq. (10). Then, we construct the following cross-domain embedding pool:

$$\bar{\mathcal{B}} = \mathcal{B}_\mu \cup \mathcal{B}_y,\ \mathcal{B}_\mu = \{\boldsymbol{\mu}_c\}_{c=1}^{C}, \mathcal{B}_y = \{\boldsymbol{z}_{y,i}\}_{i=1}^{B}, \quad (14)$$

where $\mathcal{B}_\mu$ contains prototypes of all classes, $\mathcal{B}_y$ contains invariant features from current mini-batch with size B. Accordingly, the positive set $\mathcal{P}(\boldsymbol{z}_y)$ is defined by:

$$\mathcal{P}(\boldsymbol{z}_y) = \{\boldsymbol{k}' | \boldsymbol{k}' \in \mathcal{B}_\mu, c' = c\}. \quad (15)$$

Similarly, we calculate the cross-domain contrastive loss $\mathcal{L}_{\text{CL-}\mu}$. The effect of intraclass concentration, therefore, contributes to the learning of the invariant subgraph.

Overall Objective. Finally, we aggregate all the losses to our overall objective,

$$\min_{\boldsymbol{A},\boldsymbol{\theta}} \mathcal{L}_{\text{cls}} + \mathcal{L}_{\mathcal{G}} + \mathcal{L}_{\text{CL}-\nu} + \mathcal{L}_{\text{CL}-\mu}, \quad (16)$$

where $\boldsymbol{\theta}$ is the collection of parameters of f and $\{g_i\}$. In each iteration, we iteratively update the DAG $\boldsymbol{A}$ and $\boldsymbol{\theta}$ by freezing the other one until convergence. The pseudo-code of our complete algorithm is shown in Appendix F.

5. Experiments

In this section, we demonstrate the effectiveness of iDAG on both synthetic datasets for clear illustration and four vision datasets for empirical evaluation. More details and empirical results can be found in Appedix D.

5.1. Setup

Dataset. We consider the following vision classification datasets: CMNIST [3] (50000 images, 2 classes, and 3 domains, 25% label noise), PACS [23] (9,991 images, 7 classes, and 4 domains), OfficeHome [53] (15,588 images, 65 classes, and 4 domains), and DomainNet [38] (586,575 images, 345 classes, and 6 domains) to validate the iDAG against previous methods.

Table 1. Test MSE on the synthetic dataset. The sample size stands from the amount of training data.

Sample size	5K	2K	1K	0.5K
Oracle	0.97	0.98	1.02	1.02
ERM [52]	28.40	27.22	30.32	28.66
IRMv1 [3]	2.15	4.31	8.76	13.75
REx [22]	5.55	8.65	15.4	15.12
InvRat [9]	2.25	4.15	9.03	13.66
BIRM [30]	1.82	2.90	3.17	3.86
iDAG	**1.01**	**0.98**	**1.05**	**1.10**
Weights (R^2)	0.99	0.99	0.99	0.99

Metrics. Following the commonly used leave-one-domain-out protocol [18], we specify one domain as the unseen target domain for evaluation and train with the remaining domains. For a fair comparison, we follow the evaluation protocol in DomainBed [18], splitting each source domain with 80% for training and 20% for validation. The final model is used for testing on the unseen target domain and reporting the accuracy with mean and standard deviation based on 3 independent runs.

Baselines. For conventional datasets such as PACS, OfficeHome, and DomainNet, we compare our method with ERM [52], IRM [3], ARM [64], RSC [20], CDANN [29], DRO [40], MMD [26], MTL [4], MLDG [24], Mixup [57, 58, 55], SagNet [36], CORAL [48], mDSDI [6], SWAD [8], and DNA [12]. For extremely spurious datasets such as CMNIST, we compare our method with IRM [3], and its variants DILU [56], InvRat [9], and BIRM [30].

Implementation. For CMNIST dataset, we construct the backbone network with MLPs following previous works [3, 30] (detailed in Appendix E.2). For conventional DG datasets, PACS, OfficeHome, and DomainNet, we use ImageNet pre-trained ResNet-50 as backbone network and build experiments following SWAD [8] and warm the model up by running standard ERM. All the batch normalization (BN) layers are frozen during training. We replace the last FC layer of the backbone with three-layer encoder networks with account for 256 hidden units for PACS, 512 hidden units for OfficeHome, and 1024 for DomainNet. To be consistent with the existing line of work [59, 18], we conduct the hyperparameter (HP), and model selection on the validation set for the benchmarks on PACS, OfficeHome, and DomainNet; for the models trained on CMNIST are selected by test-domain validation.

5.2. Synthetic examples

Our first series of experiments are conducted on a synthetic dataset that demonstrates when the features are stable, iDAG is capable of solving the problem of the spurious relation ideally. The synthetic dataset considers a similar case with IRM paper [3], where the spurious feature is induced by the anti-causal effect. Specifically, the dataset is generated as Eq. (1), where $\boldsymbol{\epsilon}_s^e$ varies in different environments, by setting variance of $\boldsymbol{\epsilon}_s^e = \{0.5, 1.0, 9.9\}$ we create three environments, and we only use the first two environments for training. Under this setting, $\boldsymbol{z}_s^e \to y^e$ will be a strong spurious relation that general models are confused with during training. And a model that relies on this spurious relation would perform poorly in the testing dataset in which $\boldsymbol{z}_s^e$ are less associated with y^e. We fit a linear variant of the iDAG model with generated features $\boldsymbol{z}_y^e, \boldsymbol{z}_s^e$ and y^e. Then we evaluate the Mean Squared Error (MSE) between the predicted value $\hat{y}$ and y: $\mathbb{E}[(y - \hat{y})^2]$.

Simulation results. Table 1 shows the results of each method with different amounts of training data. The poor performance of ERM indicates it relies on easier-to-learn spurious relations. And the deterioration of performances of other baselines somehow relies on the data amounts for distribution learning, even the identifiable features are given. Compared to the baselines, iDAG shows really stable performances under a wide range of training data amounts. And the iDAG also shows its relatively close to the oracle, which somehow means the concept of DAG modeling greatly resolved this spurious correlations problem. Further, we also validate the R^2 coefficients between the estimated classifier parameters and the true parameters, the results indicate iDAG can successfully recover the correct relations.

5.3. Main results

Comparison on conventional domain generalization benchmarks. We report the full out-of-domain performances on Table 2, 3, and 4. Comprehensive experiments show that iDAG consistently outperforms the baselines both in most single domains and on average. Corresponding to the domain shift quantification in two dimensions [59], PACS and OfficeHome contain diversity shift, and DomainNet contains both diversity and correlation shift. The results indicate that iDAG is more effective in tackling both of these two kinds of shifts.

Table 2. Comparison with state-of-the-art methods on PACS benchmark with ResNet-50 ImageNet pre-trained model.

Method	A	C	P	S	Avg
CDANN [25]	84.6$_{\pm 1.8}$	75.5$_{\pm 0.9}$	96.8$_{\pm 0.3}$	73.5$_{\pm 0.6}$	82.6
IRM [3]	84.8$_{\pm 1.3}$	76.4$_{\pm 1.1}$	96.7$_{\pm 0.6}$	76.1$_{\pm 1.0}$	83.5
DANN [14]	86.4$_{\pm 0.8}$	77.4$_{\pm 0.8}$	97.3$_{\pm 0.4}$	73.5$_{\pm 2.3}$	83.6
DRO [40]	83.5$_{\pm 0.9}$	79.1$_{\pm 0.6}$	96.7$_{\pm 0.3}$	78.3$_{\pm 2.0}$	84.4
Mixup [57]	86.1$_{\pm 0.5}$	78.9$_{\pm 0.8}$	97.6$_{\pm 0.1}$	75.8$_{\pm 1.8}$	84.6
MMD [26]	86.1$_{\pm 1.4}$	79.4$_{\pm 0.9}$	96.6$_{\pm 0.2}$	76.5$_{\pm 0.5}$	84.6
MTL [4]	87.5$_{\pm 0.8}$	77.1$_{\pm 0.5}$	96.4$_{\pm 0.8}$	77.3$_{\pm 1.8}$	84.6
MLDG [24]	85.5$_{\pm 1.4}$	80.1$_{\pm 1.7}$	97.4$_{\pm 0.3}$	76.6$_{\pm 1.1}$	84.9
VREx [22]	86.0$_{\pm 1.6}$	79.1$_{\pm 0.6}$	96.9$_{\pm 0.5}$	77.7$_{\pm 1.7}$	84.9
ARM [64]	86.8$_{\pm 0.6}$	76.8$_{\pm 0.5}$	97.4$_{\pm 0.3}$	79.3$_{\pm 1.2}$	85.1
RSC [20]	85.4$_{\pm 0.8}$	79.7$_{\pm 1.8}$	97.6$_{\pm 0.3}$	78.2$_{\pm 1.2}$	85.2
ERM [52]	84.7$_{\pm 0.4}$	80.8$_{\pm 0.6}$	97.2$_{\pm 0.3}$	79.3$_{\pm 1.0}$	85.5
CORAL [48]	88.3$_{\pm 0.2}$	80.0$_{\pm 0.5}$	97.5$_{\pm 0.3}$	78.8$_{\pm 1.3}$	86.2
mDSDI [6]	87.7$_{\pm 0.4}$	80.4$_{\pm 0.7}$	**98.1**$_{\pm 0.3}$	78.4$_{\pm 1.2}$	86.2
SagNet [36]	87.4$_{\pm 1.0}$	80.7$_{\pm 0.6}$	97.1$_{\pm 0.1}$	80.0$_{\pm 0.4}$	86.3
SWAD [8]	89.3$_{\pm 0.2}$	83.4$_{\pm 0.6}$	97.3$_{\pm 0.3}$	82.5$_{\pm 0.5}$	88.1
DNA [12]	89.8$_{\pm 0.2}$	83.4$_{\pm 0.4}$	97.7$_{\pm 0.1}$	82.6$_{\pm 0.2}$	88.4
iDAG	**90.8**$_{\pm 0.4}$	**83.7**$_{\pm 0.5}$	98.0$_{\pm 0.3}$	**82.7**$_{\pm 0.9}$	**88.8**

Table 3. Comparison with state-of-the-art methods on OfficeHome benchmark with ResNet-50 ImageNet pre-trained model.

Method	Ar	Cl	Pr	Rw	Avg
IRM [3]	58.9$_{\pm 2.3}$	52.2$_{\pm 1.6}$	72.1$_{\pm 2.9}$	74.0$_{\pm 2.5}$	64.3
ARM [64]	58.9$_{\pm 0.8}$	51.0$_{\pm 0.5}$	74.1$_{\pm 0.1}$	75.2$_{\pm 0.3}$	64.8
RSC [20]	60.7$_{\pm 1.4}$	51.4$_{\pm 0.3}$	74.8$_{\pm 1.1}$	75.1$_{\pm 1.3}$	65.5
CDANN [25]	61.5$_{\pm 1.4}$	50.4$_{\pm 2.4}$	74.4$_{\pm 0.9}$	76.6$_{\pm 0.8}$	65.8
DANN [14]	59.9$_{\pm 1.3}$	53.0$_{\pm 0.3}$	73.6$_{\pm 0.7}$	76.9$_{\pm 0.5}$	65.9
DRO [40]	60.4$_{\pm 0.7}$	52.7$_{\pm 1.0}$	75.0$_{\pm 0.7}$	76.0$_{\pm 0.7}$	66.0
MMD [26]	60.4$_{\pm 0.2}$	53.3$_{\pm 0.3}$	74.3$_{\pm 0.1}$	77.4$_{\pm 0.6}$	66.3
MTL [4]	61.5$_{\pm 0.7}$	52.4$_{\pm 0.6}$	74.9$_{\pm 0.4}$	76.8$_{\pm 0.4}$	66.4
VREx [22]	60.7$_{\pm 0.9}$	53.0$_{\pm 0.9}$	75.3$_{\pm 0.1}$	76.6$_{\pm 0.5}$	66.4
ERM [52]	61.3$_{\pm 0.7}$	52.4$_{\pm 0.3}$	75.8$_{\pm 0.1}$	76.6$_{\pm 0.3}$	66.5
MLDG [24]	61.5$_{\pm 0.9}$	53.2$_{\pm 0.6}$	75.0$_{\pm 1.2}$	77.5$_{\pm 0.4}$	66.8
Mixup [57]	62.4$_{\pm 0.8}$	54.8$_{\pm 0.6}$	76.9$_{\pm 0.3}$	78.3$_{\pm 0.2}$	68.1
SagNet [36]	63.4$_{\pm 0.2}$	54.8$_{\pm 0.4}$	75.8$_{\pm 0.4}$	78.3$_{\pm 0.3}$	68.1
CORAL [48]	65.3$_{\pm 0.4}$	54.4$_{\pm 0.5}$	76.5$_{\pm 0.1}$	78.4$_{\pm 0.5}$	68.7
mDSDI [6]	68.1$_{\pm 0.3}$	52.1$_{\pm 0.4}$	76.0$_{\pm 0.2}$	80.4$_{\pm 0.2}$	69.2
SWAD [8]	66.1$_{\pm 0.4}$	57.7$_{\pm 0.4}$	78.4$_{\pm 0.1}$	80.2$_{\pm 0.2}$	70.6
DNA [12]	67.7$_{\pm 0.2}$	57.7$_{\pm 0.3}$	78.9$_{\pm 0.2}$	80.5$_{\pm 0.2}$	71.2
iDAG	**68.2**$_{\pm 0.4}$	**57.9**$_{\pm 0.3}$	**79.7**$_{\pm 0.2}$	**81.4**$_{\pm 0.4}$	**71.8**

Comparison on extreme correlation shift benchmark. We report the results of CMNIST on Table 5 using '-90%' as the testing environment following [3, 30], the full results on other environments can be found in Appendix D. This dataset is constructed with extreme correlation shift, the spurious feature distributions (color) are totally flipped between train and test domains. Besides, additional 25% label noise further fortified the influence of spurious relations. As we can see that iDAG constantly outperforms the baseline methods. Moreover, iDAG consistently achieves superior results as the size of the training data decreases, while the baselines demonstrate a more significant performance drop.

5.4. Ablation study

Effect of acyclic penalty. To demonstrate the effectiveness of the proposed acyclic constraint, we compare it with the setting of removing the acyclic constraint. Specifically, we optimize the Eq. (6) without restricting $h(\boldsymbol{A}) = 0$ as

Table 4. Comparison with state-of-the-art methods on DomainNet benchmark with ResNet-50 ImageNet pre-trained model.

Method	clip	info	paint	quick	real	sketch	Avg
MMD [26]	32.1$_{\pm 13.3}$	11.0$_{\pm 4.6}$	26.8$_{\pm 11.3}$	8.7$_{\pm 2.1}$	32.7$_{\pm 13.8}$	28.9$_{\pm 11.9}$	23.4
DRO [40]	47.2$_{\pm 0.5}$	17.5$_{\pm 0.4}$	33.8$_{\pm 0.5}$	9.3$_{\pm 0.3}$	51.6$_{\pm 0.4}$	40.1$_{\pm 0.6}$	33.3
VREx [22]	47.3$_{\pm 3.5}$	16.0$_{\pm 1.5}$	35.8$_{\pm 4.6}$	10.9$_{\pm 0.3}$	49.6$_{\pm 4.9}$	42.0$_{\pm 3.0}$	33.6
IRM [3]	48.5$_{\pm 2.8}$	15.0$_{\pm 1.5}$	38.3$_{\pm 4.3}$	10.9$_{\pm 0.5}$	48.2$_{\pm 5.2}$	42.3$_{\pm 3.1}$	33.9
ARM [64]	49.7$_{\pm 0.3}$	16.3$_{\pm 0.5}$	40.9$_{\pm 1.1}$	9.4$_{\pm 0.1}$	53.4$_{\pm 0.4}$	43.5$_{\pm 0.4}$	35.5
DANN [14]	53.1$_{\pm 0.2}$	18.3$_{\pm 0.1}$	44.2$_{\pm 0.7}$	11.8$_{\pm 0.1}$	55.5$_{\pm 0.4}$	46.8$_{\pm 0.6}$	38.3
CDANN [25]	54.6$_{\pm 0.4}$	17.3$_{\pm 0.1}$	43.7$_{\pm 0.9}$	12.1$_{\pm 0.7}$	56.2$_{\pm 0.4}$	45.9$_{\pm 0.5}$	38.3
RSC [20]	55.0$_{\pm 1.2}$	18.3$_{\pm 0.5}$	44.4$_{\pm 0.6}$	12.2$_{\pm 0.2}$	55.7$_{\pm 0.7}$	47.8$_{\pm 0.9}$	38.9
Mixup [57]	55.7$_{\pm 0.3}$	18.5$_{\pm 0.5}$	44.3$_{\pm 0.5}$	12.5$_{\pm 0.4}$	55.8$_{\pm 0.3}$	48.2$_{\pm 0.5}$	39.2
SagNet [36]	57.7$_{\pm 0.3}$	19.0$_{\pm 0.2}$	45.3$_{\pm 0.3}$	12.7$_{\pm 0.5}$	58.1$_{\pm 0.5}$	48.8$_{\pm 0.2}$	40.3
MTL [4]	57.9$_{\pm 0.5}$	18.5$_{\pm 0.4}$	46.0$_{\pm 0.1}$	12.5$_{\pm 0.1}$	59.5$_{\pm 0.3}$	49.2$_{\pm 0.1}$	40.6
ERM [52]	58.1$_{\pm 0.3}$	18.8$_{\pm 0.3}$	46.7$_{\pm 0.3}$	12.2$_{\pm 0.4}$	59.6$_{\pm 0.1}$	49.8$_{\pm 0.4}$	40.9
MLDG [24]	59.1$_{\pm 0.2}$	19.1$_{\pm 0.3}$	45.8$_{\pm 0.7}$	13.4$_{\pm 0.3}$	59.6$_{\pm 0.2}$	50.2$_{\pm 0.4}$	41.2
CORAL [48]	59.2$_{\pm 0.1}$	19.7$_{\pm 0.2}$	46.6$_{\pm 0.3}$	13.4$_{\pm 0.4}$	59.8$_{\pm 0.2}$	50.1$_{\pm 0.6}$	41.5
mDSDI [6]	62.1$_{\pm 0.3}$	19.1$_{\pm 0.4}$	49.4$_{\pm 0.4}$	12.8$_{\pm 0.7}$	62.9$_{\pm 0.3}$	50.4$_{\pm 0.4}$	42.8
SWAD [8]	66.0$_{\pm 0.1}$	22.4$_{\pm 0.3}$	53.5$_{\pm 0.1}$	16.1$_{\pm 0.2}$	65.8$_{\pm 0.4}$	55.5$_{\pm 0.3}$	46.5
DNA [12]	66.1$_{\pm 0.2}$	23.0$_{\pm 0.1}$	54.6$_{\pm 0.1}$	**16.7**$_{\pm 0.1}$	65.8$_{\pm 0.2}$	56.8$_{\pm 0.1}$	47.2
iDAG	**67.9**$_{\pm 0.5}$	**24.2**$_{\pm 0.4}$	**55.0**$_{\pm 0.7}$	16.4$_{\pm 0.3}$	**66.1**$_{\pm 0.5}$	**56.9**$_{\pm 0.4}$	**47.7**

Table 5. Test accuracy on CMNIST by MLP of hidden size 390 with varied training sample size.

Sample size	50K	40K	30K	20K	15K	10K	5K
Oracle	72.45	71.61	70.19	69.45	68.11	66.99	64.15
ERM [52]	10.80	11.03	11.08	13.58	16.22	18.20	21.04
DILU [56]	50.22	52.31	45.31	44.21	48.92	43.14	43.83
IRMv1 [3]	67.45	65.25	63.46	58.67	49.51	35.60	26.19
InvRat [9]	66.35	66.61	61.05	57.25	50.04	34.28	25.42
BIRM [30]	69.97	69.47	69.06	67.02	66.78	66.40	60.01
iDAG	**71.82**	**71.73**	**70.8**	**70.40**	**69.84**	**68.52**	**64.23**

Table 6. Ablation study on conventional OOD benchmarks.

Ablation	PACS	OfficeHome	DomainNet	CMNIST
iDAG	88.8	71.8	47.8	71.8
w/o $\mathcal{L}_{CL-\nu}$	85.7	69.8	45.5	23.5
w/o $\mathcal{L}_{CL-\mu}$	86.1	70.3	46.1	30.1
w/o acyclicity	88.1	71.1	46.9	11.2
w/o sparsity	88.1	71.1	47.2	68.8

the variant of iDAG. As shown in Table 6, the acyclic constraint brings a significant performance improvement on the CMNIST benchmark. On conventional benchmarks as well, the iDAG also outperforms the variant without an acyclic penalty. It makes intuitive sense since these three datasets mainly suffer from diversity shift while containing less spurious relations, as reported in [59]. By contrast, the CMNIST is an extremely biased dataset that is dominated by spurious relations. The different performance gaps between the iDAG and variant on two dimensions of shifts indicate that cyclic constraint is closely related to spurious correlations. Therefore, our iDAG framework does indeed suppress the spurious correlations for better domain generalization performance.

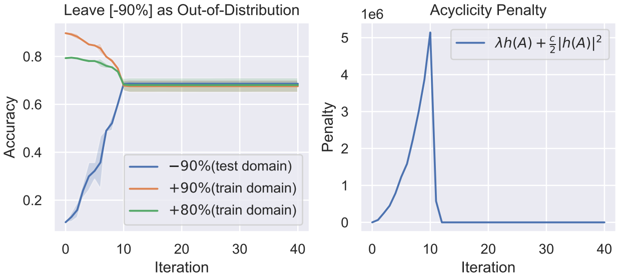

Figure 3. Illustration of training iDAG on CMNIST. As the training proceeds, the overfitting on train domains $+90\%, +80\%$ quickly decreases with the penalty being scaled up. Ultimately, both train and test domains achieve oracle-level accuracy, while the acyclic property is guaranteed.

Effect of contrastive enhancement. We ablate the contributions of two contrastive learning components of iDAG: in-domain PCL and cross-domain PCL. In particular, we compare iDAG with two variants: (i)-iDAG w/o in-domain CL $\mathcal{L}_{\mathrm{CL}-\nu}$; (ii)-iDAG w/o cross-domain CL $\mathcal{L}_{\mathrm{CL}-\mu}$. We first evaluate the effect of removing $\mathcal{L}_{\mathrm{CL}-\nu}$, the results in the Table 6 show that the iDAG with contrastive learning applied on domain-specific prototypes brings great performance improvement on all benchmarks, especially on the CMNIST. These results indicate that the clustering effect improves the representativeness of prototypes thereby leading to a more accurate DAG, otherwise it is hard to overcome the spurious relations as shown in the CMNIST case.

Second, we conduct the ablation by removing the contrastive learning loss $\mathcal{L}_{\mathrm{CL}-\mu}$ on domain invariant features, the results in the Table 6 show that iDAG outperforms the variant w/o $\mathcal{L}_{\mathrm{CL}-\mu}$. This effectiveness validates the benefits of clustering effects in improving classification.

Effect of sparsity regularization. We then explore the effect of sparsity regularization in learning DAG on iDAG performance. From Table 6 we can observe that iDAG substantially outperforms the variant of removing sparsity regularization. An intuitive sense related to sparsity is beneficial for learning more reliable DAGs. The effectiveness of sparsity in iDAG is well correlated to many empirical shreds of evidence that indicate it plays in learning more stable DAGs [66], *i.e.*, pruning weak edges (unstable correlations) for optimal DAG learning.

5.5. Further Analysis on Overfitting

Figure 3 illustrates the dynamics of training and test accuracies over the course of iDAG training, where the model is warmed up via naive ERM training. Note that the training labels contain $\approx 25\%$ noise. However, the ERM model still largely exceeds the level of oracle ($\approx 75\%$) despite the testing performance struggles. This is caused by spurious correlations between color and label in training domains being set to $+90\%, +80\%$, which is poorly generalizable to the testing domain. When start running iDAG, we first observe the acyclicity penalty term increases as it searches the invariant causal structure. Along with the penalty changing, the training/testing performance also drops/increases to the expected oracle accuracy. Once the invariant DAG is discovered, the acyclicity is (almost) ensured such that the penalty value converges to zero. As a result, the training and testing performance also converge to the oracle accuracy as expected. In view of these facts, it is evident that iDAG is effective at detecting and eliminating spurious correlations, which suppresses the overfitting problem.

6. Conclusion

In this paper, we tackle the domain generalization problem from a novel invariant causal graph identification perspective. To achieve this goal, we design a constrained optimization problem and collect the data prototypes from all domains and categories for efficient computation. We also incorporate a hierarchical contrastive learning module to promote DAG exploration as well as stable prediction. Empirically, we achieve state-of-the-art out-of-domain generalization performance on various benchmarks, especially the case that is heavily influenced by spurious correlations. We hope our work will inspire the community towards a broader view of tackling the domain generalization problem from its intrinsic causal structures.

Acknowledgment

This work is supported by the National Key R&D Program of China (2020YFB1313501), National Natural Science Foundation of China (T2293723,61972347), Zhejiang Provincial Natural Science Foundation (LR19F020005), the Key R&D program of Zhejiang Province (2021C03003, 2022C01119, 2022C01022), the Fundamental Research Funds for the Central Universities (No. 226-2022-00051). JZ also wants to thank the support by the National Key R&D Program of China (No. 2022YFB3304101), NSFC Grants (No. 62206247), Fundamental Research Funds for the Central Universities (No. 226-2022-00028) and CAAI-Huawei Open Fund.

References

[1] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Adversarial Invariant Feature Learning with Accuracy Constraint for Domain Generalization. In *Machine Learning and Knowledge Discovery in Databases*, 2020.

[2] Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H. Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching, 2019.

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant Risk Minimization. *ArXiv preprint*, 2019.

[4] Gilles Blanchard, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. Domain Generalization by Marginal Transfer Learning. *Journal of Machine Learning Research*, 2021.

[5] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Proc. of NeurIPS*, 2011.

[6] Manh-Ha Bui, Toan Tran, Anh Tran, and Dinh Phung. Exploiting Domain-Specific Features to Enhance Domain Generalization. In *Proc. of NeurIPS*, 2021.

[7] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proc. of CVPR*, 2019.

[8] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. SWAD: Domain Generalization by Seeking Flat Minima. In *Proc. of NeurIPS*, 2021.

[9] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi S. Jaakkola. Invariant rationalization. In *Proc. of ICML*, 2020.

[10] David Maxwell Chickering. Learning Equivalence Classes of Bayesian-Network Structures. *Journal of Machine Learning Research*, 2002.

[11] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 2002.

[12] Xu Chu, Yujie Jin, Wenwu Zhu, Yasha Wang, Xin Wang, Shanghang Zhang, and Hong Mei. DNA: Domain Generalization with Diversified Neural Averaging. In *Proc. of ICML*, 2022.

[13] Abhimanyu Dubey, Vignesh Ramanathan, Alex Pentland, and Dhruv Mahajan. Adaptive Methods for Real-World Domain Generalization. In *Proc. of CVPR*, 2021.

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 2016.

[15] Tian Gao and Dennis Wei. Parallel bayesian network structure learning. In *Proc. of ICML*, 2018.

[16] Mingming Gong, Kun Zhang, Biwei Huang, Clark Glymour, Dacheng Tao, and Kayhan Batmanghelich. Causal Generative Domain Adaptation Networks, 2018.

[17] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *Proc. of ICML*, 2016.

[18] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *Proc. of ICLR*, 2021.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. of CVPR*, 2020.

[20] Zeyi Huang, Haohan Wang, Eric P. Xing, and Dong Huang. Self-challenging Improves Cross-Domain Generalization. In *Proc. of ECCV*, 2020.

[21] Jean Kaddour, Aengus Lynch, Qi Liu, Matt J. Kusner, and Ricardo Silva. Causal Machine Learning: A Survey and Open Problems, 2022.

[22] David Krueger, Ethan Caballero, Jörn-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Rémi Le Priol, and Aaron C. Courville. Out-of-distribution generalization via risk extrapolation (rex). In *Proc. of ICML*, 2021.

[23] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *Proc. of ICCV*, 2017.

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

[25] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *Proc. of ICCV*, 2019.

[26] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *Proc. of CVPR*, 2018.

[27] Junnan Li, Caiming Xiong, and Steven C. H. Hoi. CoMatch: Semi-Supervised Learning With Contrastive Graph Regularization. In *Proc. of ICCV*, 2021.

[28] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations. In *Proc. of ICLR*, 2021.

[29] Ya Li, Mingming Gong, Xinmei Tian, Tongliang Liu, and Dacheng Tao. Domain generalization via conditional invariant representations. In *Proc. of AAAI*, 2018.

[30] Yong Lin, Hanze Dong, Hao Wang, and Tong Zhang. Bayesian Invariant Risk Minimization. In *Proc. of CVPR*, 2022.

[31] Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant Causal Representation Learning for Out-of-Distribution Generalization. In *Proc. of ICLR*, 2022.

[32] Marloes H. Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 2009.

[33] Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M. Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Proc. of NeurIPS*, 2018.

[34] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards Recognizing Unseen Categories in Unseen Domains. In *Proc. of ECCV*, 2020.

[35] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *Proc. of ICML*, 2013.

[36] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing Domain Gap by Reducing Style Bias. In *Proc. of CVPR*, 2021.

[37] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, U.K. ; New York, 2000.

[38] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. of ICCV*, 2019.

[39] Alexandre Rame, Corentin Dancette, and Matthieu Cord. Fishr: Invariant Gradient Variances for Out-of-Distribution Generalization. In *Proc. of ICML*, 2022.

[40] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *Proc. of ICLR*, 2020.

[41] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward Causal Representation Learning. *Proceedings of the IEEE*, 2021.

[42] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *Proc. of ICLR*, 2018.

[43] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards Out-Of-Distribution Generalization: A Survey, 2021.

[44] Yuge Shi, Jeffrey Seely, Philip H. S. Torr, N. Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient Matching for Domain Generalization, 2021.

[45] Peter Spirtes and Clark Glymour. An Algorithm for Fast Recovery of Sparse Causal Graphs. *Social Science Computer Review*, 1991.

[46] Peter Spirtes, Clark N. Glymour, Richard Scheines, and David Heckerman. *Causation, Prediction, and Search*. MIT Press, 2000.

[47] Adarsh Subbaswamy and Suchi Saria. I-SPEC: An End-to-End Framework for Learning Transportable, Shift-Stable Models, 2020.

[48] Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *Proc. of ECCV*, 2016.

[49] Joshua Tenenbaum and William Freeman. Separating Style and Content. In *Proc. of NeurIPS*, 1996.

[50] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *Proc. of CVPR*, 2011.

[51] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, 2018.

[52] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 1999.

[53] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. of CVPR*, 2017.

[54] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. PiCO: Contrastive Label Disambiguation for Partial Label Learning. In *Proc. of ICLR*, 2022.

[55] Yufei Wang, Haoliang Li, and Alex C. Kot. Heterogeneous domain generalization via domain mixup. In *Proc. of ICASSP*, 2020.

[56] Zehao Xiao, Jiayi Shen, Xiantong Zhen, Ling Shao, and Cees Snoek. A bit more bayesian: Domain-invariant learning with uncertainty. In *Proc. of ICML*, 2021.

[57] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proc. of AAAI*, 2020.

[58] Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve Unsupervised Domain Adaptation with Mixup Training, 2020.

[59] Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. OoD-Bench: Quantifying and Understanding Two Dimensions of Out-of-Distribution Generalization. In *Proc. of CVPR*, 2022.

[60] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. In *Proc. of ICML*, 2019.

[61] Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Scholkopf, and Eric P. Xing. Towards Principled Disentanglement for Domain Generalization. In *Proc. of CVPR*, 2022.

[62] Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 2008.

[63] Kun Zhang, Mingming Gong, Petar Stojanov, Biwei Huang, Qingsong Liu, and Clark Glymour. Domain adaptation as a problem of inference on graphical models. In *Proc. of NeurIPS*, 2020.

[64] Marvin Zhang, Henrik Marklund, Nikita Dhawan, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive Risk Minimization: Learning to Adapt to Domain Shift, 2020.

[65] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. Towards Unsupervised Domain Generalization. In *Proc. of CVPR*, 2022.

[66] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with NO TEARS: continuous optimization for structure learning. In *Proc. of NeurIPS*, 2018.

[67] Wenxuan Zhou, Fangyu Liu, and Muhao Chen. Contrastive out-of-distribution detection for pretrained transformers. In *Proc. of EMNLP*, 2021.

[68] Xiao Zhou, Yong Lin, Weizhong Zhang, and Tong Zhang. Sparse Invariant Risk Minimization. In *Proc. of ICML*, 2022.

[69] Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *Proc. of ICML*, 2021.