Mixup-Induced Domain Extrapolation for Domain Generalization

Meng Cao[1,2], **Songcan Chen**[1,2*]
[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
{meng.cao, s.chen}@nuaa.edu.cn

Abstract

Domain generalization aims to learn a well-performed classifier on multiple source domains for unseen target domains under domain shift. Domain-invariant representation (DIR) is an intuitive approach and has been of great concern. In practice, since the targets are variant and agnostic, only a few sources are not sufficient to reflect the entire domain population, leading to biased DIR. Derived from PAC-Bayes framework, we provide a novel generalization bound involving the number of domains sampled from the environment (N) and the radius of the Wasserstein ball centred on the target (r), which have rarely been considered before. Herein, we can obtain two natural and significant findings: when N increases, 1) the gap between the source and target sampling environments can be gradually mitigated; 2) the target can be better approximated within the Wasserstein ball. These findings prompt us to collect adequate domains against domain shift. For seeking convenience, we design a novel yet simple ***E**xtrapolation **D**omain* strategy induced by the ***M**ixup* scheme, namely EDM. Through a reverse Mixup scheme to generate the extrapolated domains, combined with the interpolated domains, we expand the interpolation space spanned by the sources, providing more abundant domains to increase sampling intersections to shorten r. Moreover, EDM is easy to implement and be plugged-and-played. In experiments, EDM has been plugged into several methods in both closed and open set settings, achieving up to 5.73% improvement.

Introduction

In conventional classification, the training set and test set generally follow independent identical distribution (i.i.d.) assumption. However, it is impractical in real-world applications due to domain shift (Li et al. 2022), including changing background, style, color, etc. To alleviate this issue, a learning paradigm, namely Domain Generalization (DG), has been presented and received increasing attention. DG aims to induce a well-performed (meta-)classifier from a set of given source domains so that it can generalize to unseen but related target domains.

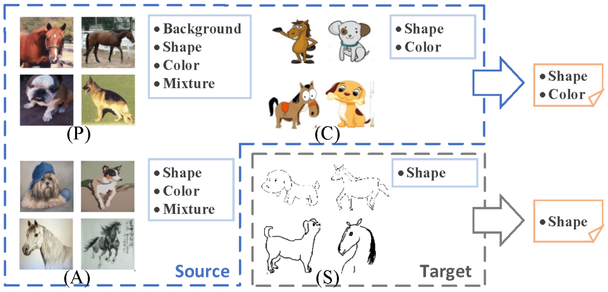


Figure 1: The illustration of biased DIR on PACS dataset, where (P) denotes Photo, (C) denotes Cartoon, (A) denotes Art, and (S) denotes Sketch. The representations provided from each domain are listed roughly in corresponding box.

Up to now, abundant methods have been proposed for DG (Wang et al. 2022). Domain-invariant representation (DIR) (Lu et al. 2022), as one of the dominant approaches, has been widely studied, which can be divided into the following categories: causal inference (Arjovsky et al. 2019), information bottleneck (Li et al. 2022), adversarial learning (Ganin et al. 2016), and others (Ding et al. 2022). These approaches aim to remove the impurity representations, that is domain dependent representations, and to find common representations depending on the downstream task as much as possible. It seems reliable. In practice, since the target domains are variant and agnostic, it is difficult to extract unbiased DIR among domains from limited sample domains that are insufficient to reflect the entire domain population. Taking PACS dataset as a sample, shown in Fig. 1, we can observe that the required representations are inconsistent between the source and target domains. If the sketch domain is selected as the target domain, shape and color will be provided from the source domains, but only the shape is required. In this case, the color is redundant on the entire domain population, which may lead to poor generalization ability. What's more, (Zhu et al. 2022) has discovered that the domains cannot be mixed and can be obviously observed within each class. This phenomenon indicates that the impurity representations remain to some extent and the discrepancy cannot be completely eliminated, which further implies biased DIR with limited domain sampling. Herein, a question arises spontaneously: will the generalization ability be improved as domain sampling increases? Going one step further, what are the factors that influence the generalization ability?

[*]Corresponding Author

To answer this question, we first have to shift our perspective in DG. Almost previous works in DG, to the best of our knowledge, can boil down to a task-oriented framework, which pays attention to the divergence between pairwise domains (Lu et al. 2023), leading to inflexibility in theory. In essence, DG is an inductive learning paradigm on multiple related tasks, which follows a two-step sampling process, namely an environment-task framework (Baxter 2000). In analogy to the standard single-task learning where data is sampled from an unknown distribution, tasks in DG are sampled from an unknown task distribution, i.e., the environment. Herein, these tasks are more commonly referred to as domains in DG. In this way, the gap between environments reflects the similarity between the domain population learned from source domains and that of target domains. And, the gap between tasks reflects the relationship between observed tasks, such as the relationship between animal categorization in two environments. Therefore, compared to the former, the environment-task perspective can be more flexible and not limited to closed-set scenario, such as open-set scenario (Shu et al. 2021).

Following this perspective, we provide a novel generalization bound for DG, derived from PAC-Bayes framework (McAllester 1998), whose key is change of measure inequality. This bound involves the number of domains sampled from the environment N and the radius of the Wasserstein ball centred on the target, which have received little attention. Herein, we can obtain two natural and significant findings: when N increases, 1) the gap between the source and target sampling environments can be gradually reduced; 2) the target can be better approximated within the Wasserstein ball. These findings prompt us to collect adequate domains against domain shift. Indeed, previous works have made efforts to generate varied domain samples through data augmentation. (Shankar et al. 2018) develop an adversarial strategy by reversing the gradient of the domain classifier. (Xiao et al. 2021) generate a new domain through an extra network module. Obviously, these methods are inflexible and computationally expensive. Mixup (Zhang et al. 2017) is another popular and widely used technique. However, the generated samples are usually mixed from pairwise instances and lie in the interpolation space spanned by those instances.

In this manuscript, in search of modeling convenience, we design a novel yet simple ***Extrapolation Domain*** strategy induced by the ***Mixup*** scheme, namely EDM. It is a two-stage Dir-Mixup (Shu et al. 2021) strategy, i.e., extrapolation followed by interpolation. In extrapolation stage, the extrapolated domains are generated through a reverse Mixup scheme. And then, in interpolation stage, the new domains are generated by mixing the generated extrapolated domains. Along this line, the interpolation space spanned by the sources can be expanded, so that the domains can be obtained not only inside but also outside this space, called interpolation domains and extrapolation domains, respectively. In this way, more abundant domains can be provided with unrestricted of the interpolation space, and then the intersections of sampled-domain sets are increased to provide a better target approximation. Moreover, EDM inherits the lightweight and flexible characteristics of Mixup, so that it can be easy to implement and be directly plugged-and-played. To sum up, our contributions are listed as follows:

1. A novel generalization bound for DG is provided, which is flexible for task settings and guides us to pay attention to the number of domains from the environment sampling perspective and the radius of the Wasserstein ball.
2. A two-stage Dir-Mixup strategy, namely EDM, is initially designed to provide more abundant domains, where the extrapolation domains outside the interpolation space can increase the sampling intersections.
3. In experiments, EDM has been plugged into several methods in both closed and open set settings, achieving up to 5.73% improvement.

Related Works

Mainstream Methods for Domain Generalization

DIR aims to find task-dependent but domain-independent representations. Classic moment-based methods align the statistics in representation space, such as MMD (Grubinger et al. 2015), CORAL (Sun and Saenko 2016). The adversarial-based methods, e.g., DANN (Ganin et al. 2016), make an attempt to confuse the domain classifier to remove the domain-related representations, so that the task-related representations can be retained. Unlike DANN, which adds an extra network module, information bottleneck (Li et al. 2022) leverages the information entropy between input and hidden representations and the one between hidden representations and output. To avoid retaining spurious correlations, causal inference has been introduced, where IRM (Albuquerque et al. 2019), a strategy with gradient penalty, is one of the well-known methods. And, VRex (Krueger et al. 2021) provides a variance penalty regularization in loss to obtain invariant representations. Meanwhile, (Sagawa et al. 2020) argue that due to minimizing average loss via empirical risk minimization, spurious correlations arise from typical examples, so that they regroup domains with underlying correlation representations, e,g., background, to avoid learning models that rely on spurious correlations. RSC (Huang et al. 2020) iteratively forces a CNN to activate features that are less dominant in the training domain, but still correlated with labels. In essence, almost of them, especially (Ding et al. 2022), make the effort to remove impurities across domains, which are domain-specific features. In recent years, some researches imply that only a few source domains are insufficient to reflect the entire domain population. For example, (Zhu et al. 2022) discover that at local regions, the domains cannot be mixed and are clustered.

Data augmentation, which is another hot topic in DG, is one of the cheap and simple ways to increase the quality and diversity of the training data. Mixup (Zhang et al. 2017) is a popular technique to achieve this goal. In (Lu et al. 2023), the samples mixed by the same class but different domains are generated to enlarge the diversity, and the samples mixed by the same domain but different classes are generated to reduce the influence of redundant domain information. Meanwhile, (Zhou et al. 2021) generate new samples by mixing the statistical information of paired samples in multiple hidden layers. (Mancini et al. 2020) advocate that mixing samples of both different domains and classes allows to obtain samples that cannot be categorized in a single class and domain of the one available during training, so that they construct some novel semantic-visual samples from triple tuple samples to recognize unseen categories in unseen domains. To further enlarge the diversity of the generated sample, (Shu et al. 2021) proposes a multi-sample mixing strategy, namely Dir-Mixup, whose mixing coefficients sample from Dirichlet distribution. Although the Mixup scheme is flexible, all of them are limited by interpolation space, and the domain distribution cannot be reflected due to the mixing statistics from a single sample. In contrast, a min-max game has been designed to generate new examples, such as CrossGrad (Shankar et al. 2018), which utilizes gradient ascent to expand both class and domain space. (Xiao et al. 2021) add a network module to sample a new domain with a meta-learning learner. Nevertheless, high computational complexity cannot be ignored and its flexibility is not well.

Generalization Bounds for Domain Generalization

In recent years, some generalization bounds for DG have been emerged to demonstrate the effectiveness of corresponding methods. (Albuquerque et al. 2019) provide a generalization bound w.r.t. the specific linear combination of empirical errors from the source domains and the divergence between the real target distribution and the approximate fake target distribution within the space spanned by the source domains. In analogy, (Dai et al. 2023) provide a similar formulation, replacing $\mathcal{H}$-divergence with Wasserstein distribution. And then, (Lu et al. 2023) explicitly provide an additional term, maximizing the divergence across the sources domains, to further reduce loss caused by alignment. These bounds focus on the distribution divergence and motivate DG methods based on DIR. Besides, based on kernel mean embedding, the bound w.r.t. the marginal distribution is given in (Blanchard, Lee, and Scott 2011), which implies that the generalization ability is related to the number of sampled domains. Obviously, this bound is valid for the domain shift depending on the marginal distributions and is difficult to explain methods in deep network.

Methodology

In this section, we introduce our motivation, theoretical framework, and proposed method EDM in detail.

Preliminaries

In DG, a common setting is to provide N domains under domain shift. Let D_n denotes n-th observed domain, which is a set of M_n independent samples from a space of examples $\mathcal{Z}$, i.e., $D_n = \{z_n^m\}_{m=1}^{M_n}$. Each sample is drawn from an unknown distribution $\mathcal{D}_n$, namely $z_n^m \sim \mathcal{D}_n$, and $z_n^m = (x_n^m, y_n^m)$, where x_n^m denotes an input instance and y_n^m denotes the corresponding label. Due to domain shift, $\mathcal{D}_{ni} \neq \mathcal{D}_{nj}, \forall ni \neq nj$. According to the environment-task perspective discussed in Introduction, we argue that these domains are generated i.i.d. from an unknown hyper-distribution τ, i.e., distribution over distribution or $\mathcal{D}_n \sim \tau$. Let $h \in \mathcal{H}$ denotes a hypothesis h belongs to a hypothesis space $\mathcal{H}$. In analogy to the standard single-task learning where a single hypothesis h is learned based on an observed sampling set D, the selected hypothesis h is induced from observed sampling sets (domains) $\{D_n\}_{n=1}^N$. To select an appropriate hypothesis, the PAC-Bayes framework, whose starting point is model average, construct a probability distribution set over $\mathcal{H}$, namely $\mathcal{M}(H)$. That is, $h \sim P$, where $P \in \mathcal{M}(H)$ denotes a probability measure in $\mathcal{M}(H)$, and is described as the prior, which is data-dependent. Based on the observed sampling set D and the prior P, the learner output a posterior distribution Q over $\mathcal{H}$ when learning a new task. Herein, the prior and posterior notations are utilized to describe the relationship between the observed task and the new task, without the need for a likelihood function to connect them. Following the environment-task framework, the above standard PAC-Bayes framework should be extended to adapt to changes in the environment. To this end, we assume a hyper-distribution over the distribution measure space $\mathcal{M}(H)$, e.g., $\mathcal{P} \in \mathcal{M}(P)$, where $\mathcal{P}$ denotes a hyper-prior distribution. $\mathcal{Q}$ is similar and denotes a hyper-posterior distribution. The expected error is denoted as $er(\mathcal{Q}, \tau) \triangleq \mathbb{E}_{Q \sim \mathcal{Q}} er(Q, \tau)$. Since $er(\mathcal{Q}, \tau)$ is not computable, we can evaluate its corresponding empirical error $\hat{er}(\mathcal{Q}, \tau) \triangleq \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{P \sim \mathcal{Q}} \hat{er}(P, D_n)$.

Generalization Bound

Theorem 1 (Domain Generalization Generalization bound)**.** *Giving a hypothesis space $\mathcal{H}$, and N domains $\{D_n\}_{n=1}^N$ sampled from τ, where each domain D_n consists of M_n samples. Let $\mathcal{P}$ denotes a hyper-prior distribution $\mathcal{P} \in \mathcal{M}(P)$, where $P \in \mathcal{M}(\mathcal{H})$ and $\mathcal{M}(\mathcal{S})$ denotes the set of all probability over $\mathcal{S}$. Then, for any $\delta \in (0, 1]$, the following inequality holds uniformly for all hyper-posterior distributions $\mathcal{Q}$ with probablity at least $1 - \delta$:*

$$\begin{aligned} \mathbb{E}_{Q \sim \mathcal{Q}} er(Q, \tau) \quad \leq \quad & \tfrac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{P \sim \mathcal{Q}} \hat{er}(P, D_n) \\ & + \sqrt{\tfrac{L_0 \cdot W_1(\mathcal{Q}, \mathcal{P}) + \ln(N/\delta)}{2(N-1)}} \\ & + \tfrac{1}{N} \sum_{n=i}^{N} \sqrt{\tfrac{L_n \cdot W_1(Q, P_n) + \ln(N M_n/\delta)}{2(M_n - 1)}} \end{aligned} \tag{1}$$

where $W_1(\cdot, \cdot)$ is the 1st order Wasserstein Distance, and L_0 and L_n are the Lipschitz constants. Its corresponding proof is provided in Appendix in detail.

From Theorem 1, we can find that the generalization error of DG is bounded by the empirical error from the source domains plus two complexity terms. The first complexity term is a so-called environment-complexity term, which measures the gap between environments $W_1(\mathcal{Q}, \mathcal{P})$, where environments sample the source and target domains, respectively. This gap is caused by observing only a finite number of tasks. Meanwhile, the second complexity term is an average task complexity term, which measures the divergence of tasks between target domain and each source domain $W_1(Q, P_n)$. For a clearer explanation, due to the domains following a hyper-distribution, we can further assume that the divergence between environments $W_1(\mathcal{Q}, \mathcal{P})$ can be relaxed to the radius of the Wasserstein ball centred on the target r, that is $W_1(\mathcal{Q}, \mathcal{P}) \leq r$. To this end, upper bound of generalization error is with respect to N, M_n, r and $W_1(Q, P_n)$. When each M_n approaches to infinity, the task complexity term will converge to zero, and when N approaches to infinity, both the task and environment complexity term will converge to zero. These findings are very natural and intuitive, prompting us to collect adequate observed samples for each domain while collecting adequate domains. In this way, the domain population learned from the observed source domains can be a better approximation of the entire population, ensuring good performance on a novel task, i.e., the task of target domain. Moreover, according to Theorem 3.4 in (Mohajerin Esfahani and Kuhn 2018), the radius r will be inversely proportional to the number of observed domains N, if the unknown distribution is light-tailed in the sense. Therefore, through increasing sampling domains, the target can be better represented on domain population, analogous to the situation with test examples on the training distribution. In summary, increasing the sampling domains is another way to improve the generalization ability of DG, similar to increasing the sampling samples from each domain. By the way, due to loose assumptions for tasks, our generalization bound is very flexible and can explain multiple settings through the gap of tasks $W_1(Q, P_n)$, such as open-set setting.

Extrapolation Domain Induced by Mixup

We know that it's unrealistic to collect an infinite number of domains. As an alternative, we design a novel yet simple ***E**xtrapolation **D**omain* strategy induced by the ***M**ixup* scheme, namely EDM. Compared to previous methods (Zhou et al. 2021; Shu et al. 2021; Lu et al. 2023), most of which mix paired samples, EDM has two significant characteristics: 1) mixing the statistics from multiple source domains; 2) constructing an extrapolation space surrounding the interpolation space spanned by source domains.

The reason behind the former is intuitive, that is, our aim is to augment domains rather than samples. Moreover, the mixing strategy in classic methods boils down to a linear interpolation strategy, which only generates new domains between two domains (the lines between vertices as shown in Fig. 2). This pairwise mixing strategy is obviously limited by the lack of domains mixed from multiple domains, i.e., the whole blue area.

The reason behind the latter is that due to the finite observed domains and the variant and agnostic target domain, as shown in Fig. 2, the interpolation space or the environment obtained from the source domains i.e., the blue area, may be biased. To mitigate this issue, inspired by $W_1(\mathcal{Q}, \mathcal{P}) \leq r$, where $r \propto \exp(1/N)$ if P_n follows a light-tailed sampling, we would like to expand the interpolation space to increase the intersections with the domains sampling from the target environment, i.e. the green area, in order to satisfy the sampling assumption as much as possible. In this way, not only more abundant domains can be generated further, but also theoretical generalization ability can be guaranteed to some extent.

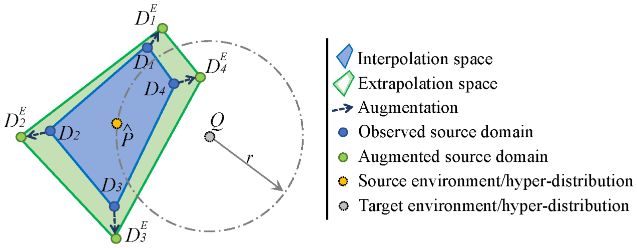


Figure 2: The illustration of EDM. Through Mixup scheme, each domain D_n is pushed outward to generate the extrapolation space, represented by the light green region, based on the corresponding new domain D_n^E.

To realize EDM, similar to Dir-Mixup (Shu et al. 2021), we formulate the foundation of EDM, i.e., the multiple domain mixing scheme, as follows:

$$\mathcal{D}_{\boldsymbol{\lambda}} = \sum_{n=1}^{N} \lambda_n \mathcal{D}_n, \forall \lambda_n \geq 0 \text{ and } \sum_{n=1}^{N} \lambda_n = 1 \tag{2}$$

where λ_n denotes the mixing coefficient of n-th domain. Unlike classic Mixup scheme, where the mixing coefficient is sampled from Beta Distribution, $\boldsymbol{\lambda}$ are sampled from Dirichlet Distribution parameterized by a parameter $\boldsymbol{\alpha}$, i.e., $\boldsymbol{\lambda} \sim \text{Dirichlet}(\boldsymbol{\alpha})$.

For a general case, we assume that $\mathcal{D}_n \triangleq \mathcal{N}(\mu_n, \sigma_n^2)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian Distribution with the mean μ and the standard deviation σ. Therefore, Eq. (2) can be reformulated as:

$$\mathcal{N}(\mu_{\boldsymbol{\lambda}}, \sigma_{\boldsymbol{\lambda}}^2) = \mathcal{N}\left(\sum_{n=1}^{N} \lambda_n \mu_n, \sum_{n=1}^{N} \lambda_n^2 \sigma_n^2\right) \tag{3}$$

And, each pair of parameters (μ_n, σ_n^2) can be calculated based on the corresponding training data $x_n^m \in \mathbb{R}^{C \times H \times W}$ in each batch, formulated as:

$$\begin{aligned} \mu_n &= \tfrac{1}{B_n HW} \textstyle\sum_{m=1}^{B_n} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_n^m)_{h,w} \\ \sigma_n^2 &= \tfrac{1}{B_n HW} \textstyle\sum_{m=1}^{B_n} \sum_{h=1}^{H} \sum_{w=1}^{W} \left((x_n^m)_{h,w} - \mu_n\right)^2 \end{aligned} \tag{4}$$

where B_n is the number of training data on n-th domain.

To avoid wasting information, a momentum strategy is adopted, which utilizes historical information through a moving average weight ρ. Then, we have

$$\mu_n^t = \rho \mu_n^{t-1} + (1 - \rho)\mu_n, \; \sigma_n^t = \rho \sigma_n^{t-1} + (1 - \rho)\sigma_n \tag{5}$$

where μ_n^{t-1} denotes the mean of n-th domain at $(t-1)$-th iteration, and σ_n^{t-1} is similar.

Combining Eq. (3) and Eq. (5), we can obtain the statistics of a new domain. It is noted that these new domains only lie in the interpolation space spanned by observed source domains, which is not our intention.

To generate extrapolated domains, we reverse the above multiple domain mixing scheme to obtain the corresponding supported domains for the extrapolation space. For example,

D_1^E is a supported domain corresponding to D_1 in Fig. 2. Specifically, each domain D_n can be regarded as an interpolated domain through mixing the corresponding extrapolated domain D_n^E and the other source domains, respectively. In other words, similar to Eq. (2), we have:

$$\mathcal{D}_n = \lambda_n \mathcal{D}_n^E + \sum_{i=1, i\neq n}^{N} \lambda_i \mathcal{D}_i, \forall \lambda_i \geq 0 \text{ and } \sum_{i=1}^{N} \lambda_i = 1 \quad (6)$$

In this way, $\mathcal{D}_n^E$ can be reformulated as:

$$\mathcal{D}_n^E = \frac{1}{\lambda_n}\mathcal{D}_n - \sum_{i=1, i\neq n}^{N} \frac{\lambda_i}{\lambda_n}\mathcal{D}_i \quad (7)$$

where λ_n has the same constraint as Eq. (6). And, its Gaussian Distribution can be referred to as

$$\begin{aligned}&\mathcal{N}\left(\left(\mu_{\boldsymbol{\lambda}}^E\right)_n, \left(\left(\sigma_{\boldsymbol{\lambda}}^E\right)_n\right)^2\right) \\ &= \mathcal{N}\left(\tfrac{1}{\lambda_n}\mu_n - \sum_{i=1, i\neq n}^{N} \tfrac{\lambda_i}{\lambda_n}\mu_i, \tfrac{1}{\lambda_n^2}\sigma_n^2 + \sum_{i=1, i\neq n}^{N} \left(\tfrac{\lambda_i}{\lambda_n}\right)^2 \sigma_i^2\right)\end{aligned} \quad (8)$$

Next, based on these supported domains outside the interpolation space, we once again employ the multiple domain mixing scheme, similar to Eq. (3), to generate the new domains, which will be located inside and outside the interpolation space spanned by the observed source domains $\{D_n\}_{n=1}^N$, as the green and blue areas in Fig. 2.

Note that this twice Mixup scheme, i.e., Eq. (8) followed by Eq. (3) employed with the corresponding supported domains, is one of the augmentation schemes to indirectly obtain the extrapolation domains. We can also select more domains with the positive coefficient in Eq. (7) to directly obtain the extrapolation domains. Obviously, it is too complex and difficult to control.

Finally, through AdaIN scheme (Zhou et al. 2021), the samples sampling from new domains can be represented as:

$$x_a^m = \frac{x_n^m - \mu_n}{\sigma_n}\sigma_a + \mu_a \quad (9)$$

where μ_a and σ_a denote the mean and standard deviation of augmented domains through twice Mixup scheme, respectively. These generated samples are directly fed to the training model along with the original samples. And, their class labels are the same as the corresponding original samples, and a new domain label will be assigned. Detailed Algorithm is provided in Appendix.

Experiments

In this section, extensive experiments are constructed to comprehensively evaluate the effectiveness of EDM on two datasets both in closed and open set settings.

Datasets and Settings

For the architecture, we use ResNet-18 as backbone on three datasets, i.e., PACS, Office-Home, and DomainNet datasets. For both settings, we follow corresponding settings from the previous methods, i.e., the same closed-set setting as (Lu et al. 2022), and the same open-set setting as (Shu et al. 2021). In closed-set setting, we compare with twelve recent strong comparison methods and two other representative methods. Except ERM, they can be divided into four categories: 1) domain-invariant representation based methods DANN (Ganin et al. 2016), MMD (Grubinger et al. 2015), CORAL (Sun and Saenko 2016), VREx (Krueger et al. 2021), DIFEX (Lu et al. 2022); 2) data augmentation based method Mixup (Zhang et al. 2017), CrossGrad (Shankar et al. 2018), MixStyle (Zhou et al. 2021); 3) learning robust features based methods: GroupDRO (Sagawa et al. 2020), RSC (Huang et al. 2020); 4) model optimization based method ANDMask (Parascandolo et al. 2020), SAGM (Wang et al. 2023). And in open-set setting, following (Shu et al. 2021), we compare with seven other popular methods, which are divided into the following categories: 1) data augmentation based method CuMix (Mancini et al. 2020); 2) learning robust features based methods PAR (Wang et al. 2019), RSC (Huang et al. 2020); 3) heterogeneous method FC (Li et al. 2019b); 4) meta-learning based methods MLDG (Li et al. 2018), Epi-FCR(Li et al. 2019a), DAML (Shu et al. 2021). For more details on datasets, comparison methods, and settings, please refer to Appendix. The code is available at https://github.com/Alrash/EDM.

Results and Analysis

Tab. 1 reports accuracy results in closed-set setting, and Tab. 2 reports accuracy and H-score (Fu et al. 2020) results in open-set setting.

From **Tab. 1**, we can observe two key findings as follows: 1) compared with previous methods, the learner + EDM can give the best results, achieving up to 5.73$\%$ improvement in the sketch on PACS; 2) cross-compared with similar methods with high complexity, EDM has shown its lightweight and flexible characteristics.

Specifically, the former key finding can be reflected in the following aspects. First, on both datasets, the learner attaching EDM can achieve the best results on average. Second, in each domain on both datasets, the best and second results can almost be obtained through attaching EDM. These phenomena can testify to the effectiveness of EDM. Third, the learner + Inter, i.e., attaching augmented interpolation domains, is usually slightly weaker than the learner + EDM, i.e., attaching both augmented interpolation and extrapolation domains, but is better than the corresponding basic learner. This fact indicates that domain augmentation is beneficial and can improve the generalization ability for DG. And, additional extrapolation space, which reflects the more complete domain population combined with interpolation space, can further improve the performance. Fourth, different basic learners have received different gains in each domain. The results in the sketch on PACS and in the clipart on Office-Home have received significant improvement, especially for DANN + EDM, which achieves up to 5.73$\%$ and 2.09$\%$, respectively. And, the second improvements are in cartoon and product, respectively. These phenomena further indicate that EDM can simulate more severe drift and make the learner perform well in scenarios with significant domain shift. Fifth, compared with SAGM, there is no sig-

	PACS					Office-Home					DNet
	Art-Painting	Cartoon	Photo	Sketch	Avg	Art	Clipart	Product	Real-World	Avg	Avg
ERM	81.10	77.94	95.03	76.94	82.75	57.77	50.63	71.30	74.45	63.54	40.10
DANN	82.86	78.33	96.11	76.99	83.57	57.60	48.52	71.16	72.99	62.57	40.20
Mixup	81.84	75.43	95.27	76.51	82.26	58.71	51.00	72.20	75.42	64.33	39.24
RSC	82.13	77.99	94.43	79.87	83.60	57.07	50.77	71.93	73.63	63.35	37.13
MMD	80.32	76.45	92.46	**83.63**	83.21	59.29	50.52	72.34	74.43	64.15	39.14
CORAL	79.39	77.90	91.98	82.03	82.83	59.29	50.15	72.25	74.20	63.97	39.16
GroupDRO	79.15	76.75	91.32	81.52	82.19	59.09	50.22	71.91	74.48	63.92	31.78
CrossGrad‡	80.37	74.87	96.59	74.98	81.70	58.67	51.18	71.66	74.80	64.08	39.49
Mixstyle‡	82.51	79.09	95.65	79.23	84.12	55.50	51.00	70.62	73.19	62.57	39.98
ANDMask	80.81	73.29	95.81	71.95	80.47	53.61	47.54	69.36	72.23	60.69	23.92
VREx	81.54	78.11	95.39	80.35	83.85	59.09	49.81	71.64	74.82	63.84	37.96
DIFEX-ori†	82.86	78.46	94.97	79.41	83.93	57.89	50.82	71.61	73.40	63.43	38.33
DIFEX-norm†	83.40	79.74	95.03	79.10	84.32	58.09	51.50	72.08	73.62	63.82	38.53
SAGM‡	82.62	78.50	96.05	79.64	84.20	59.13	51.23	72.67	75.90	64.73	38.86
ERM + Inter	83.25	77.60	95.99	81.09	84.48	58.01	50.65	72.02	74.62	63.83	-
DANN + Inter	81.93	77.82	95.99	81.57	84.33	57.27	50.13	71.53	74.04	63.24	-
Mixup + Inter	83.01	76.32	**96.65**	78.04	83.50	**59.46**	52.30	72.88	75.60	65.06	-
SAGM + Inter	82.28	79.01	96.29	80.22	84.45	58.55	**52.71**	72.85	75.51	64.91	-
ERM + EDM	82.32	79.27	96.53	81.24	84.84	58.67	51.84	72.38	75.35	64.56	40.34
DANN + EDM	82.96	78.07	96.47	82.72	85.06	58.51	50.61	72.22	74.59	63.98	**40.40**
Mixup + EDM	**83.50**	79.14	96.59	81.04	85.07	59.33	51.94	**73.15**	75.97	**65.10**	40.04
SAGM + EDM	82.47	**80.38**	96.59	80.86	**85.08**	58.84	52.33	72.94	**76.06**	65.04	39.23

Table 1: Accuracy results on PACS, Office-Home and DNet (DomainNet) in closed-set settings. + Inter denotes that the method attaches augmented interpolation domains. + EDM denotes that the method attaches augmented interpolation and extrapolation domains. The bold and underline items are the best and the second-best results, respectively. ‡ denotes our reproduced results on PACS and Office-Home, and † denotes our reproduced results on Office-Home. All results on DomainNet are reproduced.

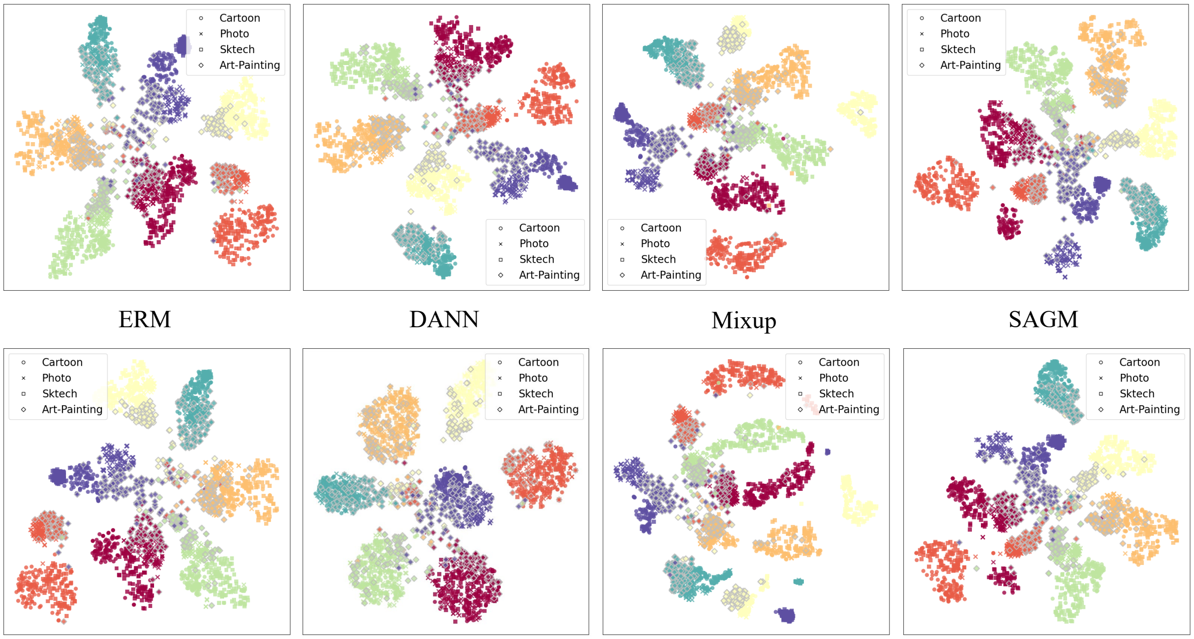
	PACS		Office-Home	
	Acc	H-score	Acc	H-score
ERM	55.17	44.78	50.43	47.41
MLDG	57.43	45.00	51.07	47.58
FC	58.13	46.69	51.03	48.02
Epi-FCR	60.64	48.47	50.25	48.48
PAR	56.56	44.95	51.26	49.03
RSC	58.92	45.05	49.56	47.89
CuMix	57.85	41.05	51.67	49.40
DAML	65.49	51.88	56.45	53.34
DAML + Inter	69.22	51.83	59.15	53.64
DAML + EDM	**70.78**	**54.12**	**59.58**	**54.19**

Table 2: Accuracy and H-score results both on PACS and Office-Home datasets in open-set settings.

nificant improvement with SAGM + Inter or SAGM + EDM. We think that the model perturbation mechanism can indirectly simulate the drift between domains so that the effect of EDM has been counteracted to some extent.

And, the latter key finding can be reflected in three aspects. First, Mixup + EDM can achieve better performance than Mixup, the results of ERM + Inter and ERM + EDM are almost better than Mixup while the gains on ERM and Mixup are approximately similar. These facts imply that domain augmentation and data augmentation can be parallel to each other to improve the generalization ability, and theoretical analysis can be empirically testified simultaneously. Second, compared with DIFEX-ori and DIFEX-norm, as a representative of domain-invariant representation based methods, DANN + EDM can be slightly better. Third, compared with CrossGrad, which contains data augmentation and domain augmentation, the results of ERM + EDM are almost better, not to mention those of Mixup + EDM. These phenomena demonstrate the convenience of EDM, which does not require extra networks or tasks (regularizers).

From **Tab. 2**, we can observe the following findings. First, although DAML is the SOTA in open-set setting, in which augmented samples are utilized, its performance can also be improved when augmented domains are attached. Second, DAML + EDM achieves the best results. This fact indicates that extrapolation domains should be considered and can further improve performance. Third, significant improvement in accuracy results both of DAML + Inter and DAML + EDM can be observed, but their improvement in H-score results is a little bit inferior. The reason is that our proposed domain augmentation strategy does not directly solve the issue of unknown class detection in open-set setting. In fact, new domains containing random classes are further generated through Inter or EDM. Therefore, the learner can capture more complete known category information to boost performance through CE loss of mixup samples in DAML. For more detailed experimental results and corresponding analysis of Tab. 2, please refer to Appendix.



ERM DANN Mixup SAGM

ERM + EDM DANN + EDM Mixup + EDM SAGM + EDM

Figure 3: Visualization of the t-SNE embeddings of learned representation spaces for PACS with different methods. Different colors correspond to different classes and different shapes correspond to different domains. Note that the diamond with the grey edge denote the target domain.

In our experiments, **ablation study** is equivalent to whether adding new domains can improve the performance of the corresponding method. These results can be found both in Tab. 1 and Tab. 2, so we will no longer report them. Overall, adding new domains can improve performance, and combining interpolation and extrapolation domains can achieve better performance. More details can be referred to in the aforementioned analysis.

From **Fig. 3**, we can discover that the methods belonging to different categories exhibit different phenomena. DANN, as a domain-invariant representation based method, aims to obtain a representation space in which the domains can be confused in each class. However, the domains can be obviously observed and each class cluster seems not tight. In contrast, in DANN + EDM, the domains can be more scattered within each class and each class cluster can be tighter. These phenomena reveal that a few domains only receive biased domain-invariant representations, which contain undesired domain-dependent yet task-dependent representations. Domain augmentation, which mixes domain information rather than class information, can alleviate this issue to some extent. ERM and Mixup, as representative of aggregation methods, have not paid too much attention to domain information. Therefore, the domains exhibit a certain degree of clustering within each class. With EDM, the discriminative ability of tasks has not been harmed, and the classes in target domain prefer to classify the classes on a more similar source domain, such as the photo. For example, compared to ERM, the orange and the blue in ERM + EDM are closed to the corresponding classes in the photo, and the domains are more scattered within each class. These phenomena also imply that domain augmentation can rich domain information, and obtain more essential representations even on non-domain-invariant representation based methods. Finally, compared to SAGM, a model optimization based method, EDM as a data perturbation strategy does not compromise the model perturbation mechanism. The tightness of each class has ups and downs on both sides.

Conclusion

Domain generalization (DG) is regarded as an inductive learning paradigm on multiple related tasks, which belongs to an environment-task framework. Following this perspective, we give a novel generalization bound for DG, derived from PAC-Bayes framework. In light of this bound, we argue that the factors that influence the generalization ability involve four aspects. In this manuscript, we focus on two factors: the number of observed domains and the gap between sampling environments, which have received little attention in previous methods. After relaxing this gap to the radius of a Wasserstein ball centred on the target, we discover once again that increasing the sampling domains can improve the generalization ability. To this end, we design a novel yet simple Extrapolation Domain strategy induced by the Mixup scheme, namely EDM, which indirectly constructs an extrapolation space surrounding the interpolation space spanned by source domains to provide more abundant domains. In addition, EDM is easy to implement and can be plugged and played. Finally, extensive experiments are conducted to testify to the effectiveness of EDM.

Acknowledgments

This work was supported by the Natural Science Foundation of China (NSFC) (Grant No.62076124), Scientific and technological achievements transferring project of Jiangsu Province (BA2021005).

References

Albuquerque, I.; Monteiro, J.; Falk, T. H.; and Mitliagkas, I. 2019. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 8.

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

Baxter, J. 2000. A model of inductive bias learning. *Journal of artificial intelligence research*, 12: 149–198.

Blanchard, G.; Lee, G.; and Scott, C. 2011. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in neural information processing systems*, volume 24.

Dai, R.; Zhang, Y.; Fang, Z.; Han, B.; and Tian, X. 2023. Moderately Distributional Exploration for Domain Generalization. In *International Conference on Learning Representations*.

Ding, Y.; Wang, L.; Liang, B.; Liang, S.; Wang, Y.; and Chen, F. 2022. Domain generalization by learning and removing domain-specific features. In *Advances in Neural Information Processing Systems*, volume 35, 24226–24239.

Fu, B.; Cao, Z.; Long, M.; and Wang, J. 2020. Learning to detect open classes for universal domain adaptation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, 567–583. Springer.

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

Grubinger, T.; Birlutiu, A.; Schöner, H.; Natschläger, T.; and Heskes, T. 2015. Domain generalization based on transfer component analysis. In *Advances in Computational Intelligence: 13th International Work-Conference on Artificial Neural Networks, IWANN 2015, Palma de Mallorca, Spain, June 10-12, 2015. Proceedings, Part I 13*, 325–334. Springer.

Huang, Z.; Wang, H.; Xing, E. P.; and Huang, D. 2020. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 124–140. Springer.

Krueger, D.; Caballero, E.; Jacobsen, J.-H.; Zhang, A.; Binas, J.; Zhang, D.; Le Priol, R.; and Courville, A. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, 5815–5826. PMLR.

Li, B.; Shen, Y.; Wang, Y.; Zhu, W.; Li, D.; Keutzer, K.; and Zhao, H. 2022. Invariant information bottleneck for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7399–7407.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. 2018. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

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

Li, Y.; Yang, Y.; Zhou, W.; and Hospedales, T. 2019b. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, 3915–3924. PMLR.

Lu, W.; Wang, J.; Li, H.; Chen, Y.; and Xie, X. 2022. Domain-invariant Feature Exploration for Domain Generalization. *Transactions on Machine Learning Research*.

Lu, W.; Wang, J.; Wang, Y.; and Xie, X. 2023. Towards Optimization and Model Selection for Domain Generalization: A Mixup-guided Solution. In *The KDD'23 Workshop on Causal Discovery, Prediction and Decision*, 75–97. PMLR.

Mancini, M.; Akata, Z.; Ricci, E.; and Caputo, B. 2020. Towards recognizing unseen categories in unseen domains. In *European Conference on Computer Vision*, 466–483. Springer.

McAllester, D. A. 1998. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, 230–234.

Mohajerin Esfahani, P.; and Kuhn, D. 2018. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2): 115–166.

Parascandolo, G.; Neitz, A.; ORVIETO, A.; Gresele, L.; and Schölkopf, B. 2020. Learning explanations that are hard to vary. In *International Conference on Learning Representations*.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*.

Shankar, S.; Piratla, V.; Chakrabarti, S.; Chaudhuri, S.; Jyothi, P.; and Sarawagi, S. 2018. Generalizing Across Domains via Cross-Gradient Training. In *International Conference on Learning Representations*.

Shu, Y.; Cao, Z.; Wang, C.; Wang, J.; and Long, M. 2021. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9624–9633.

Sun, B.; and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, 443–450. Springer.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, volume 32.

Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; and Yu, P. 2022. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, P.; Zhang, Z.; Lei, Z.; and Zhang, L. 2023. Sharpness-aware gradient matching for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3769–3778.

Xiao, Z.; Shen, J.; Zhen, X.; Shao, L.; and Snoek, C. 2021. A bit more bayesian: Domain-invariant learning with uncertainty. In *International Conference on Machine Learning*, 11351–11361. PMLR.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021. Mixstyle neural networks for domain generalization and adaptation. *arXiv preprint arXiv:2107.02053*.

Zhu, W.; Lu, L.; Xiao, J.; Han, M.; Luo, J.; and Harrison, A. P. 2022. Localized adversarial domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7108–7118.