Lost Domain Generalization Is a Natural Consequence of Lack of Training Domains

Yimu Wang[1], **Yihan Wu**[2], **Hongyang Zhang**[1]

[1]University of Waterloo
[2]University of Maryland, College Park
{yimu.wang,hongyang.zhang}@uwaterloo.ca, ywu42@umd.edu

Abstract

We show a hardness result for the number of training domains required to achieve a small population error in the test domain. Although many domain generalization algorithms have been developed under various domain-invariance assumptions, there is significant evidence to indicate that out-of-distribution (o.o.d.) test accuracy of state-of-the-art o.o.d. algorithms is on par with empirical risk minimization and random guess on the domain generalization benchmarks such as DomainBed. In this work, we analyze its cause and attribute the lost domain generalization to the lack of training domains. We show that, in a minimax lower bound fashion, *any* learning algorithm that outputs a classifier with an ϵ excess error to the Bayes optimal classifier requires at least $\mathrm{poly}(1/\epsilon)$ number of training domains, even though the number of training data sampled from each training domain is large. Experiments on the DomainBed benchmark demonstrate that o.o.d. test accuracy is monotonically increasing as the number of training domains increases. Our result sheds light on the intrinsic hardness of domain generalization and suggests benchmarking o.o.d. algorithms by the datasets with a sufficient number of training domains.

Introduction

Domain generalization (Mahajan, Tople, and Sharma 2021; Wang et al. 2020; Dou et al. 2019; Yang et al. 2021; Bui et al. 2021; Robey, Pappas, and Hassani 2021; Wald et al. 2021; Recht et al. 2019; Wang et al. 2023; Wang, Shi, and Zhang 2023)—where the training distribution is different from the test distribution—has been a central research topic in machine learning (Blanchard et al. 2021; Chuang, Torralba, and Jegelka 2020; Zhou et al. 2021; Wu et al. 2023a,b), computer vision (Piratla, Netrapalli, and Sarawagi 2020; Gan, Yang, and Gong 2016; Huang et al. 2021; Song et al. 2019; Taori et al. 2020; Wu, Zhang, and Huang 2022; Wu, Huang, and Zhang 2023), and natural language processing (Wang, Lapata, and Titov 2021; Fried, Kitaev, and Klein 2019). In machine learning, the study of domain generalization has led to significant advances in the development of new algorithms for out-of-distribution (o.o.d.) generalization (Li et al. 2022; Bitterwolf et al. 2022; Thulasidasan et al. 2021). In computer vision and natural language processing, new benchmarks such as DomainBed (Gulrajani and Lopez-Paz 2021) and WILDs (Koh et al. 2021; Sagawa et al. 2021) are built toward closing the gap between the developed methodology and real-world deployment. In both cases, the problem can be stated as given a set of training domains $\{P_e\}_{e=1}^E$ which are drawn from a domain distribution $\mathcal{P}$ and given a set of training data $\{(\mathbf{x}_i^e, y_i^e)\}_{i=1}^n$ which are drawn from P_e, the goal is to develop an algorithm based on the training data and their domain labels e so that the algorithm in expectation performs well on the unseen test domains drawn from $\mathcal{P}$.

Despite progress on domain generalization, many fundamental questions remain unresolved. For example, in search of lost domain generalization, Gulrajani and Lopez-Paz (2021) conducted extensive experiments using DomainBed and found that, when carefully implemented, empirical risk minimization (ERM) shows state-of-the-art performance across all datasets despite many algorithms being carefully designed for the out-of-distribution tasks. For example, when the algorithm is trained on the "$+90\%$" (the degree of correlation between color and label) and "$+80\%$" domains of the ColoredMNIST dataset (Arjovsky et al. 2019) and is tested on the "-90%" domain, the best-known o.o.d. algorithm achieves test accuracy no better than a random-guess algorithm under all three model selection methods in Gulrajani and Lopez-Paz (2021). So, it is natural to ask what causes the lost domain generalization and how to find it?

In this paper, we attribute the lost domain generalization to the lack of training domains. Our study is motivated by an observation that off-the-shelf benchmarks often suffer from few training domains. For example, the number of training domains in DomainBed (Gulrajani and Lopez-Paz 2021) for all its 7 datasets is at most 6; in WILDs (Koh et al. 2021; Sagawa et al. 2021), 7 out of 10 datasets have the number of training domains fewer than 350 (see Table 1). Therefore, one may conjecture that increasing the number of training domains might improve the empirical performance of existing domain generalization algorithms significantly. In this paper, we show that, information-theoretically, one requires at least $\mathrm{poly}(1/\epsilon^2)$ number of training domains in order to achieve a small excess error ϵ for any learning algorithm. This is in sharp contrast to many existing benchmarks in which the number of training domains is limited.

WILDs	iWildCam	Camelyon17	RxRx1	OGB-MolPCBA	BlobalWheat	CicilComments	FMoW
# domains	323	5	51	120,084	47	16	80

DomainBed	CMNIST	RMNIST	VLCS	PACS	Office-Home	Terra Incognita	DomainNet
# domains	3	6	4	4	4	4	6

Table 1: The number of domains in the o.o.d. benchmarks WILDs (Koh et al. 2021; Sagawa et al. 2021) and DomainBed (Gulrajani and Lopez-Paz 2021). It shows that most of the datasets in the two benchmarks suffer from a small number of domains, which might not be sufficient to learn a classifier with good domain generalization.

Related Work

Out-of-distribution (o.o.d) generalization (Hendrycks and Dietterich 2019; Shankar et al. 2018; Zhou et al. 2021) has received extensive attention in recent years. One representative way is the causal modeling inspired by Invariant Risk Minimization (IRM) (Arjovsky et al. 2019). IRM tries to learn an invariant feature representation to capture the underlying causal mechanism of interest across domains such that the classifier based on this invariant feature representation shall be invariant across all domains. Given multiple training domains, IRM learns invariant representations approximately by adding a regularization. The results of IRM indicate that failing to generalize to o.o.d. data comes from failing to capture the causal factors of variation in different domains. Following IRM, Risk Extrapolation (REx) (Krueger et al. 2021) proposes to reduce differences in risk across training domains. Derivative Invariant Risk Minimization (DIRM) (Bellot and van der Schaar 2020) maintains the invariance of the gradient of training risks across different domains.

Another line of research uses different metrics to tackle the o.o.d problem. For example, Maximum Mean Discrepancy-Adversarial AutoEncoder (Li et al. 2018b) employs Generative Adversarial Networks and the maximum mean discrepancy metric (Gretton et al. 2012) to align different feature distributions. Mixture of Multiple Latent Domains (Matsuura and Harada 2020) learns domain-invariant features by clustering techniques without knowing which domain the training samples belong to. Recently, Meta-Learning Domain generalization (Li et al. 2020) employs a lifelong learning method to tackle the sequential problem of new incoming domains.

To explore the o.o.d problem, one line of research focuses on the case where only one training domain is accessible. Causal Semantic Generative model (CSG) (Liu et al. 2021) uses two sets of correlated latent variables, *i.e.*, the semantic and non-semantic features, to model the relation between the data and the corresponding labels. In their assumption, the semantic features relate the data to their corresponding labels while the non-semantic features only affect the generation of data. CSG decouples the semantic and non-semantic features to improve o.o.d generalization given only one training domain. Another related line of research (Ben-David et al. 2010; Zhao et al. 2019; Wang et al. 2022) might be analyzing the domain complexity for unsupervised domain adaption, which is a sub-problem of domain generalization. They mainly focus on analyzing the performance of learning algorithms on another distribution (domain), while our analysis targets a broader scenario, *i.e.*, the performance of learning algorithms on all distributions (domains).

However, recent work (Gulrajani and Lopez-Paz 2021) claims that all existing algorithms cannot capture the true invariant feature and observes that their performance is on par with ERM and random guess on several datasets. In this paper, to explain why it occurs, we theoretically analyze the o.o.d. generalization problem and provide a minimax lower bound for the number of training domains required to achieve a small population error in the test domain. Massart and Nédélec (2006) have proved that it requires at least $\Omega(1/\epsilon^2)$ samples from a distribution to estimate the success probability of a Bernoulli variable with an ϵ error. Motivated by this, we observe a similar phenomenon and prove that the learning algorithms need at least $\Omega(1/\epsilon^2)$ number of training domains. Recently, a concurrent work (Li, Gouk, and Hospedales 2022) presents an upper bound on the expected excess error of the ERM algorithm using the Rademacher complexity. Similarly, another work (Blanchard et al. 2021) gives an upper bound on the excess error of general learning algorithms with high probability and shows that the sample size of each domain is inversely proportional to the excess error. On the other side, while Li, Gouk, and Hospedales (2022); Blanchard et al. (2021) showed positive results on the domain generalization, we present a negative result (*i.e.*, a lower bound regarding the number of training domains) on the expected excess error for all possible learning algorithms.

Minimax Lower Bound for Domain Generalization

In this section, we provide a minimax lower bound for domain generalization. Our results lower bound the number of training domains required for good o.o.d. generalization.

Notation. We will use *bold capital* letters such as $\mathbf{X}$ to represent a random vector, *bold lower-case* letters such as $\mathbf{x}$ to represent the implementation of a random vector, capital letters such as Y to represent a random variable, and lower-case letters such as y to represent the implementation of a random variable. Specifically, we denote by $\mathbf{X}$ the random vector of instance, denote by $\mathbf{x}$ the implementation of random vector $\mathbf{X}$, denote by Y the random variable of the label, and denote by $y \in \{0, 1\}$ the implementation of a random variable Y. We will use $L(f)$ to represent the expected 0-1 loss of classifier f w.r.t. the mixture of data distributions of all domains, *i.e.*, $L(f) = \Pr_{(\mathbf{X},Y)}(f(\mathbf{X}) \neq Y)$. Throughout the paper, we will frequently use $\mathcal{P}$ and $\mathcal{D}$ to represent a set of distributions, and the distribution of distribution, *i.e.*,

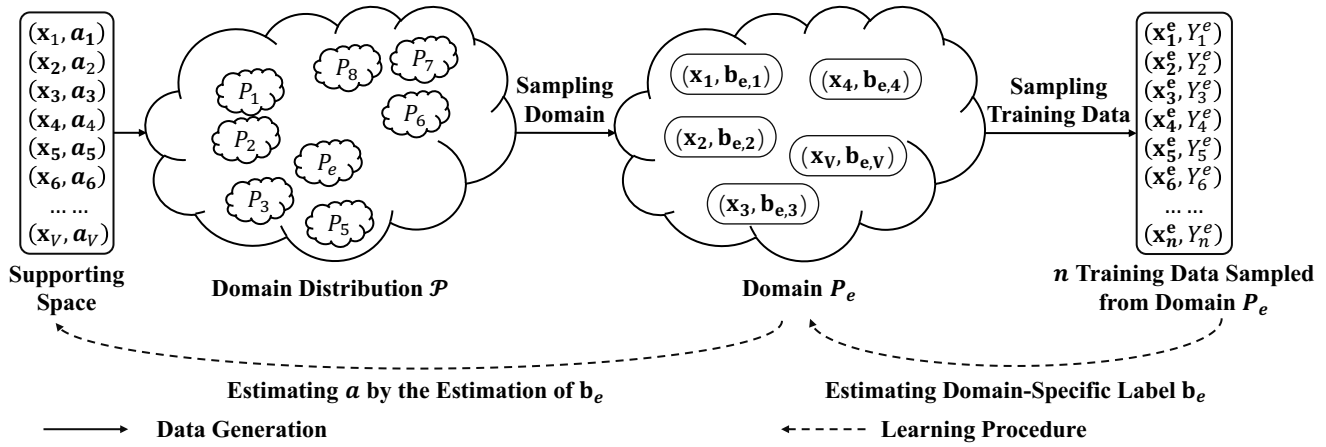


Figure 1: Illustration of our o.o.d. generalization problem. We show how data is sampled and how learning algorithms learn the knowledge. When generating training data, E domains from the data distribution $\mathcal{D}$ are first sampled, and after that for each domain, n training data are sampled to form the training dataset which will be fed into a learning algorithm. The learning algorithm recovers the underlying label α by the estimation of the underlying label $\mathbf{a}^e, e \in [E]$ under the observation of training data.

the domain distribution, will use P_e to represent the data distribution of the e-th domain, and will use $(\mathbf{x}^e, y^e)$ to represent the data sampled from the e-th domain P_e. We call $e \in \{1, 2, 3, ...\}$ the domain labels, which are accessible to the learner.

Problem setups. In our hard instance, we view the e-th domain as a data distribution P_e given by $\Pr(\mathbf{X}, Y|\mathbf{A}^e = \mathbf{a}^e)$, where e is the domain label and $\mathbf{A}^e$'s represent i.i.d. Bernoulli random vectors that parameterize the data distribution of the e-th domain. In this paper, we will regard $\Pr(\mathbf{X}, Y|\mathbf{A}^{e_1})$ and $\Pr(\mathbf{X}, Y|\mathbf{A}^{e_2})$ as two different domains as long as $e_1 \neq e_2$. We assume that each domain is sampled from a domain distribution $\mathcal{D}$ (*i.e.*, the distribution of $\mathbf{A}^e$), and the data in the e-th domain are sampled from a data distribution $\mathcal{D}^e$ given by $\Pr(\mathbf{X}, Y|\mathbf{A}^e = \mathbf{a}^e)$. All the domains from $\mathcal{D}$ share the same supporting space containing m data points. Let f^* be the Bayes optimal classifier of the mixture of data distributions across all domains, and assume $f^* \in \mathcal{F}$, where $\mathcal{F}$ can be any function class such as deep neural networks. For any $h \in [0, 1]$, we define a class of domain distributions by $\mathcal{P}(h, \mathcal{F}) := \{P : |2\Pr(\mathbf{A}^e = 1) - 1| \geq h\}$. Note that the margin parameter h determines the randomness of the domain: large h (e.g., $h = 1$) means $\Pr(\mathbf{A}^e = 1)$ is bounded away from $1/2$. We will investigate the following minimax risk:

$$R_{E,n}(h, \mathcal{F}) := \inf_{\widetilde{f}_{E,n} \in \mathcal{F}} \sup_{\mathcal{P}} \mathbf{E}_{P_e \sim \mathcal{P}} \mathbf{E}_{(\mathbf{X}^e, Y^e) \sim P_e} \left[L(\widetilde{f}_{E,n}) - L(f^*) \right], \quad (1)$$

where E is the number of training domains, n is the number of training samples from each domain, and the two expectations are taken over the sampling of training data and domains to learn $\widetilde{f}_{E,n}$. The minimax problem in Equation (1) characterizes the access risk of the best learning algorithm with access to E training domains and n data samples under the worst-case domain distribution.

Let V be the VC dimension of $\mathcal{F}$, which is defined as the maximum number of points that can be arranged so that $\mathcal{F}$ shatters them. Our main results are as follows:

Theorem 1. *For $n = \infty$, $V \geq m$, any $h \in [0, 1]$, and any $E \geq V$, we have the lower bound*

$$R_{E,\infty}(h, \mathcal{F}) \geq c \min \left(\frac{m-1}{Eh}, \sqrt{\frac{m-1}{E}} \right), \quad (2)$$

where $c > 0$ is an absolute constant.

We defer the proofs of Theorem 1 to the Appendix . The theorem provides a lower bound on the number of training domains required to achieve a small population error, even though one can sample as many data points as possible from each domain. The case of $n = \infty$ captures the "easiest" case for the learner, where the learning algorithm can access full knowledge about each training domain. The case of finite n is harder than $n = \infty$, as the learner has only partial knowledge about each training domain and $R_{E,n}(h, \mathcal{F}) \geq R_{E,\infty}(h, \mathcal{F})$. Therefore, *Equation* (2) *provides a universal lower bound for general $n \geq 1$.* Theorem 1 implies that, information-theoretically, one requires at least $\mathrm{poly}(V/\epsilon)$ number of training domains in order to achieve a small excess error ϵ for any learning algorithm of $\mathcal{F}$. This is in sharp contrast to many existing benchmarks on which the number of training domains is limited (see Table 1). For example, in the celebrated ColoredMNIST dataset (Arjovsky et al. 2019), there are only 2 training domains. When the algorithm is trained on the "$+90\%$" and "$+80\%$" domains and is tested on the "-90%" domain, the best-known o.o.d. algorithm achieves test accuracy no better than random guess under all three model selection methods in Gulrajani and Lopez-Paz (2021). Theorem 1 predicts the failures of future algorithms on these datasets and attributes the poor performance of existing o.o.d. algorithms to the lack of training domains.

Differences between our work and Massart and Nédélec (2006). The major differences include: 1) the construction of the hard instance, *i.e.*, a two-stage data generative procedure (Section), and 2) the strategy of splitting the hard problem into two sub-problems (see Figure 1). These two aspects are original and separate our contributions from previous works. For 1), our data generative model first samples E domains from the domain distribution $\mathcal{D}$ by generating the domain-specific label $\mathbf{a}^e, \forall e \in [E]$ and then samples the training data from each sampled domain. On the other hand, Massart and Nédélec (2006) considered a totally different scenario: they investigated the effect of training sample size on the excess risk in the single-domain problem when the training and test data are i.i.d. For 2), our proof has to deal with two expectations given that we have designed a novel two-stage recovery strategy. Our two-stage problem splits the hard problem into two simpler problems which estimate a binary string α and $\mathbf{a}^e, \forall e \in [E]$, while Massart and Nédélec (2006) only considered one binary string estimation problem. The two binary string estimation problems are entangled, making our analysis more challenging.

Experiments

Theorem 1 shows that *any* learning algorithm that outputs a classifier with an ϵ excess error to the Bayes optimal classifier requires at least $\mathrm{poly}(1/\epsilon)$ number of training domains, even though the number of training data sampled from each training domain is large. In this section, we complement our theoretical results with an empirical study to evaluate the impact of the number of training domains. Although the datasets we use, ColoredMNIST and RotatedMNIST, may not necessarily adhere to the Bernoulli distribution utilized in our theoretical results, the empirical results are still useful for supporting our theoretical results, since our lower bounds consider the worst-case distributions.

Datasets

We conducted extensive experiments on two datasets from DomainBed, *i.e.*, ColoredMNIST (Arjovsky et al. 2019) and RotatedMNIST (Ghifary et al. 2015). We notice that there are other popular domain generalization datasets, *e.g.*, PACS (Li et al. 2017), VLCS (Fang, Xu, and Rockmore 2013), Office-Home (Venkateswara et al. 2017), and Terra Incognita (Beery, Horn, and Perona 2018). However, these datasets are hard to generate more training domains synthetically as their data generation process cannot be parameterized by a single variable (*e.g.*, correlation between color and label in ColoredMNIST, or rotation degree in RotatedMNIST). Thus, in our paper, we do not consider these datasets.

ColoredMNIST (Arjovsky et al. 2019) is a variant of the MNIST hand written digit classification dataset (LeCun et al. 1998). It is a synthetic dataset containing three domains $p_e \in [0.1, 0.2, 0.9]$ colored either red or blue formalizing $70,000$ examples of dimension $(2, 28, 28)$ and 2 classes. The label is a noisy function of the digit and color, such that color bears correlation p_E with the label and the digit bears correlation 0.75 with the label. Inspired by the protocol introduced in DomainBed, we randomly split the training dataset into 10 sub-datasets with equal training samples. Each domain of ColoredMNIST is generated as follows: 1) Assign a preliminary binary label y' to the image based on the digit: $y' = 0$ for digits $0-4$ and $y' = 1$ for $5-9$; 2) Obtain the final label y by flipping y' with probability 0.25; 3) Sample the color id z by flipping y with probability p_e; 4) Color the image red if $z = 1$ or green if $z = 0$. The only parameter of a training domain is p_e. We use the domain with $p_e = 0.5$ as the test domain and uniformly sample E parameters p_e from $(0, 1)/\{0.5\}$ to form E training domains.

RotatedMNIST (Ghifary et al. 2015) is another variant of MNIST with 6 domains containing digits rotated by $\{0, 15, 30, 45, 60, 75\}$ degrees. It contains $70,000$ examples of dimension $(1, 28, 28)$ and 10 classes. Similar to ColoredMNIST, we use the domain with 45 degrees rotation as the test domain and uniformly sample E rotation degrees from $[0, 90)/\{45\}$ to form E training domains.

Algorithms and Evaluation Settings

Algorithms. To validate our theoretical results, we evaluate the effect of the number of training domains on o.o.d. algorithms, including ERM (Vapnik 1991), IRM (Arjovsky et al. 2019), GroupDRO (Sagawa et al. 2020), Mixup (Xu et al. 2020), MLDG (Li et al. 2018a), CORAL (Sun and Saenko 2016), MMD (Li et al. 2018b), DANN (Ganin et al. 2016), and C-DANN (Li et al. 2018c). As we can not evaluate the empirical performance of all the possible algorithms, we randomly use 9 algorithms from DomainBed (Gulrajani and Lopez-Paz 2021). The details of the algorithms are shown in the Appendix. For each algorithm, we employ the default hyper-parameter introduced in Section D.2 of DomainBed, as our goal is not to show the best performance of algorithms but to show the correlations to our theoretical results. Following DomainBed, we use MUNIT (Table 4 in the Appendix) for ColoredMNIST and RotatedMNIST.

Model Evaluation. We train models using 9 different Domain Generalization algorithms, with a varying number of training domains on ColoredMNIST and RotatedMNIST. Each trial is done with 5 different random seeds, and we present the average results. We use the code repository of DomainBed with PyTorch (Paszke et al. 2019). Following DomainBed, we employ and adapt three different model selection methods for training algorithms. The details of the three model selection methods are shown in the Appendix.

Experimental Results on ColoredMNIST and RotatedMNIST

We first introduce the average results on two different datasets using 9 algorithms with the number of training domains varying from 2 to 192 and then present the results with a limited number of domains. Due to the limitation of space, we present the most important results in our paper while leaving the left results in the Appendix.

Evaluating the effect of the number of training domains Results. We run the experiments on ColoredMNIST and RotatedMNIST with ERM, IRM, GroupDRO, Mixup, MLDG, CORAL, MMD, DANN, and C-DANN while the number of training domains varies from 2 to 192. The average accuracy

\#	ERM	IRM	GroupDRO	Mixup	MLDG	CORAL
4	0.6697±0.0120	0.5500±0.0091	0.6710±0.0134	0.6081±0.0144	0.6744±0.0046	0.6586±0.0139
6	0.7135±0.0027	0.5910±0.0072	0.7158±0.0013	0.6703±0.0088	0.7107±0.0035	0.7141±0.0020
8	0.7183±0.0018	0.6278±0.0031	0.7199±0.0016	0.7129±0.0017	0.7195±0.0005	0.7199±0.0013
10	0.7226±0.0005	0.6685±0.0086	0.7220±0.0004	0.7159±0.0006	0.7271±0.0011	0.7228±0.0004
12	0.7280±0.0011	0.6968±0.0034	0.7288±0.0012	0.7223±0.0015	0.7287±0.0008	0.7278±0.0010
14	0.7289±0.0016	0.6709±0.0123	0.7284±0.0016	0.7215±0.0007	0.7316±0.0015	0.7285±0.0015
16	0.7268±0.0011	0.6777±0.0055	0.7272±0.0010	0.7230±0.0014	0.7322±0.0008	0.7258±0.0010
18	0.7304±0.0017	0.7031±0.0045	0.7292±0.0018	0.7255±0.0015	0.7338±0.0009	0.7297±0.0008
20	0.7305±0.0018	0.6957±0.0069	0.7321±0.0011	0.7239±0.0010	0.7336±0.0008	0.7311±0.0013
22	0.7323±0.0011	0.6935±0.0078	0.7298±0.0010	0.7276±0.0012	0.7368±0.0014	0.7296±0.0011
24	0.7330±0.0015	0.6908±0.0086	0.7358±0.0009	0.7269±0.0014	0.7366±0.0012	0.7354±0.0012
26	0.7350±0.0019	0.6995±0.0026	0.7343±0.0016	0.7323±0.0013	0.7366±0.0011	0.7353±0.0015
28	0.7336±0.0016	0.6997±0.0076	0.7347±0.0014	0.7327±0.0011	0.7370±0.0013	0.7332±0.0014
30	0.7331±0.0023	0.7113±0.0027	0.7326±0.0023	0.7297±0.0020	0.7391±0.0012	0.7329±0.0020
48	0.7386±0.0014	0.7219±0.0007	0.7398±0.0015	0.7352±0.0014	0.7410±0.0010	0.7385±0.0014
96	0.7427±0.0014	0.7182±0.0015	0.7424±0.0013	0.7399±0.0012	0.7444±0.0011	0.7424±0.0014
192	0.7437±0.0014	0.7287±0.0012	0.7443±0.0014	0.7424±0.0013	0.7461±0.0010	0.7437±0.0014

Table 2: The experimental results on ColoredMNIST with ERM, IRM, GroupDRO, Mixup, MLDG, and CORAL w.r.t the number of training domains using the training-domain validation set model selection method.

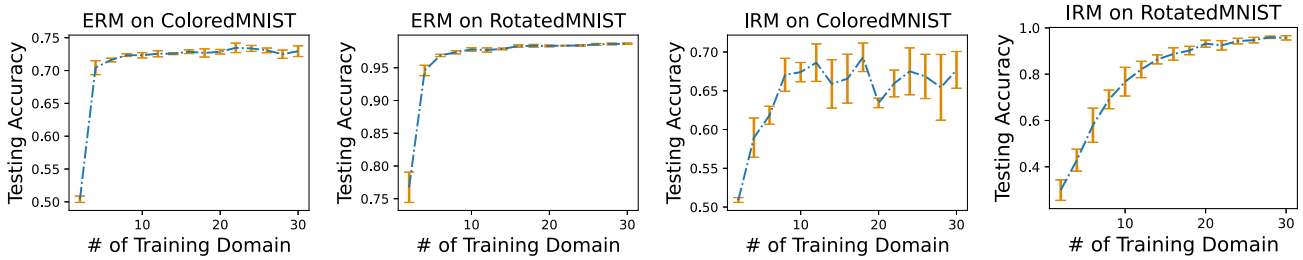


Figure 2: The experimental results on ColoredMNIST and RotatedMINST using ERM and IRM w.r.t the number of training domains using the leave-one-domain-out cross-validation method.

w.r.t the number of training domains is shown in Tables 2 and 3 in the main paper, Tables 6 and 8 and Figures 4 and 5 in the Appendix. It shows that the test accuracy is proportional to the number of training domains with all the algorithms on both ColoredMNIST and RotatedMNIST which is consistent with our theoretical results (Theorem 1).

Training-domain validation set analysis. The results are shown in Tables 2 and 6 and Figure 4. We observe that the test accuracy of almost all the algorithms on both ColoredMNIST and RotatedMNIST is monotonically increasing as the number of training domains grows while the accuracy of IRM on both datasets, MMD on ColoredMNIST, DANN, and CDANN on RotatedMNIST experiences slight drops for a certain number of training domains. We also find that the standard deviations of MMD are quite big which might be due to the hyperparameter setting as we did not try to tune the hyperparameters to gain the best performance. Besides, the standard deviations of all the algorithms in the first experiments (the least number of training domains) are quite large. That is because the number of training domains is limited and the algorithms are hard to capture general patterns. As the number of training domains grows, the standard deviations of almost all the algorithms decrease.

Test-domain validation set (oracle) analysis. Figure 5 and Tables 3 and 8 show the results using the oracle model selection method. Similar observations can be obtained. The accuracy of ERM, DANN, CDANN, CORAL, GroupDRO, and Mixup on ColoredMNIST and RotatedMNIST is pro-

\#	ERM	IRM	GroupDRO	Mixup	MLDG	CORAL
4	0.6697±0.0120	0.5517±0.0085	0.6710±0.0134	0.6081±0.0144	0.6750±0.0046	0.6586±0.0139
6	0.7138±0.0027	0.5915±0.0073	0.7158±0.0013	0.6703±0.0088	0.7133±0.0038	0.7141±0.0020
8	0.7203±0.0014	0.6278±0.0031	0.7205±0.0017	0.7129±0.0017	0.7213±0.0009	0.7209±0.0010
10	0.7244±0.0012	0.6685±0.0086	0.7236±0.0012	0.7159±0.0006	0.7271±0.0011	0.7244±0.0012
12	0.7284±0.0009	0.6968±0.0034	0.7288±0.0012	0.7224±0.0015	0.7302±0.0010	0.7280±0.0010
14	0.7291±0.0017	0.6709±0.0123	0.7284±0.0016	0.7216±0.0007	0.7317±0.0015	0.7286±0.0015
16	0.7274±0.0011	0.6777±0.0055	0.7274±0.0010	0.7230±0.0013	0.7326±0.0010	0.7265±0.0011
18	0.7314±0.0012	0.7031±0.0045	0.7311±0.0012	0.7260±0.0016	0.7343±0.0008	0.7307±0.0010
20	0.7311±0.0015	0.6958±0.0069	0.7321±0.0011	0.7259±0.0010	0.7341±0.0010	0.7313±0.0012
22	0.7323±0.0011	0.6935±0.0078	0.7305±0.0012	0.7278±0.0011	0.7371±0.0014	0.7306±0.0014
24	0.7357±0.0013	0.6908±0.0086	0.7358±0.0009	0.7281±0.0018	0.7372±0.0012	0.7354±0.0012
26	0.7351±0.0018	0.6995±0.0026	0.7345±0.0015	0.7323±0.0013	0.7368±0.0011	0.7353±0.0015
28	0.7341±0.0015	0.6997±0.0076	0.7351±0.0014	0.7331±0.0011	0.7370±0.0013	0.7338±0.0012
30	0.7333±0.0023	0.7113±0.0027	0.7338±0.0024	0.7300±0.0018	0.7396±0.0014	0.7334±0.0021
48	0.7390±0.0012	0.7219±0.0007	0.7398±0.0015	0.7362±0.0012	0.7415±0.0009	0.7390±0.0014
96	0.7432±0.0014	0.7182±0.0015	0.7425±0.0013	0.7401±0.0011	0.7446±0.0011	0.7427±0.0014
192	0.7439±0.0012	0.7287±0.0012	0.7448±0.0012	0.7426±0.0014	0.7468±0.0010	0.7439±0.0013

Table 3: The experimental results on ColoredMNIST with ERM, IRM, GroupDRO, Mixup, MLDG, and CORAL w.r.t the number of training domains using the test-domain validation set (oracle) model selection method.

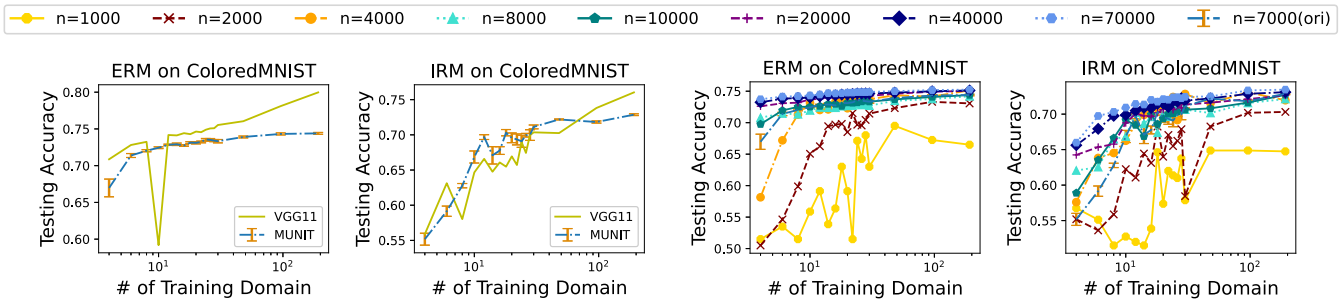
portional to the number of training domains while there are fluctuations in the lines of MMD and IRM on both datasets, which might be due to the fact that MMD and IRM are sensitive to the hyperparameters as we did not tune the hyperparameters for the best performance. The line of IRM on RotatedMNIST drops slightly when the number of training domains is over 100. That might be caused by the limited number of training images n in our theorem. In that case, algorithms might not be able to extract general patterns and might learn biased information, which causes the performance drop. Besides, as we only conduct 5 trials for each experiment, the randomness of the experiments might also be another reason why the performance of IRM on RotatedMNIST drops slightly. Overall, the results using the test-domain validation set and training-domain validation set model selection methods are the same, which supports our theoretical results.

Leave-one-domain-out cross-validation analysis. As the leave-one-domain-out cross-validation requires huge computational resources, we only conduct the experiments with the number of training domains from 2 to 30 with a step of 2. The results are shown in fig. 2 in the main paper, fig. 6 and tables 9 and 10 in the Appendix. Observed from the two tables and the figure, we conclude that the test accuracy of most algorithms is proportional to the number of training domains, while there are some exceptions, *e.g.*, IRM on ColoredMNIST and GroupDRO on ColoredMNIST. For all the results on RotatedMNIST, we observe that the results perfectly match our theoretical results even without any hyper-parameter tuning, especially for the experiments on IRM. But we still can observe exactly the same fact that without any hyper-parameter tuning, the test accuracy of IRM on RotatedMNIST grows with the increase of the number of training domains.

Ablation Study

Analysis on different architectures. To test how our theoretical results generalize to other architecture of neural networks, we further conduct experiments on ColoredMNIST with MUNIT (Table 4 in the Appendix) and VGG11 (Simonyan and Zisserman 2014) with the oracle model selection method. We use the learning rate of $5e-5$ while the remaining other hyparameters are the same. The corresponding results are shown in Figure 3(a) in the main paper, Figure 7(a) and tables 11 to 14 in the Appendix. A similar conclusion can be summarized that the test accuracy still grows with the increase of the number of training domains while using a totally different architecture. But we observe more fluctuation in line with VGG11 as we only conduct 1 trial for VGG11. There is a big "valley" around $E = 100$ in the experiments of ERM on ColoredMNIST, which is quite unusual as it is too big compared with other fluctuations.

Analysis on the number n of data from each domain. To testify the effect of the number of data from each domain, we conduct experiments on ColoredMNIST using ERM and IRM with n from 1000 to 20000 and the oracle model selection method, while the original n is set to be 7000. The experimental results are shown in Figure 3(b) in the main paper,



(a) Different architectures

(b) Different number of data

Figure 3: The experimental results on ColoredMNIST using ERM and IRM, w.r.t the number of training domains with the oracle model selection method. The left two figures show the results with different architectures, *i.e.*, MUNIT and VGG11 (Simonyan and Zisserman 2014), while the right two figures present the corresponding results with a different number of n.

Figure 7(b), and Tables 15 to 18 in the Appendix. It shows that when n is relatively small, especially when $n = 1000$, the line of the accuracy experiences lots of fluctuations. The randomness may be the biggest reason as we only conducted one trial for the ablation study, while we still observe that, the test accuracy is "overall" proportional to the number of training domains. When $n \geq 2000$, the line of test accuracy is absolutely proportional to the number of training domains, which fits our theoretical results well. But that also arises a question, what is the minimum requirement on n for a similar theoretical result?

Discussion on $E < V$. Under this assumption, the lower bound on the excess error might be higher than the current results (Theorem 1). But we might be able to have a similar conclusion with our theoretical result. Experimental results shown in Figure 2 in the main paper and Table 9, Table 10 in the Appendix indicate that, even when the number of domains (less than 30) is relatively small compared with the dimension of the training data, the performance is still proportional to the number of training domain E in the most of cases, which supports our theoretical results.

Conclusion

In this paper, we investigated the out-of-distribution problem and analyzed how many training domains were required to achieve a small population error in the test domain under reasonable assumptions. Our results theoretically characterized the phenomenon of the lost domain generalization which had been found by Gulrajani and Lopez-Paz (2021) in 2021. And our work showed that in a minimax lower bound fashion, *any* learning algorithm with an ϵ excess error to the Bayes optimal classifier required at least $\mathrm{poly}(1/\epsilon)$ number of training domains, even when the number of training data sampled from each training domain was large. There were strong correlations between our work and some empirical results (Arjovsky et al. 2019; Liu et al. 2021; Krueger et al. 2021) in the o.o.d area. Besides, though we used Bernoulli (discrete) random variables to present our theoretical results, our lower bounds hold true for the broader distribution class as we look at the worst-case distributions.

To complement our theoretical results, we conduct experiments on two typical o.o.d benchmarks, *i.e.*, ColoredMNIST and RotatedMNIST, with 9 different o.o.d methods, showing that for the methods used in this paper, the test accuracy on the test domain was proportional to the number of training domains under three different model selection methods. That matched our theoretical results perfectly and indicated us that it is possible to increase the number of domains to improve the domain generalization ability of algorithms practically.

There are several future directions for our work. Our theorem assumed that the number of data samples n from each domain was ∞. This assumption was used to lower bound the case of general n because intuitively, the case of $n = \infty$ should be simpler than the case where n is a finite number. It is interesting to understand how n affects a tight minimax lower bound. Another future direction is to explore the case where the numbers of samples from each domain are different. It would be interesting to see which domain dominates the training procedure and how to design o.o.d training algorithms under this scenario. Moreover, in our case, the instance support (feature space) was shared across domains. Another case we should consider is that each domain only has its own instance support. This domain shift is frequently observed in real-world scenarios and it would help us understand the o.o.d problem further. Besides, we would also like to explore the upper bound of o.o.d problems to see whether our lower bound results match the upper bound. Next, exploring the relationship between the theoretical analysis of domain adaptation and domain generalization would be inspiring. Last, though multi-class classification can be seen as a combination of multiple binary classification problems (*e.g.*, one-vs.-rest classifier), it is interesting to extend our results to the multi-classification problem.

References

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

Beery, S.; Horn, G. V.; and Perona, P. 2018. Recognition

in Terra Incognita. In *European Conference on Computer Vision*, Lecture Notes in Computer Science, 472–489.

Bellot, A.; and van der Schaar, M. 2020. Accounting for Unobserved Confounding in Domain Generalization. *arXiv preprint arXiv:2007.10653*.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A theory of learning from different domains. *Mach. Learn.*, 79(1-2): 151–175.

Bitterwolf, J.; Meinke, A.; Augustin, M.; and Hein, M. 2022. Revisiting Out-of-Distribution Detection: A Simple Baseline is Surprisingly Effective.

Blanchard, G.; Deshmukh, A. A.; Dogan, U.; Lee, G.; and Scott, C. 2021. Domain Generalization by Marginal Transfer Learning. *Journal of Machine Learning Research*, 22(2): 1–55.

Bui, M.; Tran, T.; Tran, A.; and Phung, D. Q. 2021. Exploiting Domain-Specific Features to Enhance Domain Generalization. In *Advances in Neural Information Processing Systems*, 21189–21201.

Chuang, C.; Torralba, A.; and Jegelka, S. 2020. Estimating Generalization under Distribution Shifts via Domain-Invariant Representations. In *International Conference on Machine Learning*, 1984–1994.

Dou, Q.; de Castro, D. C.; Kamnitsas, K.; and Glocker, B. 2019. Domain Generalization via Model-Agnostic Learning of Semantic Features. In *Advances in Neural Information Processing Systems*, 6447–6458.

Fang, C.; Xu, Y.; and Rockmore, D. N. 2013. Unbiased Metric Learning: On the Utilization of Multiple Datasets and Web Images for Softening Bias. In *IEEE International Conference on Computer Vision*, 1657–1664. Sydney, Australia.

Fried, D.; Kitaev, N.; and Klein, D. 2019. Cross-Domain Generalization of Neural Constituency Parsers. In *Annual Meeting of the Association for Computational Linguistics*, 323–330.

Gan, C.; Yang, T.; and Gong, B. 2016. Learning Attributes Equals Multi-Source Domain Generalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 87–97.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; and Lempitsky, V. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(59): 1–35.

Ghifary, M.; Kleijn, W. B.; Zhang, M.; and Balduzzi, D. 2015. Domain Generalization for Object Recognition with Multi-task Autoencoders. In *IEEE International Conference on Computer Vision*, 2551–2559.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25): 723–773.

Gulrajani, I.; and Lopez-Paz, D. 2021. In Search of Lost Domain Generalization. In *International Conference on Learning Representations*.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*.

Huang, J.; Guan, D.; Xiao, A.; and Lu, S. 2021. FSDR: Frequency Space Domain Randomization for Domain Generalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 6891–6902.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; Lee, T.; David, E.; Stavness, I.; Guo, W.; Earnshaw, B.; Haque, I.; Beery, S. M.; Leskovec, J.; Kundaje, A.; Pierson, E.; Levine, S.; Finn, C.; and Liang, P. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *International Conference on Machine Learning*, 5637–5664.

Krueger, D.; Caballero, E.; Jacobsen, J.-H.; Zhang, A.; Binas, J.; Zhang, D.; Priol, R. L.; and Courville, A. 2021. Out-of-Distribution Generalization via Risk Extrapolation (REx). In *International Conference on Machine Learning*, 5815–5826.

LeCam, L. 1973. Convergence of estimates under dimensionality restrictions. *The Annals of Statistics*, 38–53.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; and others. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, D.; Gouk, H.; and Hospedales, T. 2022. Finding lost DG: Explaining domain generalization via model complexity. *arXiv preprint arXiv:2202.00563*.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. 2018a. Learning to Generalize: Meta-Learning for Domain Generalization. In *AAAI Conference on Artificial Intelligence*.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. 2020. Sequential learning for domain generalization. In *European Conference on Computer Vision*, 603–619. Springer.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, Broader and Artier Domain Generalization. In *IEEE International Conference on Computer Vision*, 5543–5551. Venice, Italy.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018b. Domain Generalization with Adversarial Feature Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5400–5409.

Li, X.; Dai, Y.; Ge, Y.; Liu, J.; Shan, Y.; and DUAN, L. 2022. Uncertainty Modeling for Out-of-Distribution Generalization. In *International Conference on Learning Representations*.

Li, Y.; Tian, X.; Gong, M.; Liu, Y.; Liu, T.; Zhang, K.; and Tao, D. 2018c. Deep Domain Generalization via Conditional Invariant Adversarial Networks. In *European Conference on Computer Vision*, 647–663.

Liu, C.; Sun, X.; Wang, J.; Tang, H.; Li, T.; Qin, T.; Chen, W.; and Liu, T.-Y. 2021. Learning Causal Semantic Representation for Out-of-Distribution Prediction. In *Advances in Neural Information Processing Systems*, 6155–6170.

Mahajan, D.; Tople, S.; and Sharma, A. 2021. Domain generalization using causal matching. In *International Conference on Machine Learning*, 7313–7324.

Massart, P.; and Nédélec, É. 2006. Risk Bounds for Statistical Learning. *The Annals of Statistics*, 34(5): 2326–2366.

Matsuura, T.; and Harada, T. 2020. Domain Generalization Using a Mixture of Multiple Latent Domains. In *AAAI Conference on Artificial Intelligence*, 11749–11756.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, 8024–8035.

Piratla, V.; Netrapalli, P.; and Sarawagi, S. 2020. Efficient Domain Generalization via Common-Specific Low-Rank Decomposition. In *International Conference on Machine Learning*, 7728–7738.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do ImageNet Classifiers Generalize to ImageNet? In *International Conference on Machine Learning*, 5389–5400.

Robey, A.; Pappas, G. J.; and Hassani, H. 2021. Model-Based Domain Generalization. In *Advances in Neural Information Processing Systems*, 20210–20229.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*.

Sagawa, S.; Koh, P. W.; Lee, T.; Gao, I.; Xie, S. M.; Shen, K.; Kumar, A.; Hu, W.; Yasunaga, M.; Marklund, H.; Beery, S.; David, E.; Stavness, I.; Guo, W.; Leskovec, J.; Saenko, K.; Hashimoto, T.; Levine, S.; Finn, C.; and Liang, P. 2021. Extending the WILDS Benchmark for Unsupervised Adaptation. 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*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2019. Improving the Generalization of Adversarial Training with Domain Adaptation. In *International Conference on Learning Representations*.

Sun, B.; and Saenko, K. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *European Conference on Computer Vision*, Lecture Notes in Computer Science, 443–450.

Taori, R.; Dave, A.; Shankar, V.; Carlini, N.; Recht, B.; and Schmidt, L. 2020. Measuring Robustness to Natural Distribution Shifts in Image Classification. In *Advances in Neural Information Processing Systems*, 18583–18599.

Thulasidasan, S.; Thapa, S.; Dhaubhadel, S.; Chennupati, G.; Bhattacharya, T.; and Bilmes, J. 2021. A Simple and Effective Baseline for Out-of-Distribution Detection using Abstention.

Vapnik, V. 1991. Principles of Risk Minimization for Learning Theory. In *Advances in Neural Information Processing Systems*.

Venkateswara, H.; Eusebio, J.; Chakraborty, S.; and Panchanathan, S. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5385–5394.

Wald, Y.; Feder, A.; Greenfeld, D.; and Shalit, U. 2021. On Calibration and Out-of-Domain Generalization. In *Advances in Neural Information Processing Systems*, 2215–2227.

Wang, B.; Lapata, M.; and Titov, I. 2021. Meta-Learning for Domain Generalization in Semantic Parsing. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 366–379.

Wang, H.; Si, H.; Li, B.; and Zhao, H. 2022. Provable Domain Generalization via Invariant-Feature Subspace Recovery. In *International Conference on Machine Learning*, 23018–23033.

Wang, Y.; Shi, P.; and Zhang, H. 2023. Gradient-Based Word Substitution for Obstinate Adversarial Examples Generation in Language Models. arXiv:2307.12507.

Wang, Y.; Song, R.-J.; Wei, X.-S.; and Zhang, L. 2020. An Adversarial Domain Adaptation Network For Cross-Domain Fine-Grained Recognition. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1217–1225.

Wang, Y.; Zhang, D.; Wu, Y.; Huang, H.; and Zhang, H. 2023. Cooperation or Competition: Avoiding Player Domination for Multi-Target Robustness via Adaptive Budgets. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20564–20574.

Wu, X.; Sun, J.; Hu, Z.; Li, J.; Zhang, A.; and Huang, H. 2023a. Federated conditional stochastic optimization. *arXiv preprint arXiv:2310.02524*.

Wu, X.; Sun, J.; Hu, Z.; Zhang, A.; and Huang, H. 2023b. Solving a class of non-convex minimax optimization in federated learning. *arXiv preprint arXiv:2310.03613*.

Wu, Y.; Huang, H.; and Zhang, H. 2023. A law of robustness beyond isoperimetry. In *International Conference on Machine Learning*, 37439–37455. PMLR.

Wu, Y.; Zhang, H.; and Huang, H. 2022. Retrievalguard: Provably robust 1-nearest neighbor image retrieval. In *International Conference on Machine Learning*, 24266–24279. PMLR.

Xu, M.; Zhang, J.; Ni, B.; Li, T.; Wang, C.; Tian, Q.; and Zhang, W. 2020. Adversarial Domain Adaptation with Domain Mixup. In *AAAI Conference on Artificial Intelligence*, 6502–6509.

Yang, F.; Cheng, Y.; Shiau, Z.; and Wang, Y. F. 2021. Adversarial Teacher-Student Representation Learning for Domain Generalization. In *Advances in Neural Information Processing Systems*, 19448–19460.

Zhao, H.; des Combes, R. T.; Zhang, K.; and Gordon, G. J. 2019. On Learning Invariant Representations for Domain Adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, 7523–7532.

Zhou, K.; Yang, Y.; Qiao, Y.; and Xiang, T. 2021. Domain Generalization with MixStyle. In *International Conference on Learning Representations*.