Contrastive-ACE: Domain Generalization Through Alignment of Causal Mechanisms

Yunqi Wang, Furui Liu, *Member, IEEE*, Zhitang Chen, *Member, IEEE*, Yik-Chung Wu, *Senior Member, IEEE*, Jianye Hao, *Member, IEEE*, Guangyong Chen, *Member, IEEE*, and Pheng-Ann Heng, *Senior Member, IEEE*

***Abstract*—Domain generalization aims to learn knowledge invariant across different distributions while semantically meaningful for downstream tasks from multiple source domains, to improve the model's generalization ability on unseen target domains. The fundamental objective is to understand the underlying "invariance" behind these observational distributions and such invariance has been shown to have a close connection to causality. While many existing approaches make use of the property that causal features are invariant across domains, we consider the invariance of the average causal effect of the features to the labels. This invariance regularizes our training approach in which interventions are performed on features to enforce stability of the causal prediction by the classifier across domains. Our work thus sheds some light on the domain generalization problem by introducing invariance of the mechanisms into the learning process. Experiments on several benchmark datasets demonstrate the performance of the proposed method against SOTAs. The codes are available at: https://github.com/lithostark/Contrastive-ACE.**

***Index Terms*—Causal inference, domain generalization, deep learning.**

I. Introduction

The past decades have witnessed the remarkable success of machine learning, especially deep learning models in solving different problems in various fields. However, the performance guarantee of models is under the assumption that the training and testing data are independent and identically distributed, which can be easily violated in real-world applications since the data-generating processes are usually affected by time, environment, and experimental conditions, etc. As a result, models that work well on training data may perform poorly on new data unseen in the model training stage, and thus restrain their deployment for further applications. It is of great interest to learn a domain-robust model that can be generalized to the domains beyond source data. To this end, researchers proposed the domain generalization (DG) problem [1], which aims at improving the robustness of models on unseen data (i.e., target domain) by learning from several training datasets (i.e., source domains).

The work was supported in part by a Key Research Project of Zhejiang Lab (No. 2022PE0AC04). (*Corresponding author: Furui Liu.*)

Yunqi Wang and Yik-Chung Wu are with The University of Hong Kong, Hong Kong, China (email: yunqi9@connect.hku.hk; ycwu@eee.hku.hk).

Furui Liu and Guangyong Chen are with Zhejiang Lab, Hangzhou 311121, China (email: liufurui@zhejianglab.com; gychen@zhejianglab.com).

Zhitang Chen and Jianye Hao are with Noah's Ark Lab, Huawei Technologies, Hong Kong, China (email: chenzhitang2@huawei.com; haojianye@huawei.com).

Pheng-Ann Heng is with The Chinese University of Hong Kong, Hong Kong, China (email: pheng@cse.cuhk.edu.hk).

The most straightforward domain generalization approach is the leave-one-out strategy, which defines one as the target domain for testing and the rest as source domains for training. To tackle the challenging problem that no data from the target domain is available in training, efforts have been made to extract "invariance" from source domains. A natural idea is to frame the network to extract stable features which yield invariant predictions across domains [2], [3], [4], [5], [6], [7]. Methods under this branch either enforce consistency of the distributions of latent features across source domains [3], [5], or minimize the gradients of the classification loss with respect to latent features [6].

The underlying assumption behind these approaches is the postulate that causal features are to certain degree stable. However, recent studies show that the causal mechanism, rather than the distribution of features, is stable across domains [8]. If we only rely on enforcing feature invariance by a regularizer, it may end up with spurious causal features as they can vary across domains, thus leading to unstable trained models that fail to generalize [9]. Besides, by incorporating cross-domain mechanism invariance, one is able to recover the causal mechanism as well as causal features, with good interpretability of the contributions of individual features on the task at hand. This also benefits tasks like troubleshooting and identification of important features.

To this end, we tackle the problem of domain generalization from a causal perspective by treating machine learning models as Structural Causal Models (SCM). Instead of aligning the distributions of latent features across domains, we propose a novel constraint based on the causal attributions in networks measured by Average Causal Effect (ACE) [10]. By viewing samples of the same class across domains as positive pairs and those of different classes as negative pairs, a contrastive-ACE loss is introduced to regularize the learning procedure and encourage domain-independent causal attributions of extracted features. Furthermore, by leveraging the contrastive-ACE loss, domain labels are no longer necessary for learning a domain-robust model, which makes our proposed approach applicable to a wider range of scenarios no matter domain labels are available or not. Even without the assistance of the domain labels, experimental results show that the proposed method achieves better performance compared to those with extra domain label information.

Definitely, information from domain labels can help improve performance even further. So we also construct a domain-predictor to perform a domain classification task based on the causal mechanisms. The domain-predictor is trained

in an adversarial manner to the featurizer and classifier. This encourages the emergence of domain-invariant causal mechanisms during optimization and complements contrastive-ACE from the perspective of domain labels. The proposed method achieves significant improvement than other SOTA methods on VLCS, Office-Home, and miniDomainNet. In particular, an increase of 1.9% in accuracy is achieved by contrastive-ACE compared with CDANN on VLCS, an increase of 0.8% on Office-Home, and an increase of 4.3% on miniDomainNet. The superior experimental results on several benchmark datasets demonstrate the effectiveness of the proposed approach in model generalization compared with several baseline methods and thus show the importance of involving causal attributions in the training of the models.

The contributions of this paper are listed as follows.

1) We propose to quantify the causality of features (we refer to as the causal mechanism) by ACE, and align such causality across domains to obtain causal features. A key point is that ACE does not directly give causal feature, but aligning them would.
2) We design a novel objective function, termed as contrastive-ACE loss, to evaluate the difference between the ACE values of the input feature on output across domains.
3) We also introduce a domain adversarial loss to further utilize domain information through adversarial training and enforce the network to be less dependent on domain-specific features when performing classification task.
4) We carry out extensive experiments on benchmark datasets. The overall performance demonstrates the superiority of the proposed approach in solving DG problems no matter domain labels are available or not.

The rest of the paper is organized as follows. Section II reviews algorithms on domain generalization as well as related works in causal neural network attribution. Section III illustrates the alignment of causal mechanisms and introduces the proposed two novel loss functions. Section IV demonstrates the experimental results. Finally, the paper is concluded in Section V.

II. Related Work

A. Domain Generalization

Domain generalization remains a challenging yet important problem that has been investigated by many studies in the literature. The classic way of learning models with good generalization ability is to train feature extractors that can generate invariant representations across different source domains. Various methods have been proposed including naive approaches where a single network is trained by directly aggregating all data from source domains together [11], with a designed structure for more robust performance on data of multi-domain distributions [12], or modified optimization algorithms which minimize dissimilarity of features between different domains [13]. Specifically, domain invariant component analysis has been proposed to train models under distribution variations resulted from domain shift [3]. In [14], they leverage the maximum mean discrepancy as the measurement to guide training on multi-task auto-encoders, under the principle of aligning source data across domains. Some other works [13], [15], [16] have introduced meta-learning with adaptive regularizers to improve generalization ability. By employing Model-Agnostic Meta-Learning or similar strategies in domain generalization, domain-specific gradients have been normalized [13], [16] and models are encouraged to extract features respecting inter-class relationships [15]. Data augmentation, as utilized in various applications, has also been demonstrated to be effective in domain generalization [17], [18], [19]. Several attempts have been made to enlarge the support of the distributions in training data such as mixing up or blending data points from different domains [17], [20], [21]. Moreover, adversarial data augmentation, as well as several alternatives based on GANs, have also been investigated and show improvements in addressing domain generalization [22], [23], [24]. Recently, a special case of domain generalization, namely single-domain generalization, has received attention, where the training dataset only contains samples from a single domain. In this problem, domain augmentation, i.e., the creation of augmented domains with a different distribution from the source domain, becomes important. Typical works include [25], which introduces adversarial domain augmentation to organize the training of fictitious domains, [26], which uses domain expansion subnetworks to incrementally generate simulated domains, and [27], which proposes an adaptive normalization to learn training samples that are generated by adversarial domain augmentation.

To better interpret domain generalization, literature aiming at capturing invariant relations under the structure of causality has emerged. It is argued that causal features with respect to the task, such as shape for classifying objects, are stable and invariant features one wants to learn. However, simply enforcing invariance to train feature extractors without causal considerations, one may obtain only correlated but non-causal features, that are spurious invariant representations for the task. Consider the image classification as an example. A dataset contains a lot of cows on the grass. Feeding them to a model, the grass may also be learned as "invariant" representation, but it is not the causal feature for identifying the cow [28]. To avoid this, the Invariant Causal Prediction (ICP) is first proposed in [29]. It tries to exploit the invariant property of feature set in causality, in the sense that a structural causal model, as well as the invariant distribution of features, are considered. Several latter studies then made improvements by adding intervention on the target variable and attempt to learn invariant predictors or classifiers [30], [31]. By reformulating the optimization problems, Arjovsky et al. [6] introduce invariant risk minimization to distinguish between spurious correlations and the causal ones, which is then extended to nonlinear settings by [32].

B. Causal Neural Network Attribution

Causal neural network attribution refers to the causal effect of a specific input feature on output prediction in neural networks, which aims to quantify inherent causal influences

in machine learning [33]. Most attribution-based studies [33], [34], [35] have focused on applying overall functional values to define the contribution of input features, while some other methods leverage the gradients or perturbations with occlusion maps to identify the effect of different features [36]. However, these types of methods are prone to artifacts, which are unlikely to be measured accurately due to non-identifiability of the errors. Specifically, they can be treated as approaches for estimating individual causal effect, which fail to consider the complicated interactions among neurons and thus result in a biased measurement of the importance of the input feature.

In a recent work [10], a new unbiased attribution method, called Average Causal Effect (ACE), has been proposed to calculate causal attribution. This metric is derived based on the first principles of causality [37]. Specifically, structural causal models are leveraged by interpreting the original networks as acyclic graphs where higher layers are generated through a hierarchy of interactions on nodes from lower layers and the operator $do(\cdot)$, known as do-calculus, is also used [38]. Do-calculus or intervention in causality literature refers to the artificial perturbation on some variables of the system, expecting to measure their causal influences on others. When one applies an intervention to a variable, it is set to the fixed value. Tracking the system under this condition, the distributions of other variables belonging to the system then are called interventional distributions. The causal effect of the intervened variable on others is defined mathematically based on the interventional distributions [37]. In a similar way, ACE is defined as the subtraction between the expectation of the output when a particular input feature is under intervention, and a baseline output when the same feature is uniformly perturbed in a fixed interval of values.

III. Methods

A domain is defined as a joint distribution over space $\mathcal{X}\times\mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input and label space, respectively. The input consists of images $\{\boldsymbol{x}_i\}$ and the label consists of one-hot vectors $\{\boldsymbol{y}_i\}$. The training data $\mathbf{D}$ in domain generalization consists of several datasets, each of which contains independent and identically distributed instances sampled from one domain. A naïve way to tackle the distribution shift across domains is to aggregate instances from all domains and conduct model training. Suppose there are S instances in total after combining all source data, written as $\mathbf{D}=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^{S}$. The corresponding Empirical Risk Minimization (ERM) loss is

$$\mathcal{L}(\mathbf{D};\theta,\phi)=\sum_{i=1}^{S}\ell\left(g_\phi\left(f_\theta(\boldsymbol{x}_i)\right),\boldsymbol{y}_i\right) \tag{1}$$

where ℓ is an appropriate loss function, $f_\theta:\mathcal{X}\to\mathcal{Z}$ denotes an encoding model parameterized by θ that maps the raw input (image) to a latent feature vector, and $g_\phi:\mathcal{Z}\to\mathcal{Y}$ denotes a model parameterized by ϕ that maps the latent feature vector to the output label. $\mathcal{Z}$ is the space of latent features and

$$\boldsymbol{z}_i=f_\theta(\boldsymbol{x}_i), \tag{2}$$

is the encoding of the observation $\boldsymbol{x}_i$.

To avoid over-fit to the training domains, several different regularizers have been used in addition to the loss as penalty for reducing domain gaps in the space of latent features [3], [5]. Unlike minimizing the cross-domain distance directly in the space of latent features, we provide a novel perspective from causality and impose the invariance on the mechanism for all environments. The basic idea is that the true underlying causal mechanisms that map features to labels are cross-domain invariant. It only depends on class but does not depend on the domain index. For samples in the same class, the true causal mechanism from features to label is similar. However, when the sample is with another class, the mechanism shifts. We design a quantification of the mechanism and use a contrastive loss to enforce this principle in structure learning. Our framework does not rely on domain labels, similar to ERM that collects samples from multiple domains and aggregates them together as the training data. This is an advantage over most of the domain generalization methods, where the domain indexes are essential for representation learning.

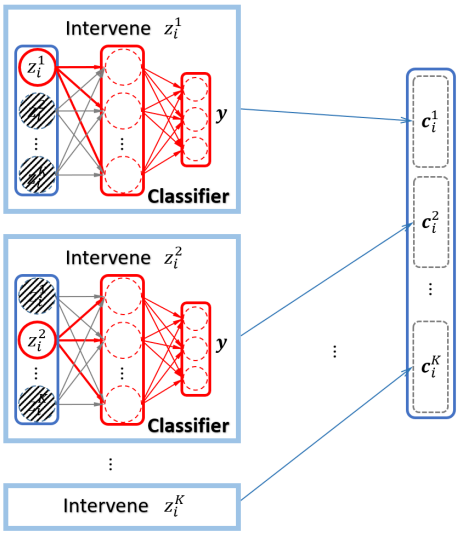

Fig. 1. Computation of the ACE vector. Given the features of a sample $\boldsymbol{z}_i=f_\theta(\boldsymbol{x}_i)$, the ACE quantification of the classifier g_ϕ with K input neurons $\{z^k\}_{k=1}^K$ is a vector $\boldsymbol{c}_i$, which is generated by treating the k-th neuron intervened as $do(z^k=z_i^k)$.

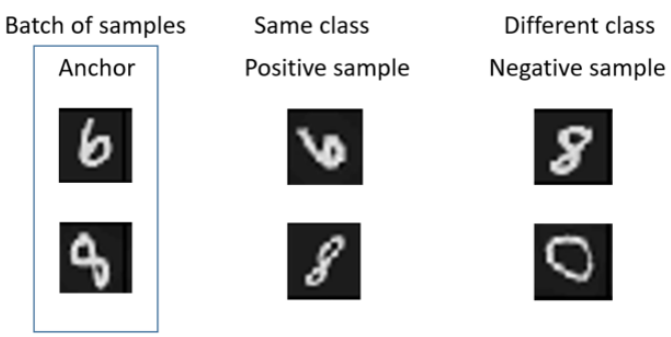

Fig. 2. Triplet generation: for one observational image $\boldsymbol{x}_i$ with label $\boldsymbol{y}_i$, its positive set $\mathcal{P}_i$ consists of images that are in the same class as $\boldsymbol{y}_i$, and its negative set $\mathcal{N}_i$ consists of images that are with a class different from $\boldsymbol{y}_i$.

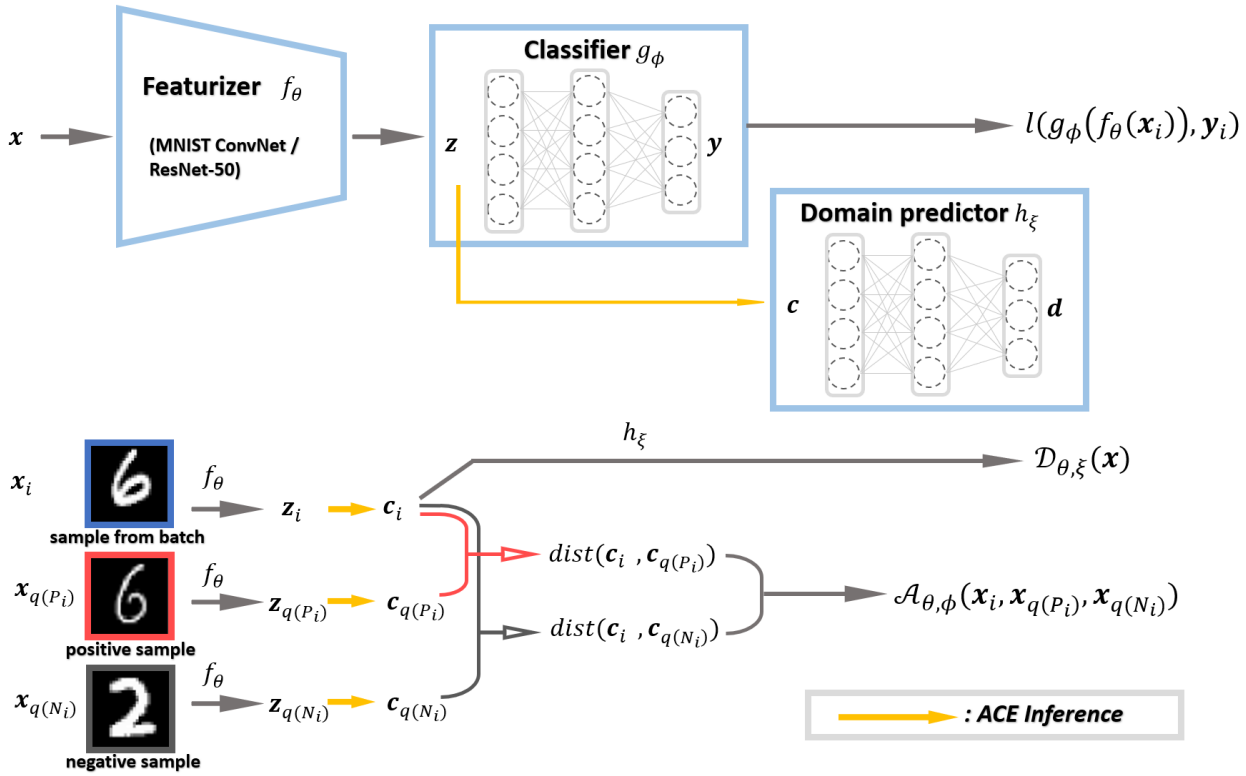

Fig. 3. The framework of our method. For each observational image $\boldsymbol{x}_i$, it first generates the features by the featurizer (or encoder), implemented by residual neural nets as $\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i)$. g_ϕ takes the features to generate the label. The ACE vector $\mathbf{c}_i$ can be computed given the model g_ϕ, and the contrastive-ACE loss for $\boldsymbol{x}_i$ is obtained from the distance between its ACE vector $\mathbf{c}_i$, and the one ($\mathbf{c}_{q(\mathcal{P}_i)}$ or $\mathbf{c}_{q(\mathcal{N}_i)}$) computed using random samples from its positive and negative sets.

Some theoretical ties are linking the neural networks and the causal models [10]. By viewing the domain generalization problem from the causal perspective, one deems all datasets as generated from a typical causal framework, known as Structural Causal Model (SCM). Denote $N(l_1, l_2, \ldots, l_T)$ by a network of l-layers and $l_t \in L = \{l_1, l_2, \ldots, l_T\}$ be the set of neurons in the t-th layer. For neuron L, the set of functions defining causal mechanisms is represented by γ, and the set of exogenous random variables often considered as unobserved common causes is represented by U. The corresponding SCM thus can be expressed as function $f_{\text{SCM}}(L, U, \gamma, P_U)$ with P_U referring to the probability distribution of exogenous random variables in set U. By interpreting the network $N(l_1, l_2, \ldots, l_T)$ as directed acyclic graphs, SCM constructs a hierarchical model which generates outputs of interactions between nodes from lower layers [10]. The flexibility of neural networks also raises confidence in the success of the task of using the neural model to capture the causal mechanism from observational data.

A. Average Causal Effect

To identify the causal mechanisms of the task, it is necessary to quantify the causal effect of each input feature to the output. Correspondingly, we use g_ϕ as the causal quantification model with K input neurons denoted as $\{z^k\}_{k=1}^K$, and an output predicted label $\boldsymbol{y} \in \mathbb{R}^N$, where N is the number of classes. The causal attribution of the neuron z^k on the output $\boldsymbol{y}$ is defined as the average causal effect $\boldsymbol{c}^{\boldsymbol{y}}_{do(z^k=\alpha)}$ with value α. It can be calculated by subtracting the baseline of z^k from the interventional expectation of $\boldsymbol{y}$ when $z^k = \alpha$ [10], i.e.,

$$\boldsymbol{c}^{\boldsymbol{y}}_{do(z^k=\alpha)} = \mathbb{E}\left[\boldsymbol{y}|do(z^k = \alpha)\right] - \mathbb{E}_{z'}\left[\mathbb{E}\left[\boldsymbol{y}|do(z^k = z')\right]\right]. \tag{3}$$

The interventional value α can be set to any value in the input domain of z^k as

$$[\text{low}^k, \text{high}^k]. \tag{4}$$

When not intervened, the input neuron z^k is assumed to be uniformly distributed between low^k and high^k. Specifically, the term $\mathbb{E}[\boldsymbol{y}|do(z^k = \alpha)]$, known as the interventional expectation of output label $\boldsymbol{y}$ conditioning on the intervention operation $do(z^k = \alpha)$, is defined as

$$\mathbb{E}[\boldsymbol{y}|do(z^k = \alpha)] = \int_{\boldsymbol{y}} \boldsymbol{y} \cdot p(\boldsymbol{y}|do(z^k = \alpha))d\boldsymbol{y}. \tag{5}$$

The average interventional expectation of $\boldsymbol{y}$ with respect to z^k, $\mathbb{E}_{z'}[\mathbb{E}[\boldsymbol{y}|do(z^k = z')]]$ is used as the baseline value of $\boldsymbol{y}$, i.e.

$$\mathbb{E}_{z'}[\mathbb{E}[\boldsymbol{y}|do(z^k = z')]] = \int_{\text{low}^k}^{\text{high}^k} p(z') \cdot \int_{\boldsymbol{y}} \boldsymbol{y} \cdot p(\boldsymbol{y}|do(z^k = z'))d\boldsymbol{y}dz^k, \tag{6}$$

which has been demonstrated to be unbiased [10]. Hence, the causal attribution of a feature neuron z^k to an output label $\boldsymbol{y}$ can be quantified by ACE $\boldsymbol{c}^{\boldsymbol{y}}_{do(z^k=\alpha)}$.

We introduce the ACE vector as a quantification of the causal influences of all features on the label of the i-th sample. The feature of the i-th sample is a K-dimensional vector as

$$\boldsymbol{z}_i = f_\theta(\boldsymbol{x}_i) = [z_i^1, z_i^2, ..., z_i^K]^T. \tag{7}$$

The ACE of its k-th feature on the label $\boldsymbol{y}$ is defined as

$$\boldsymbol{c}_i^k = \boldsymbol{c}_{do(z^k=z_i^k)}^{\boldsymbol{y}}. \tag{8}$$

Going through all dimensions, we get an ACE vector of the i-th sample $\boldsymbol{c}_i \in \mathbb{R}^Q$

$$\boldsymbol{c}_i = \left[(\boldsymbol{c}_i^1)^T, (\boldsymbol{c}_i^2)^T, ..., (\boldsymbol{c}_i^K)^T\right]^T, \tag{9}$$

where $Q = NK$.

An illustration of ACE vector computation is in Fig. 1. A simple example is also given in Appendix A.

B. Contrastive-ACE

Inspired by the contrastive representation learning [39], we propose a new objective function named contrastive-ACE loss, to evaluate the difference between the ACE values of the input feature on output $\boldsymbol{y}$ across domains. To match the ACE of an input to an output for all instances of the same class across domains, the contrastive-ACE loss is optimized by minimizing the distance between inputs of the same class and maximizing those from different classes. We treat the inputs of the same class as the positive matches of i-th sample, and the ones of different classes as negative samples. The positive sets and negative sets for the i-th sample are

$$\mathcal{P}_i = \{s|\boldsymbol{y}_i = \boldsymbol{y}_s, \forall s \neq i\}, \tag{10}$$

$$\mathcal{N}_i = \{s|\boldsymbol{y}_i \neq \boldsymbol{y}_s, \forall s \neq i\}, \tag{11}$$

respectively. An intuitive example is also depicted in Fig. 2, taking Rotated MNIST dataset as an example.

A direct way to measure the overall pairwise distance between ACE vectors is to calculate a distance averaged over all samples in the whole set, which is under extremely heavy computational workloads when the number of samples is large. Thus, we use an efficient random sampling technique. Denote $q(\mathcal{P}_i)$ an index sampled uniformly at random from the set $\mathcal{P}_i$, and $q(\mathcal{N}_i)$ an index sampled uniformly at random from the set $\mathcal{N}_i$. The contrastive-ACE loss $\mathcal{A}_{\theta,\phi}(\boldsymbol{x}_i, \boldsymbol{x}_{q(\mathcal{P}_i)}, \boldsymbol{x}_{q(\mathcal{N}_i)})$ is then defined as

$$\begin{aligned}&\mathcal{A}_{\theta,\phi}(\boldsymbol{x}_i, \boldsymbol{x}_{q(\mathcal{P}_i)}, \boldsymbol{x}_{q(\mathcal{N}_i)}) = \\ &\quad \max\left\{dist(\mathbf{c}_i, \mathbf{c}_{q(\mathcal{P}_i)}) - dist(\mathbf{c}_i, \mathbf{c}_{q(\mathcal{N}_i)}) + \delta, 0\right\},\end{aligned} \tag{12}$$

where $\delta > 0$ is a small margin variable [40]. The loss $\mathcal{A}_{\theta,\phi}(\boldsymbol{x}_i, \boldsymbol{x}_{q(\mathcal{P}_i)}, \boldsymbol{x}_{q(\mathcal{N}_i)})$ becomes large when $dist(\mathbf{c}_i, \mathbf{c}_{q(\mathcal{P}_i)})$ is large, or $dist(\mathbf{c}_i, \mathbf{c}_{q(\mathcal{N}_i)})$ is very small. Thus, it penalizes the intra-class dissimilarity and inter-class similarity. The margin variable here is used to reduce the non-robustness brought by the $\max$ operation, avoiding cases that the $dist(\mathbf{c}_i, \mathbf{c}_{q(\mathcal{P}_i)}) - dist(\mathbf{c}_i, \mathbf{c}_{q(\mathcal{N}_i)})$ is always below 0 and never penalized. The distance we use is the Manhattan Distance between the pair of vectors as

$$dist(\mathbf{c}_i, \mathbf{c}_s) = |\mathbf{c}_i - \mathbf{c}_s|_{\mathrm{M}} = \sum_{k=1}^{K} |c_i^k - c_s^k|. \tag{13}$$

Algorithm 1: Contrastive-ACE without domain label (CACE-ND)

Input: Data $\mathbf{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^S$, parameter ρ and δ.
Output: Optimal Network.

Initialize f_θ, g_ϕ;
for $i = 1$ *to* S **do**
  Construct $\mathcal{P}_i$ and $\mathcal{N}_i$

$$\begin{aligned}\mathcal{P}_i &= \{s|\boldsymbol{y}_i = \boldsymbol{y}_s, \forall s \neq i\},\\ \mathcal{N}_i &= \{s|\boldsymbol{y}_i \neq \boldsymbol{y}_s, \forall s \neq i\},\end{aligned}$$

while *Not converged* **do**
  Compute the loss

$$\begin{aligned}\mathcal{L}_{\mathcal{A}}(\mathbf{D}; \theta, \phi) = &\sum_{i=1}^{S} \ell(g_\phi(f_\theta(\boldsymbol{x}_i)), \boldsymbol{y}_i)\\ &+ \rho \sum_{i=1}^{S} \mathcal{A}_{\theta,\phi}(\boldsymbol{x}_i, \boldsymbol{x}_{q(\mathcal{P}_i)}, \boldsymbol{x}_{q(\mathcal{N}_i)}).\end{aligned}$$

  Update θ and ϕ by gradient descent;
return f_θ, g_ϕ;

Combined with the ERM original loss in Eq. 1, the loss with weighting parameter ρ can be written as

$$\begin{aligned}\mathcal{L}_{\mathcal{A}}(\mathbf{D}; \theta, \phi) = &\sum_{i=1}^{S} \ell(g_\phi(f_\theta(\boldsymbol{x}_i)), \boldsymbol{y}_i)\\ &+ \rho \sum_{i=1}^{S} \mathcal{A}_{\theta,\phi}(\boldsymbol{x}_i, \boldsymbol{x}_{q(\mathcal{P}_i)}, \boldsymbol{x}_{q(\mathcal{N}_i)}).\end{aligned} \tag{14}$$

The whole training framework is shown in Fig. 3, and the pseudo-code of the method is presented in Algorithm 1, which does not make use of domain labels. Intuitively, our structural loss originates from the principle that the causal mechanism or structural causal model from features to labels is class-dependent, but domain-independent, or cross-domain stable. Given an observation, the ACE vector is a quantification of its features' influence on its labels. For samples that are within the same class, we minimize the gap between their quantification vectors; but for samples that are with different classes, a larger gap is preferred. The loss that explicitly addresses the principle is designed in a contrastive way that positive and negative pairs are used. Incorporating this in the training procedure, we expect our model to recover the true invariant causal structure, which can achieve stable performance in the presence of domain shifts.

The overall objective of the DG problem is ensuring networks to focus on the classification task without being distracted by domain-induced differences. Under circumstances where domain labels are not available, the proposed contrastive-ACE is able to generalize to novel domains by only leveraging the class information. To be specific, it enhances the similarity between intra-class samples by enforcing same causal mechanism of the features, while enlarging variance between inter-class samples via enforcing different causal mechanisms of the features.

Algorithm 2: Contrastive-ACE with domain label (CACE-D)

Input: Data $\mathbf{D} = \left\{ \{(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{d}_i)\}_{i=1}^{S_m} \right\}_{m=1}^{M}$, parameter ρ, δ and λ.
Output: Optimal Network.

Initialize f_θ, g_ϕ, h_ξ;
for $i = 1$ *to* S **do**
  Construct $\mathcal{P}_i$ and $\mathcal{N}_i$

$$\begin{aligned} \mathcal{P}_i &= \{s | \boldsymbol{y}_i = \boldsymbol{y}_s, \forall s \neq i\}, \\ \mathcal{N}_i &= \{s | \boldsymbol{y}_i \neq \boldsymbol{y}_s, \forall s \neq i\}. \end{aligned}$$

while *Not converged* **do**
  Compute the loss

$$\begin{aligned} \mathcal{L}_{\mathcal{A}}(\mathbf{D}; \theta, \phi) = &\sum_{i=1}^{S} \ell(g_\phi(f_\theta(\boldsymbol{x}_i)), \boldsymbol{y}_i) \\ &+ \rho \sum_{i=1}^{S} \mathcal{A}_{\theta,\phi}(\boldsymbol{x}_i, \boldsymbol{x}_{q(\mathcal{P}_i)}, \boldsymbol{x}_{q(\mathcal{N}_i)}). \end{aligned}$$

  Compute the domain classification loss

$$\mathcal{D}_{\theta,\xi}(\boldsymbol{x}) = \sum_{m=1}^{M} \left(\frac{1}{S_m} \sum_{i=1}^{S_m} \ell\left(h_\xi(\boldsymbol{c}_i), \boldsymbol{d}_i\right) \right).$$

  Obtain the overall loss by

$$\min_{\theta,\phi} \max_{\xi} \mathcal{L}(\mathbf{D}; \theta, \phi, \xi) = \mathcal{L}_{\mathcal{A}}(\mathbf{D}; \theta, \phi) - \lambda \mathcal{D}_{\theta,\xi}(\boldsymbol{x})$$

  Update ξ by gradient ascent;
  Update θ and ϕ by gradient descent;
return f_θ, g_ϕ, h_ξ;

For scenarios where informative domain labels are available, the generalization capability of our approach can be further improved by adopting a pincer attack strategy to handle DG problem. In other words, besides encouraging the network to rely on features that best representing class information when making decisions, we can also utilize domain information through adversarial training to make sure the network to be less dependent on domain-specific features when performing classification task.

Remark 1. The baseline is not necessary when calculating the triplet loss because the baseline is the same for all samples and this term will be eliminated when calculating the difference between the causal vectors of the two samples. However, the purpose of keeping this term is to ensure the conceptual integrity of the causal vectors. When applying a different contrastive loss, such as NT-Xent loss or Margin loss, where the similarity between the two causal vectors is used instead of their difference, the causal vectors defined in Eq. 3 can still be applied without further modification.

C. Incorporating Domain-adversarial Loss

In scenarios where domain labels are available, the generalization performance of our proposed approach can be further improved. Inspired by the domain-adversarial training introduced in [42], we add an additional module h_ξ named domain-predictor that transforms ACE vectors $\{\boldsymbol{c}_i\}$ into domain labels $\{\boldsymbol{d}_i\}$. Suppose M is the number of training domains and S_m is the number of samples in the m-th training domain. The domain classification loss can be expressed as

$$\mathcal{D}_{\theta,\xi}(\boldsymbol{x}) = \sum_{m=1}^{M} \left(\frac{1}{S_m} \sum_{i=1}^{S_m} \ell\left(h_\xi(\boldsymbol{c}_i), \boldsymbol{d}_i\right) \right). \tag{15}$$

Different from the ACE loss $\mathcal{A}_{\theta,\phi}$, the calculation of domain classification loss $\mathcal{D}_{\theta,\xi}$ only involves ACE vectors from anchor samples $\{\boldsymbol{x}_i\}$. The overall objective function is designed in an adversarial manner, i.e.,

$$\min_{\theta,\phi} \max_{\xi} \mathcal{L}(\mathbf{D}; \theta, \phi, \xi) = \mathcal{L}_{\mathcal{A}}(\mathbf{D}; \theta, \phi) - \lambda \mathcal{D}_{\theta,\xi}(\boldsymbol{x}), \tag{16}$$

where λ is a weighting parameter of the domain classification loss. The whole training framework and the pseudo-code of the method when the domain labels are available is illustrated in Algorithm 2. The parameter ξ of domain-predictor h_ξ is optimized to minimize the domain classification loss $\mathcal{D}_{\theta,\xi}(\boldsymbol{x})$ to leverage the domain information in the ACE vectors $\boldsymbol{c}$ across different domains. On the other hand, the parameters θ, ϕ of featurizer f_θ and classifier g_ϕ are optimized to minimize the classification loss and ACE regularizer $\mathcal{L}_{\mathcal{A}}(\mathbf{D}; \theta, \phi)$, while maximizing the domain classification loss $\mathcal{D}_{\theta,\xi}(\boldsymbol{x})$. In this way, the domain labels $\boldsymbol{d}$ are used to reduce domain information during training, thereby improving the generalization performance on unseen domains.

IV. Experiments

In this section, we perform experiments to test the performance of our method on several benchmark datasets, including simulated dataset (Rotated MNIST [14]) and real-world datasets (PACS [11], VLCS [50], Office-Home [51], miniDomainNet [52]). We compare our method with a set of domain generalization approaches. Out of them, ERM is without using the domain indexes, and all other methods take use of the domain indexes. Accuracy is the main metric being compared.

A. Experimental Settings

We mostly follow the model set up in the paper [53]. The models are trained on source domains that are generated from training dataset and evaluated on the target domain which is generated from testing dataset. Source and target domains are generated by the leave-one-out strategy, that one domain is the test and others are as training domains. When image data is the input, the domain generalization models contain the encoder f_θ and the classifier g_ϕ. For fair comparisons, all models are with the same f_θ and g_ϕ as the DomainBed [53]. When we perform experiments on Rotated MNIST, the featurizer f_θ of all the compared algorithms is implemented by the architecture called "MNIST ConvNet"[53] , while on other dataset (VLCS, PACS, Office-Home, miniDomainNet), the featurizer is implemented by Resnet-50 pretrained on ImageNet for all the compared algorithms. The classifier g_ϕ consists of 3 linear layers, each followed by a rectified linear unit (ReLU) activation function. For model selection, we use the test-domain-validation-set, where a validation set that

TABLE I
Accuracy of Rotated MNIST dataset on target domains from 0° to 75°. *This result is obtained from normalized Rotated MNIST dataset.

Method	0°	15°	30°	45°	60°	75°	**Avg**	**Domain Label**
IRM [6]	94.9 ± 0.6	98.7 ± 0.2	98.6 ± 0.1	98.6 ± 0.2	98.7 ± 0.1	95.2 ± 0.3	97.45	required
Mixup [7]	95.8 ± 0.3	98.7 ± 0.0	99.0 ± 0.1	98.8 ± 0.1	98.8 ± 0.1	96.6 ± 0.2	97.95	
MLDG [41]	95.7 ± 0.2	98.9 ± 0.1	98.8 ± 0.1	98.9 ± 0.1	98.6 ± 0.1	95.8 ± 0.4	97.78	
CORAL [4]	96.2 ± 0.2	98.8 ± 0.1	98.8 ± 0.1	98.8 ± 0.1	98.9 ± 0.1	96.4 ± 0.2	97.98	
MMD [5]	96.1 ± 0.2	98.9 ± 0.0	99.0 ± 0.0	98.8 ± 0.0	98.9 ± 0.0	96.4 ± 0.2	98.02	
DANN [42]	95.9 ± 0.1	98.9 ± 0.1	98.6 ± 0.2	98.7 ± 0.1	98.9 ± 0.0	96.3 ± 0.3	97.88	
CDANN [43]	95.9 ± 0.2	98.8 ± 0.0	98.7 ± 0.1	98.9 ± 0.1	98.8 ± 0.1	96.1 ± 0.3	97.87	
SAND-mask [44]	94.7 ± 0.2	98.5 ± 0.2	98.6 ± 0.1	98.6 ± 0.1	98.5 ± 0.1	95.2 ± 0.1	97.35	
MTL [45]	96.1 ± 0.2	98.9 ± 0.0	99.0 ± 0.0	98.7 ± 0.1	99.0 ± 0.0	95.8 ± 0.3	97.92	
SagNet [46]	95.9 ± 0.1	99.0 ± 0.1	98.9 ± 0.1	98.6 ± 0.1	98.8 ± 0.1	96.3 ± 0.1	97.92	
ARM [47]	95.9 ± 0.4	99.0 ± 0.1	98.8 ± 0.1	98.9 ± 0.1	99.1 ± 0.1	96.7 ± 0.2	98.07	
VREx [48]	95.5 ± 0.2	99.0 ± 0.0	98.7 ± 0.2	98.8 ± 0.1	98.8 ± 0.0	96.4 ± 0.0	97.87	
RSC [49]	95.4 ± 0.1	98.6 ± 0.1	98.6 ± 0.1	98.9 ± 0.0	98.8 ± 0.1	95.4 ± 0.3	97.62	
CACE-D	96.5 ± 0.3	98.8 ± 0.1	99.4 ± 0.1	99.4 ± 0.1	99.4 ± 0.1	97.5 ± 0.1	**98.50**	
ERM [2]	95.3 ± 0.2	98.7 ± 0.1	98.9 ± 0.1	98.7 ± 0.2	98.9 ± 0.0	96.2 ± 0.2	97.78	not required
CACE-ND	96.4 ± 0.3	98.7 ± 0.2	99.4 ± 0.1	99.4 ± 0.1	99.4 ± 0.1	97.4 ± 0.2	**98.45**	
CACE-ND*	97.4 ± 0.2	99.5 ± 0.1	99.6 ± 0.1	99.6 ± 0.1	99.5 ± 0.1	98.0 ± 0.1	**98.93**	
Contrastive-feature	95.8 ± 0.2	98.5 ± 0.2	99.2 ± 0.1	99.3 ± 0.1	99.3 ± 0.1	96.8 ± 0.1	98.15	

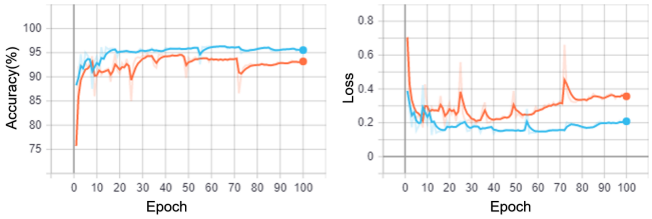

Fig. 4. Performance on target domain. The orange curve records the historical model accuracy when the training algorithm is ERM, and the blue curve records our results.

follows the distribution of the test domain is used to select the best model. The training epoch is fixed to be 100 with batch size 32. Adam optimizer is used without weight decay, with a learning rate to be 0.001. The hyperparameters of our method, namely the weight of the ACE regularizer and the margin variable, are $\rho = 1$ and $\delta = 0.05$. The experiments are run 2 times for each dataset, and the average performance, as well as its statistical variation, are reported.

In order to illustrate the difference between aligning feature vectors and aligning ACE vectors, we also conduct an experiment on all the datasets to investigate the effect of both alignment strategies on different benchmark datasets. For comparison, all the experiments conducted on different benchmark datasets use exactly the same architecture as we proposed, except that the ACE vectors $\boldsymbol{c}_i$ are replaced by the feature vectors $\boldsymbol{z}_i$. This compared approach is denoted as Contrastive-feature.

B. Experiments on Rotated MNIST

This dataset is an artificial dataset constructed from the popular MNIST handwritten digit sets. It contains grayscale MNIST handwritten digits with different rotations, with a degree from 0° to 75°, with 15° as one step interval. The images that are with the same degree of rotation thus naturally form one domain, so that each domain is indexed by the rotation angle.

As reported in Table I, we obtain an average accuracy of 98.50% when the domain labels are available and 98.45% when the domain labels are not available, which is the best among all other approaches. An interesting observation is that the proposed method without the assist from domain information achieves better performance than other approaches with extra domain information. It can also be observed that, by applying contrastive loss on the feature vectors, Contrastive-feature presents noticeably inferior performance in average accuracy compared with our proposed method. This indicates the superiority of aligning causal mechanisms rather than aligning features. When the domain of 0° and 75° rotation is used as the testing domains, methods, in general, perform slightly worse than other settings. Recently, debating about the role of normalization emerges [54] and we also explore its effect on the performance of our approaches. With a simple mean-std normalization of the data, we find that our ACE-based approach achieves a higher accuracy of 98.93%. This is possibly because that the normalized data is with a more stable range, which is less sensitive to additive noises and thus with a ground for making better ACE estimation and recovery of the causal mechanism.

To make an in-depth analysis of the training procedure, we plot the accuracy on testing domains of the models trained by ERM and CACE-ND in Fig. 4. We observe that in the first 10 training epochs, ERM performs much similar to ours, with no obvious difference in-between. Our model clearly outperforms ERM as the training proceeds. This is because that in the initial exploration stage, the structure of neural networks are unstable, and the causal mechanism and features are not recovered to a satisfactory degree. However, when it approaches a relatively mature stage, CACE-ND takes its effect to help guide the model to discover causal features so that stable performance is achieved.

C. Experiments on VLCS

As one of the classic benchmark datasets for domain generalization, VLCS collects natural images from four datasets, i.e. PASCAL VOC2007 (V), LabelMe (L), Caltech (C), and

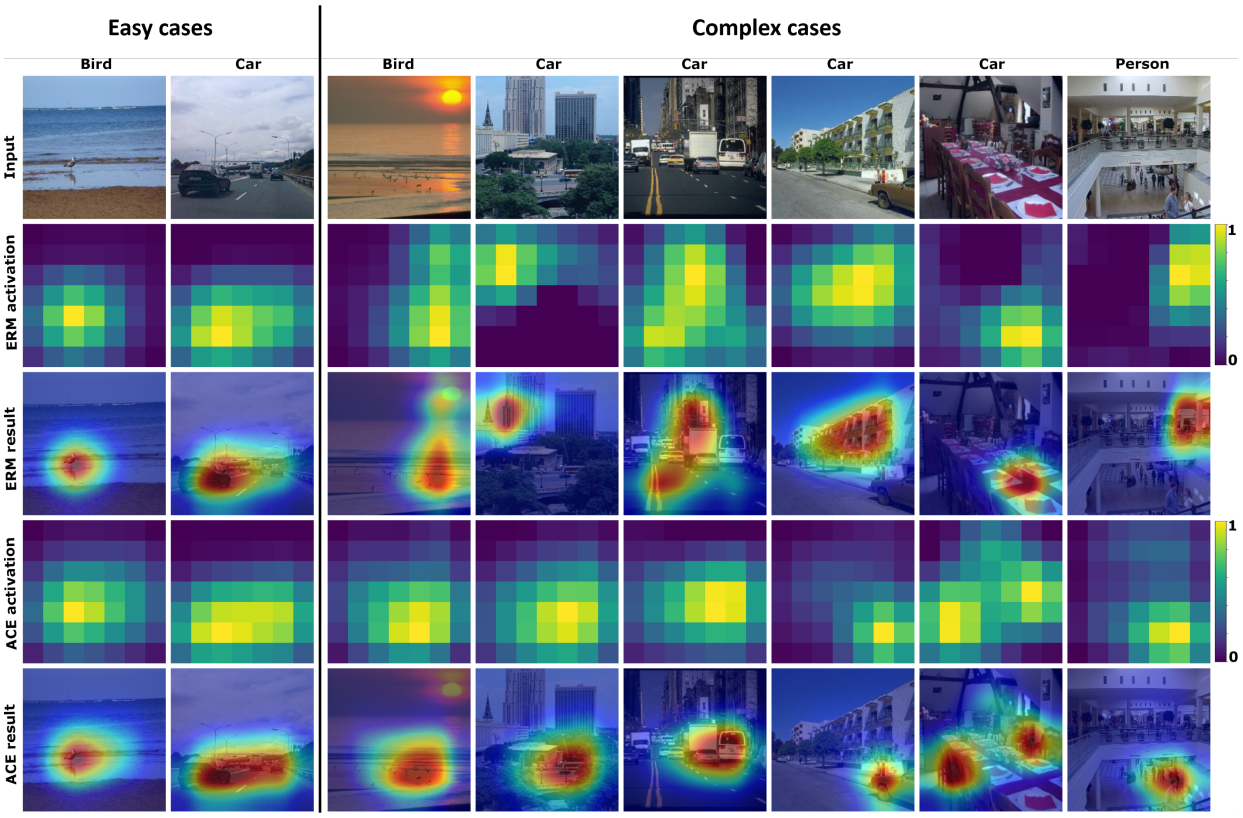

Fig. 5. Visualizations on the receptive fields of features learned by CACE-ND and ERM for samples from the testing dataset of VLCS.

TABLE II
Model accuracy on target domains of VLCS dataset.

Method	C	L	S	V	Avg	Domain Label
IRM [6]	97.3 ± 0.2	66.7 ± 0.1	71.0 ± 2.3	72.8 ± 0.4	76.9	
Mixup [7]	97.8 ± 0.4	67.2 ± 0.4	71.5 ± 0.2	75.7 ± 0.6	78.1	
MLDG [41]	97.1 ± 0.5	66.6 ± 0.5	71.5 ± 0.1	75.0 ± 0.9	77.5	
CORAL[4]	97.3 ± 0.2	67.5 ± 0.6	71.6 ± 0.6	74.5 ± 0.0	77.7	
MMD [5]	98.8 ± 0.0	66.4 ± 0.4	70.8 ± 0.5	75.6 ± 0.4	77.9	
DANN [42]	99.0 ± 0.2	66.3 ± 1.2	73.4 ± 1.4	80.1 ± 0.5	79.7	
CDANN [43]	98.2 ± 0.1	68.8 ± 0.5	74.3 ± 0.6	78.1 ± 0.5	79.9	required
SAND-mask [44]	97.6 ± 0.3	64.5 ± 0.6	69.7 ± 0.6	73.0 ± 1.2	76.2	
MTL [45]	97.9 ± 0.7	66.1 ± 0.7	72.0 ± 0.4	74.9 ± 1.1	77.7	
SagNet [46]	97.4 ± 0.3	66.4 ± 0.4	71.6 ± 0.1	75.0 ± 0.8	77.6	
ARM [47]	97.6 ± 0.6	66.5 ± 0.3	72.7 ± 0.6	74.4 ± 0.7	77.8	
VREx [48]	98.4 ± 0.2	66.4 ± 0.7	72.8 ± 0.1	75.0 ± 1.4	78.1	
RSC [49]	98.0 ± 0.4	67.2 ± 0.3	70.3 ± 1.3	75.6 ± 0.4	77.8	
CACE-D	99.4 ± 0.2	70.2 ± 0.1	76.8 ± 0.3	80.9 ± 0.2	**81.8**	
ERM [2]	97.6 ± 0.3	67.9 ± 0.7	70.9 ± 0.2	74.0 ± 0.6	77.6	
CACE-ND	99.2 ± 0.4	69.5 ± 0.3	75.4 ± 1.0	79.3 ± 0.7	**80.9**	not required
Contrastive-feature	98.9 ± 0.3	65.1 ± 0.2	72.5 ± 0.5	80.1 ± 0.3	79.2	

SUN09 (S), and contains a total of five classes for recognition task (bird, car, chair, dog and person). The images in VLCS are all collected from the real world and have larger intra-class variance and significantly higher domain shift compared to the simulated dataset such as Rotated MNIST. The task of domain generalization thus becomes much more challenging.

As reported in Table II, we achieve an average accuracy of 81.8% when the domain labels are available and 80.9% when the domain labels are not available, which is the best among all other approaches. It is worth noting that, even without the assistant of the additional information from domain labels, the proposed method can still achieve competitive or better results compared to those with extra domain information. We observe a huge difference in the performance across different domains (from 99.2% to 69.5%), which indicates the large distribution shift across domains. As opposed to Rotated MNIST, the approaches that require domain labels perform generally better than those do not in the VLCS dataset. It is in accordance with natural intuition that domain information brought by labels makes much more critical impact when handling datasets of larger distribution shifts. Compared with the increase in accuracy of ERM ($+0.67\%$) in Rotated MNIST,

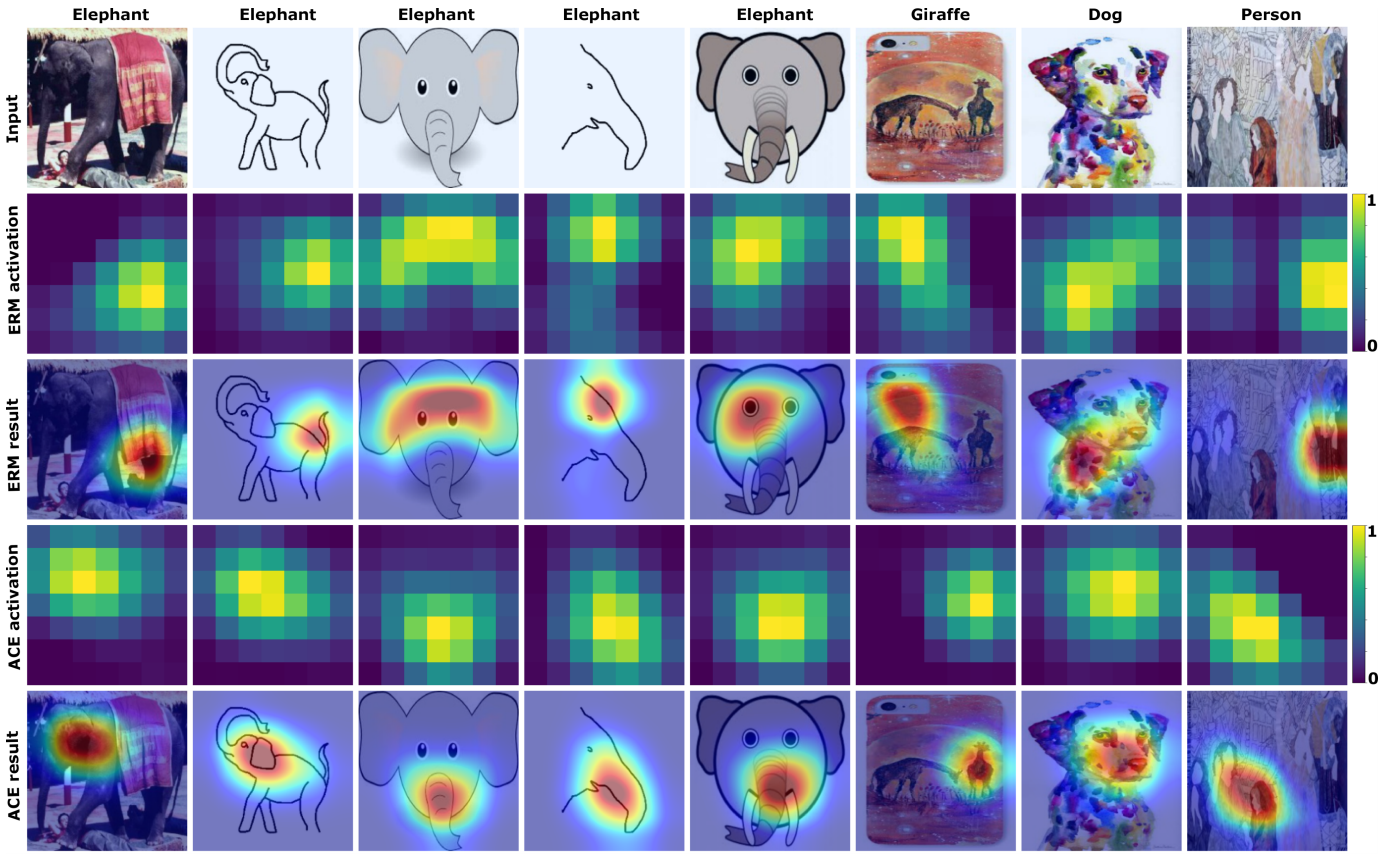

Fig. 6. Visualizations on the receptive fields of features learned by CACE-ND and ERM for samples from the testing dataset of PACS.

TABLE III
Model accuracy on target domains of PACS dataset.

Method	A	C	P	S	Avg	Domain Label
IRM[6]	84.2 ± 0.9	79.7 ± 1.5	95.9 ± 0.4	78.3 ± 2.1	84.5	required
Mixup [7]	87.5 ± 0.4	81.6 ± 0.7	97.4 ± 0.2	80.8 ± 0.9	86.8	
MLDG [41]	87.0 ± 1.2	82.5 ± 0.9	96.7 ± 0.3	81.2 ± 0.6	86.8	
CORAL [4]	86.6 ± 0.8	81.8 ± 0.9	97.1 ± 0.5	82.7 ± 0.6	87.1	
MMD [5]	88.1 ± 0.8	82.6 ± 0.7	97.1 ± 0.5	81.2 ± 1.2	87.2	
DANN [42]	87.0 ± 0.4	80.3 ± 0.6	96.8 ± 0.3	76.9 ± 1.1	85.2	
CDANN [43]	87.7 ± 0.6	80.7 ± 1.2	97.3 ± 0.4	77.6 ± 1.5	85.8	
SAND-mask [44]	86.1 ± 0.6	80.3 ± 1.0	97.1 ± 0.3	80.0 ± 1.3	85.9	
MTL [45]	87.0 ± 0.2	82.7 ± 0.8	96.5 ± 0.7	80.5 ± 0.8	86.7	
SagNet [46]	87.4 ± 0.5	81.2 ± 1.2	96.3 ± 0.8	80.7 ± 1.1	86.4	
ARM [47]	85.0 ± 1.2	81.4 ± 0.2	95.9 ± 0.3	80.9 ± 0.5	85.8	
VREx [48]	87.8 ± 1.2	81.8 ± 0.7	97.4 ± 0.2	82.1 ± 0.7	87.2	
RSC [49]	86.0 ± 0.7	81.8 ± 0.9	96.8 ± 0.7	80.4 ± 0.5	86.2	
CACE-D	89.2 ± 0.5	82.1 ± 0.4	98.0 ± 0.3	80.5 ± 0.4	**87.5**	
ERM [2]	86.5 ± 1.0	81.3 ± 0.6	96.2 ± 0.3	82.7 ± 1.1	86.7	not required
CACE-ND	88.8 ± 1.3	81.9 ± 1.2	97.7 ± 0.2	80.6 ± 0.3	**87.3**	
Contrastive-feature	88.2 ± 0.6	78.3 ± 0.7	97.9 ± 0.3	76.5 ± 0.4	85.2	

the improvement of aligning causal mechanism in VLCS is diminished. As complex real-world images contain more complicated and diverse features than simulated images, it is more difficult to infer the causal relationship between features and predictions, despite the presence of the contrastive-ACE penalty.

As demonstrated in Fig. 5, ERM fails to capture the correct object while CACE-ND can still recognize the object correctly. Moreover, it is worth noting that the receptive field of features learned by ERM tends to cover features that do not belong to the object. As shown in Fig. 5, the ERM focuses on the buildings and double amber lines instead of the cars. Furthermore, for sample input Bird, ERM focuses on the sky and trees rather than the birds. On the contrary, the receptive field of features learned by CACE-ND is concentrated precisely on the object. The reasons for the failure of ERM are probably lie in the biased training dataset, since a large percent of car images are either taken in urban areas or on the road, with backgrounds of buildings and double amber lines. Therefore, the high occurrence rate of these "non-car" objects or scenes

TABLE IV
Model accuracy on target domains of Office-Home dataset.

Method	A	C	P	R	Avg	Domain Label
IRM[6]	56.4 ± 3.2	51.2 ± 2.3	71.7 ± 2.7	72.7 ± 2.7	63.0	required
Mixup[7]	63.5 ± 0.2	54.6 ± 0.4	76.0 ± 0.3	78.0 ± 0.7	68.0	
MLDG[41]	60.5 ± 0.7	54.2 ± 0.5	75.0 ± 0.2	76.7 ± 0.5	66.6	
CORAL[4]	64.8 ± 0.8	54.1 ± 0.9	76.5 ± 0.4	78.2 ± 0.4	68.4	
MMD[5]	60.4 ± 1.0	53.4 ± 0.5	74.9 ± 0.1	76.1 ± 0.7	66.2	
DANN[42]	60.6 ± 1.4	51.8 ± 0.7	73.4 ± 0.5	75.5 ± 0.9	65.3	
CDANN[43]	57.9 ± 0.2	52.1 ± 1.2	74.9 ± 0.7	76.2 ± 0.2	65.3	
SAND-mask [44]	59.9 ± 0.7	53.6 ± 0.8	74.3 ± 0.4	75.8 ± 0.5	65.9	
MTL [45]	60.7 ± 0.8	53.5 ± 1.3	75.2 ± 0.6	76.6 ± 0.6	66.5	
SagNet [46]	62.7 ± 0.5	53.6 ± 0.5	76.0 ± 0.3	77.8 ± 0.1	67.5	
ARM [47]	58.8 ± 0.5	51.8 ± 0.7	74.0 ± 0.1	74.4 ± 0.2	64.8	
VREx [48]	59.6 ± 1.0	53.3 ± 0.3	73.2 ± 0.5	76.6 ± 0.4	65.7	
RSC [49]	61.7 ± 0.8	53.0 ± 0.9	74.8 ± 0.8	76.3 ± 0.5	66.5	
CACE-D	64.9 ± 0.4	54.6 ± 0.3	77.8 ± 0.3	79.4 ± 0.2	**69.2**	
ERM[2]	61.7 ± 0.7	53.4 ± 0.3	74.1 ± 0.4	76.2 ± 0.6	66.4	not required
CACE-ND	64.1 ± 0.3	54.4 ± 0.4	77.3 ± 0.3	79.3 ± 0.3	**68.8**	
Contrastive-feature	66.7 ± 0.8	52.3 ± 0.3	74.8 ± 0.4	76.5 ± 0.7	67.6	

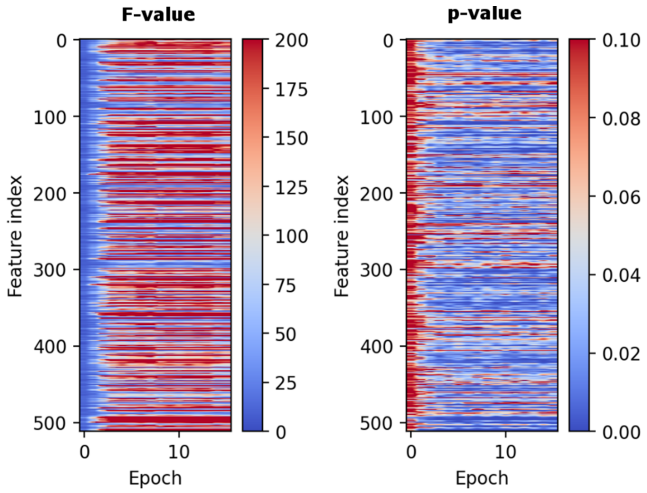

Fig. 7. Results from ANOVA test on the ACE scores of the features.

leads to a high correlation between these features and the class label. It is the same case for the images of birds since birds are more likely to be in the sky or trees when being captured by cameras. Hence, the features representing the sky or trees are highly correlated to the label (i.e., bird). Consequently, ERM is prone to rely on these bias features rather than the ones indicating the target objects when performing classification tasks. As opposed to ERM, by aligning the causal mechanisms, CACE-ND leverages the small number of samples where the cars are not in urban areas and the birds are not in the sky or tree. CACE-ND focuses on the causal features that represent the object itself, and thus improves the overall performance.

D. Experiments on PACS

PACS dataset recently emerges as a widely adopted benchmark dataset for domain generalization, which is even more challenging than VLCS. A total of 7 classes of images (dog, elephant, giraffe, guitar, house, horse, and person) from 4 different domains (art painting, cartoon, photo, and sketch) are included. PACS is considered to have a significantly higher domain shift than VLCS, which attributes to the large difference in style. The objects in PACS dataset are better positioned compared to those in the VLCS dataset. In particular, they take up a large portion of the image occupied and are well centralized.

As reported in Table III, we achieve an average classification accuracy of 87.5% when the domain labels are available and 87.3% when the domain labels are not available, which is the best among all other approaches. An interesting observation is that, even though Contrastive-feature shows slightly better performance on the "Photo" (P) domain, it exhibits lower accuracy on all other domains, especially on the "Cartoon" (C) and "Sketch" (S) domains where its accuracy degrades significantly. Hence the overall inferior performance of Contrastive-feature indicates the lack of generalization ability when contrastive loss is directly applied on the feature vectors.

Interestingly, we observe that, despite the larger inter-source domain divergence in the PACS dataset than in the VLCS dataset [55], [56], the improvement made by the proposed model is relatively smaller on PACS than on VLCS. Because of the better-positioned objects of interest, the representations extracted by the featurizer already contain more information related to the object (casual-related) rather than from the background. Thus, the benefits from aligning causal mechanisms are less significant.

When using domain P for testing and others for training, the performance is superior to that under other settings, owing to the featurizer realized by the backbone of ResNet-50 pretrained on ImageNet, which contains real-world photos from domain P. The bias brought by the featurizer (the pretrained backbone) still influences the generalization capability of the domain models. Although reduced, this bias cannot be completely removed even using ACE contrastive learning. It remains a challenging problem that hasn't been well addressed in the existing literature.

Methods that impose cross-domain invariant representations aim at extracting features that contain domain-independent information while eliminating domain-specific information like styles. However, simply enforcing invariance, one may obtain "over-fixed" patterns without cross-domain flexibility, which is beneficial for classification. It leads to inferior

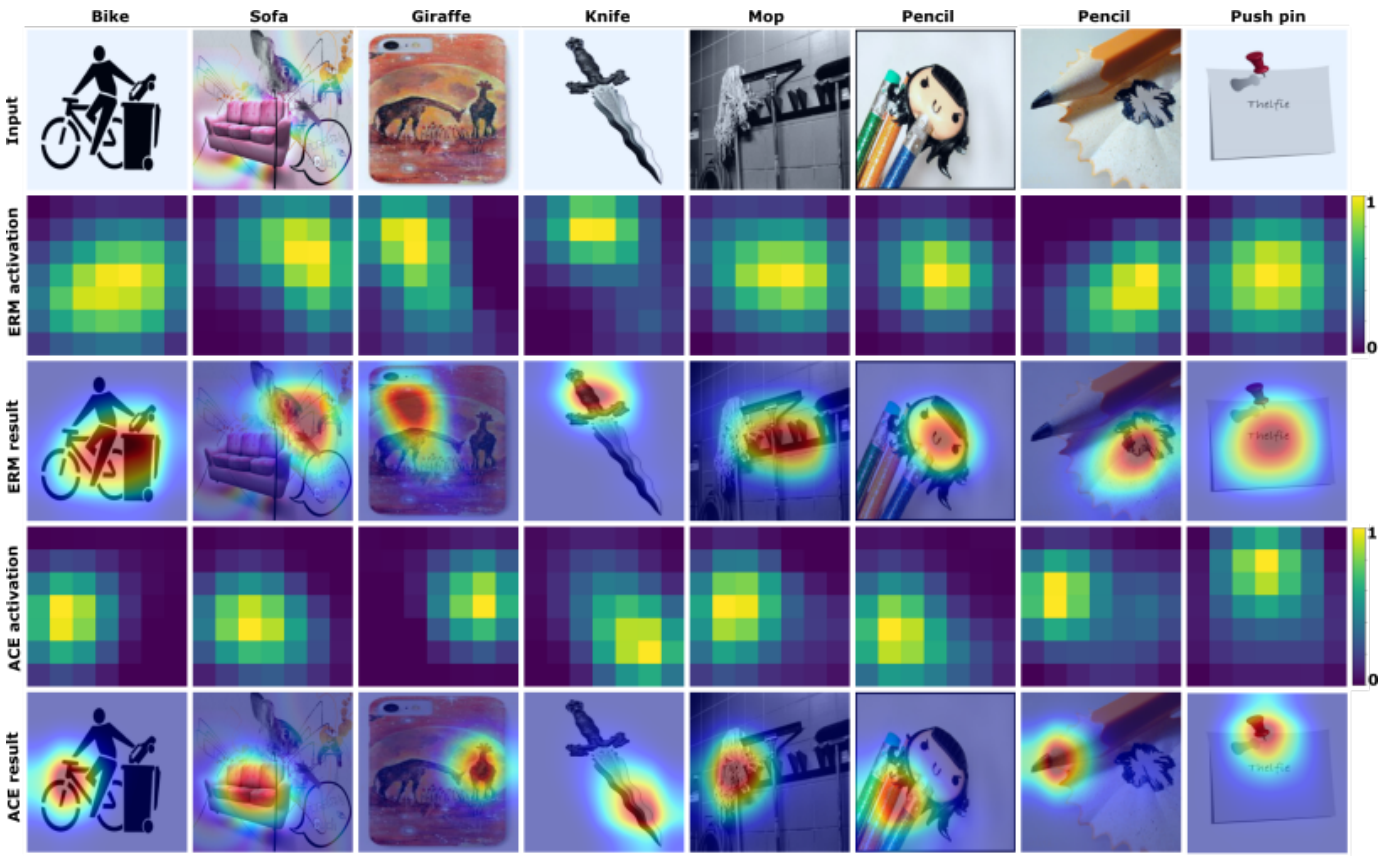

Fig. 8. Visualizations on the receptive fields of features learned by CACE-ND and ERM for samples from the testing dataset of Office-Home.

classification accuracies of their methods. Instead of enforcing learning domain-invariant features, the proposed contrastive-ACE aligns the causal mechanism quantified by the ACE of latent representations to predictions, with room for reasonable variations among patterns. It might be the reason that we get better features for the task at hand.

Fig. 6 presents the receptive fields of features learned by CACE-ND and ERM for selected images from the testing dataset of PACS. We find that ACE is better in capturing causal features of the objects compared with ERM for most cases. It can be observed that the objects in PACS are all clearly presented in the central area of the images, as opposed to VLCS. The domain shift in PACS is mainly due to the style change. Thus, both methods are able to correctly locate the object. However, we can still notice a clear difference. For example, CACE-ND tends to focus on the most distinguishable features of the target, such as the nose of the elephant and the head of the dog, while ERM often emphasizes less representative parts such as the eyes of the elephant or the body of the dog.

We assume that the average causal effect of the features to the labels is invariant. It means that the causal mechanism between features and labels should be of higher intra-class similarity and lower inter-class similarity during the inference and classification process. We then apply ANOVA to analyze the ACE scores of the features for demonstration.

Let the null hypothesis be $H_0 : \mu_1^k = \mu_2^k = ... = \mu_N^k, k = 1, ..., K$, where N represents the number of groups, K represents the number of features and μ_1^k to μ_N^k are the means of ACE scores of the k-th feature for class 1 to N. It indicates no statistical relationship between the ACE scores of each feature and the classes. Here, we use ANOVA for statistics analysis. Under the assumption of the null hypothesis H_0, the F-score should be around 1. The higher the F-score is, the lower probability for H_0 to be true. To reject H_0 with statistical significance, one relies on the p-value, i.e., the score that represents the probability for the obtained F-score to be the least value expected under H_0. Commonly, if $p \leq 0.05$, then the null hypothesis H_0 can be rejected. It can be observed in Fig. 7 that the F-score arises rapidly above 1 and the p-score of most features drops below 0.05 as the training continues. Hence, we demonstrate that the causal mechanisms of the features to the labels are significantly correlated to the classes.

E. Experiments on Office-Home

Office-Home dataset contains $15,500$ images of 65 classes over 4 different domains (artistic images, clip art, product images, and real-world images). It is far more complex compared to PACS and VLCS datasets as it covers much more classes. The office-home has a larger intra-domain variance than PACS and VLCS. For example, the single domain of artistic images includes artistic depictions, sketches, and paintings, which are considered as three separate domains in PACS.

As reported in Table IV, we achieve an average classification accuracy of 69.2% when the domain labels are

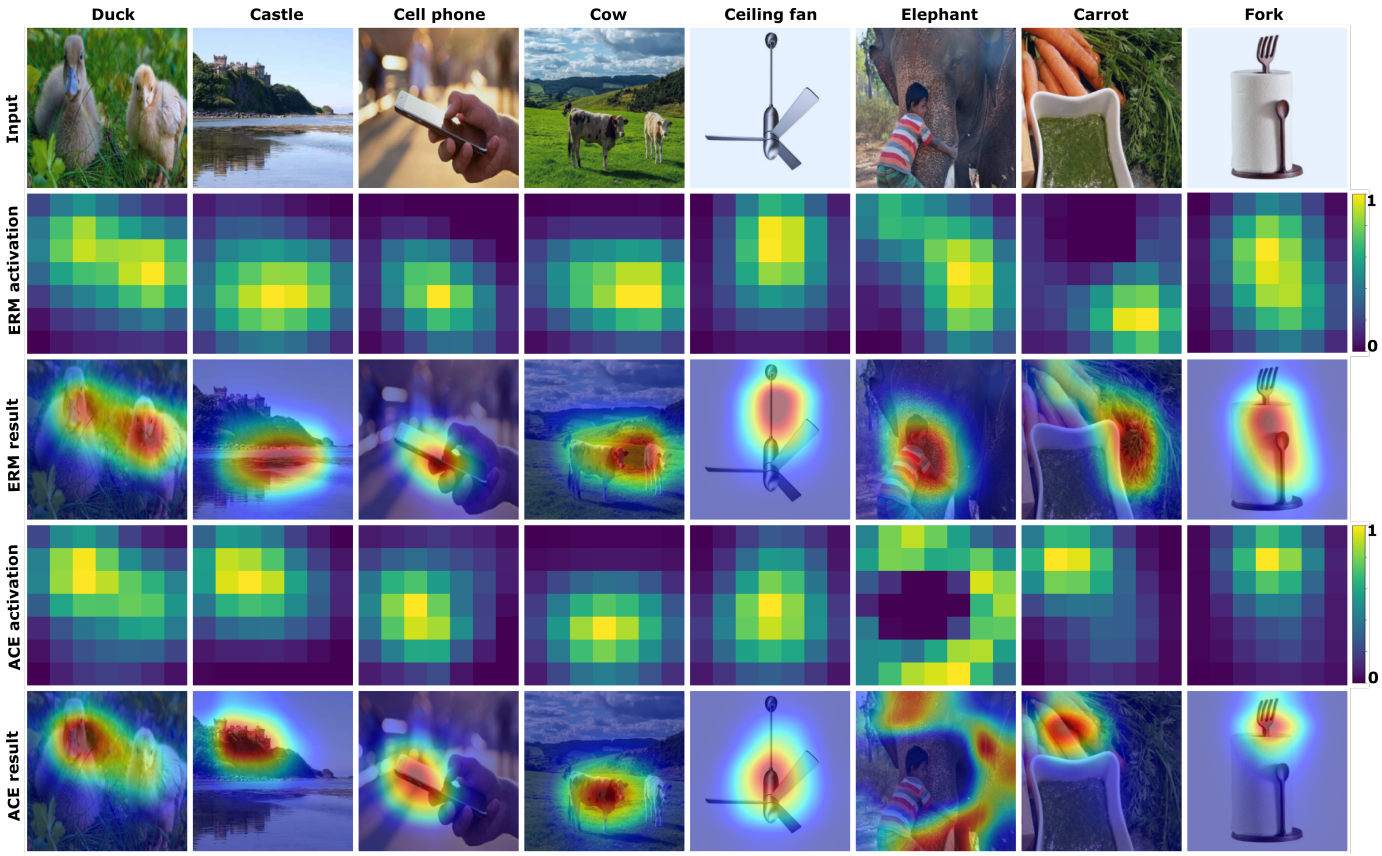

Fig. 9. Visualizations on the receptive fields of features learned by CACE-ND and ERM for samples from the testing dataset of miniDomainNet.

TABLE V
Model accuracy on target domains of miniDomainNet dataset.

Method	Clipart	Painting	Real	Sketch	Avg	Domain Label
DANN[42]	52.8 ± 0.2	50.8 ± 0.5	63.2 ± 0.2	52.6 ± 0.2	54.9	required
CORAL[4]	64.2 ± 0.4	60.0 ± 0.2	73.2 ± 0.4	59.6 ± 0.1	64.3	
Mixup[7]	62.9 ± 0.2	61.8 ± 0.4	71.7 ± 0.3	61.1 ± 0.2	64.4	
CACE-D	70.2 ± 0.3	65.2 ± 0.5	75.8 ± 0.6	63.6 ± 0.3	**68.7**	
ERM[2]	64.1 ± 0.8	62.4 ± 0.9	73.2 ± 0.2	61.1 ± 0.5	65.2	not required
CACE-ND	69.3 ± 0.3	64.8 ± 0.5	75.1 ± 0.4	63.4 ± 0.4	**68.2**	
Contrastive-feature	67.9 ± 0.3	63.1 ± 0.6	74.8 ± 0.3	62.3 ± 0.3	67.0	

available and 68.8% when the domain labels are not available, which is the best among all other approaches. Notice that contrastive-ACE consistently generates superior accuracy on all four datasets, while some SOTA algorithms perform differently on these datasets. Specifically, Mixup and CORAL achieve better performance in Office-Home than in PACS and VLCS, while DANN and C-DANN perform better in VLCS rather than in PACS and Office-Home. The reasons could be that the underlying mechanisms of these approaches are not universal to datasets with different characteristics. For example, the Mixup approach produces samples by linearly interpolating examples between pairs of instances and between their labels, where a larger intra-domain variance could bring more diverse samples. As for CORAL, it matches the mean and variance of the feature distributions, which helps the algorithm focus on stable features across different domains. As a result, both algorithms could benefit from the larger intra-domain variance. For the two approaches that leverage adversarial samples, DANN and C-DANN, the decline in performance on Office-Home and PACS could be due to the greater difficulty in generating adversarial samples for datasets with larger shift across different domains. In comparison, as our proposed contrastive-ACE relies on a universal assumption that the causal mechanism between features and classes should be stable for samples within the same class, aligning such mechanism brings consistently superior performance on all these datasets.

As presented in Fig. 8, the difference between CACE-ND and ERM in the receptive field of the features they learned is much more noticeable. Particularly, CACE-ND shows better capability of accurately locating the target object from the complicated background. For example, it can precisely separate the bike from the person, the push pin from the memo, lamp from other stuff on the desk.

F. Experiments on miniDomainNet

The newly emerged large-scale multi-source benchmark dataset, DomainNet, consists of 0.6M images under a total of 345 classes from 6 domains. It is extremely costly in computing resources when training the full dataset, which imposes stringent requirements on its deployment. To enable wider deployment, a smaller dataset miniDomainNet is constructed following the idea of mini-ImageNet, a popular dataset in the few-shot learning community. miniDomainNet contains a subset of images with the reduced image size of 96×96 from 4 domains of the original DomainNet, i.e., $18,703$ images from Clipart, $31,202$ images from Painting, $65,609$ images from Real, and $24,492$ images from Sketch. Overall, miniDomainNet is an excellent choice for fast prototyping and experimentation, as it largely alleviates the high demand in computation resources for full dataset but still maintains data variety and complexity.

As reported in Table V, we achieve an average classification accuracy of 68.7% when the domain labels are available and 68.2% when the domain labels are not available, which is the best among all other approaches. Specifically, our proposed Contrastive-ACE exceeds the SOTAs that require domain labels up to around 4.3% in the accuracy with domain labels, and also outperforms ERM, when domain labels are not involved, with an improvement of 3%.

The receptive fields of features learned by both ACE and ERM for testing samples from miniDomainNet are visualized and compared in Fig. 9. It is noticeable that ACE can accurately locate the target of interest among all presented examples, even with backgrounds of high complexity, while ERM encounters difficulty in distinguishing the target object from other non-target distractions. For instance, the results of ERM are distracted by the chicken (column 1), the child (column 6), or the spoon (column 8) in the given images, when trying to locating different targets (duck/elephant/fork). Moreover, as opposed to ERM, ACE is capable of identifying the most representative part of the target object for correct classification, such as the visual presentations in column 3 and 5 of Fig. 9, when recognizing the cell phone and the ceiling fan.

G. Ablation Study of low^k and high^k Choices

When calculating the interventional expectation and baseline in Equation 3, the values of $[\text{low}^k, \text{high}^k]$ for feature z^k is determined as $\text{low}^k = \mathbb{E}[\boldsymbol{z}] - 0.5$, $\text{high}^k = \mathbb{E}[\boldsymbol{z}] + 0.5$ for $k = 1, \cdots, K$, where K is the dimension of feature vector. The only hyperparameter is the value 0.5, which is decided empirically. To investigate the effect of different [low, high] choices, we have conducted several experiments on different benchmark datasets. The results are shown in Table VI. It can be observed that the performance of the proposed method is robust to different values of the [low, high] interval.

V. Conclusion

In this paper, we provide a novel perspective on domain generalization by making use of the causal invariance between the average causal effect of the latent representations to the labels. By assuming the mechanism to be label-dependent but domain-independent, we align causal quantification vectors of samples. A novel contrastive-ACE loss is introduced into the training to enforce cross-domain stability in predictions. Without using domain labels, our method still achieves good performance on benchmark datasets compared to SOTAs. The feasibility and effectiveness are demonstrated by extensive experiments. When domain labels are available, we introduce a domain adversarial loss to reduce domain information in the causal mechanism to further improve generalization ability. To the best of our knowledge, this work presents the first investigation on aligning causal mechanisms across domains in the learning process to address domain generalization. We expect that it can motivate researchers to explore this direction.

Appendix A
A Simple Example of Calculating ACE Vector

Given the feature vector $\boldsymbol{z} = f_\theta(\boldsymbol{x})$, the ACE value can be calculated as

$$\boldsymbol{c}^{\boldsymbol{y}}_{do(z^k=\alpha)} = \mathbb{E}\left[\boldsymbol{y}|do(z^k=\alpha)\right] - \mathbb{E}_{z'}\left[\mathbb{E}\left[\boldsymbol{y}|do(z^k=z')\right]\right]. \tag{17}$$

The first term is the interventional expectation, which is computed by

$$\begin{aligned}
&\mathbb{E}\left[\boldsymbol{y}|do(z^k=\alpha)\right] \\
&= \int_{\boldsymbol{y}} \boldsymbol{y}\cdot p(\boldsymbol{y}|do(z^k=\alpha))d\boldsymbol{y} \\
&= \int_{\text{low}^1}^{\text{high}^1}\cdots\int_{\text{low}^{k-1}}^{\text{high}^{k-1}}\int_{\text{low}^{k+1}}^{\text{high}^{k+1}}\cdots\int_{\text{low}^K}^{\text{high}^K} \\
&\quad \boldsymbol{y}\cdot p(\boldsymbol{y}|do(z^k=\alpha))dz^1\cdots dz^{k-1}dz^{k+1}\cdots dz^K,
\end{aligned} \tag{18}$$

where $\boldsymbol{y} = g_\phi(\boldsymbol{z})$, and $\boldsymbol{z} = [z^1, z^2, \cdots, z^K]$. In practice, this expectation is approximated using the sampling approach. It is computed by averaging the output $\boldsymbol{y}$ when sampling all other features $\{z^j\}_{j\neq k}$ from the interval $[\text{low}^j, \text{high}^j]$ while keeping $z^k = \alpha$ fixed.

The second term is the baseline and is computed by

$$\begin{aligned}
&\mathbb{E}_{z'}\left[\mathbb{E}\left[\boldsymbol{y}|do(z^k=z')\right]\right] \\
&= \int_{\text{low}^k}^{\text{high}^k} p(z')\cdot\int_{\boldsymbol{y}}\boldsymbol{y}\cdot p(\boldsymbol{y}|do(z^k=z'))d\boldsymbol{y}dz^k \\
&= \int_{\text{low}^1}^{\text{high}^1}\cdots\int_{\text{low}^K}^{\text{high}^K}\boldsymbol{y}\cdot p(\boldsymbol{y}|\boldsymbol{z})dz^1\cdots dz^K.
\end{aligned} \tag{19}$$

Similar the the first term, it is computed by averaging the output values $\boldsymbol{y}$ when sampling all the features $\{z^k\}$ from the empirical distribution $\mathcal{U}[\text{low}^k, \text{high}^k]$.

For better understanding, we illustrate the detailed procedure with a simplified example. Suppose the training dataset consists of two samples $\boldsymbol{x}_1, \boldsymbol{x}_2$ whose feature vectors are $\boldsymbol{z}_1 = [0.5, 1], \boldsymbol{z}_2 = [0.7, 9]$ respectively, and the classifier g_ϕ is implemented by a linear layer followed by a ReLU function

$$\boldsymbol{y} = \text{ReLU}\left(\mathbf{W}\boldsymbol{z} + \mathbf{b}\right), \tag{20}$$

TABLE VI
Model accuracy with different values of [low high] on benchmark datasets.

Dataset	[low, high]				
	$[\mathbb{E}[\boldsymbol{z}]-0.5, \mathbb{E}[\boldsymbol{z}]+0.5]$	$[\mathbb{E}[\boldsymbol{z}]-2.5, \mathbb{E}[\boldsymbol{z}]+2.5]$	$[\mathbb{E}[\boldsymbol{z}]-5, \mathbb{E}[\boldsymbol{z}]+5]$	$[\mathbb{E}[\boldsymbol{z}]-10, \mathbb{E}[\boldsymbol{z}]+10]$	$[\mathbb{E}[\boldsymbol{z}]-25, \mathbb{E}[\boldsymbol{z}]+25]$
VLCS	80.9	81.1	80.6	80.8	81.2
PACS	87.3	87.5	87.3	87.6	87.6
Office-Home	68.8	68.9	68.6	68.7	68.9

where $\mathbf{W}$ and $\mathbf{b}$ are defined as

$$\mathbf{W} = \begin{bmatrix} 0.5 & 0.1 \\ 2 & -9 \end{bmatrix} \tag{21}$$

$$\mathbf{b} = \begin{bmatrix} 0.3 \\ -0.2 \end{bmatrix}. \tag{22}$$

To compute the interventional expectation, we first derive the expectation of the feature vector $\boldsymbol{z}$, which is calculated as

$$\mathbb{E}[\boldsymbol{z}] = \begin{bmatrix} \mathbb{E}[z^1] \\ \mathbb{E}[z^2] \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0.5+0.7 \\ 1+9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 5 \end{bmatrix} \tag{23}$$

Next, the values of [low, high] for each feature dimension are determined as

$$\begin{aligned} \begin{bmatrix} \text{low}^1 \\ \text{low}^2 \end{bmatrix} &= \begin{bmatrix} \mathbb{E}[z^1] \\ \mathbb{E}[z^2] \end{bmatrix} - 0.5 = \begin{bmatrix} 0.1 \\ 4.5 \end{bmatrix} \\ \begin{bmatrix} \text{high}^1 \\ \text{high}^2 \end{bmatrix} &= \begin{bmatrix} \mathbb{E}[z^1] \\ \mathbb{E}[z^2] \end{bmatrix} + 0.5 = \begin{bmatrix} 1.1 \\ 5.5 \end{bmatrix}. \end{aligned} \tag{24}$$

Accordingly, the features follow uniform distributions $z^1 \sim \mathcal{U}(0.1, 1.1)$, $z^2 \sim \mathcal{U}(4.5, 5.5)$.

We now explain the detailed steps in computing the interventional expectation and baseline for the first sample $\boldsymbol{x}_1$. The interventional expectations can be calculated as

$$\mathbb{E}\left[\boldsymbol{y}|do(z^1 = 0.5)\right] = \int_{\text{low}^2}^{\text{high}^2} \boldsymbol{y} \cdot p(\boldsymbol{y}|do(z^1 = 0.5))dz^2 \tag{25}$$

and the baseline can be calculated as

$$\begin{aligned} &\mathbb{E}_{z^1}\left[\mathbb{E}\left[\boldsymbol{y}|do(z^1 = 0.5)\right]\right] \\ &= \int_{\text{low}^1}^{\text{high}^1} \int_{\text{low}^2}^{\text{high}^2} \boldsymbol{y} \cdot p(\boldsymbol{y}|do(z^1 = 0.5))dz^1 dz^2. \end{aligned} \tag{26}$$

In practice, both expectations are approximated using the sampling approach.

REFERENCES

[1] G. Blanchard, G. Lee, and C. Scott, "Generalizing from several related classification tasks to a new unlabeled sample," *Advances in Neural Information Processing Systems*, vol. 24, pp. 2178–2186, 2011.

[2] V. Vapnik, "Principles of risk minimization for learning theory," in *Advances in Neural Information Processing Systems*, 1992, pp. 831–838.

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

[4] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *European Conference on Computer Vision*. Springer, 2016, pp. 443–450.

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

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

[7] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang, "Adversarial domain adaptation with domain mixup," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6502–6509.

[8] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters, "Invariant models for causal transfer learning," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 1309–1342, 2018.

[9] D. Mahajan, S. Tople, and A. Sharma, "Domain generalization using causal matching," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7313–7324.

[10] A. Chattopadhyay, P. Manupriya, A. Sarkar, and V. N. Balasubramanian, "Neural network attributions: A causal perspective," in *International Conference on Machine Learning*. PMLR, 2019, pp. 981–990.

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

[12] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, "Undoing the damage of dataset bias," in *European Conference on Computer Vision*. Springer, 2012, pp. 158–171.

[13] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[14] M. Ghifary, W. Bastiaan Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2551–2559.

[15] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," *Advances in Neural Information Processing Systems*, vol. 32, pp. 6450–6461, 2019.

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

[17] L. Zhang, X. Wang, D. Yang, T. Sanford, S. Harmon, B. Turkbey, H. Roth, A. Myronenko, D. Xu, and Z. Xu, "When unseen domain generalization is unnecessary? rethinking data augmentation," *arXiv preprint arXiv:1906.03347*, 2019.

[18] O. Nuriel, S. Benaim, and L. Wolf, "Permuted AdaIN: Enhancing the representation of local cues in image classifiers," *arXiv preprint arXiv:2010.05785*, 2020.

[19] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Learning to generate novel domains for domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 561–578.

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

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

[22] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

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

[24] M. M. Rahman, C. Fookes, M. Baktashmotlagh, and S. Sridharan, "Multi-component image translation for deep domain generalization," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 579–588.

[25] F. Qiao, L. Zhao, and X. Peng, "Learning to learn single domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 556–12 565.

[26] L. Li, K. Gao, J. Cao, Z. Huang, Y. Weng, X. Mi, Z. Yu, X. Li, and B. Xia, "Progressive domain expansion network for single domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 224–233.

[27] X. Fan, Q. Wang, J. Ke, F. Yang, B. Gong, and M. Zhou, "Adversarially adaptive normalization for single domain generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8208–8217.

[28] Z. Shen, P. Cui, K. Kuang, B. Li, and P. Chen, "Causally regularized learning with agnostic data selection bias," in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018, pp. 411–419.

[29] J. Peters, P. Bühlmann, and N. Meinshausen, "Causal inference by using invariant prediction: identification and confidence intervals," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

[30] S. Magliacane, T. Van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J. M. Mooij, "Domain adaptation by using causal inference to predict invariant conditional distributions," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[31] A. Subbaswamy, B. Chen, and S. Saria, "A universal hierarchy of shift-stable distributions and the tradeoff between stability and performance," *arXiv preprint arXiv:1905.11374*, 2019.

[32] K. Ahuja, K. Shanmugam, K. Varshney, and A. Dhurandhar, "Invariant risk minimization games," in *International Conference on Machine Learning*. PMLR, 2020, pp. 145–155.

[33] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[34] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, p. e0130140, 2015.

[35] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[36] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[37] J. Pearl, *Causality*. Cambridge University Press, 2009.

[38] M. Kocaoglu, C. Snyder, A. G. Dimakis, and S. Vishwanath, "CausalGAN: Learning causal implicit generative models with adversarial training," *arXiv preprint arXiv:1709.02023*, 2017.

[39] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.

[40] D. P. Vassileios Balntas, Edgar Riba and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proceedings of the British Machine Vision Conference (BMVC)*, E. R. H. Richard C. Wilson and W. A. P. Smith, Eds. BMVA Press, September 2016, pp. 119.1–119.11. [Online]. Available: https://dx.doi.org/10.5244/C.30.119

[41] B. Wang, M. Lapata, and I. Titov, "Meta-learning for domain generalization in semantic parsing," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 366–379.

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

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

[44] S. Shahtalebi, J.-C. Gagnon-Audet, T. Laleh, M. Faramarzi, K. Ahuja, and I. Rish, "SAND-mask: An enhanced gradient masking strategy for the discovery of invariances in domain generalization," *arXiv preprint arXiv:2106.02266*, 2021.

[45] G. Blanchard, A. A. Deshmukh, Ü. Dogan, G. Lee, and C. Scott, "Domain generalization by marginal transfer learning," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 46–100, 2021.

[46] H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo, "Reducing domain gap by reducing style bias," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8690–8699.

[47] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn, "Adaptive risk minimization: Learning to adapt to domain shift," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 664–23 678, 2021.

[48] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, "Out-of-distribution generalization via risk extrapolation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5815–5826.

[49] Z. Huang, H. Wang, E. P. Xing, and D. Huang, "Self-challenging improves cross-domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 124–140.

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

[51] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.

[52] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain adaptive ensemble learning," *IEEE Transactions on Image Processing*, vol. 30, pp. 8008–8018, 2021.

[53] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *International Conference on Learning Representations*, 2020.

[54] B. Li, F. Wu, S.-N. Lim, S. Belongie, and K. Q. Weinberger, "On feature normalization and data augmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 383–12 392.

[55] S. Wang, L. Yu, C. Li, C.-W. Fu, and P.-A. Heng, "Learning from extrinsic and intrinsic supervisions for domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 159–176.

[56] P. Pandey, M. Raman, S. Varambally, and P. AP, "Discrepancy minimization in domain generalization with generative nearest neighbors," *arXiv preprint arXiv:2007.14284*, 2020.