2023 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2023, ROME, ITALY

ON THE FAILURE OF INVARIANT RISK MINIMIZATION AND AN EFFECTIVE FIX VIA CLASSIFICATION ERROR CONTROL

Thuan Nguyen[¶†] *Matthias Scheutz*[¶] *Shuchin Aeron*[†]

[¶] Department of Computer Science, Tufts University, Medford, MA 02155
[†] Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155

ABSTRACT

Invariant Risk Minimization is a well-known Domain Generalization framework that has received much attention over the past few years. Invariant Risk Minimization is capable of learning domain-invariant features from multiple domains by finding representation features such that the optimal classifier on top of these features matches all training domains. In this paper, we show that even though the Invariant Risk Minimization algorithm is based on a compelling idea, it is easily vulnerable in a simple toy example where multiple domain-invariant features exist and each possesses a corresponding classifier that is optimal for all domains. Based on this observation, we propose an effective modification of the traditional Invariant Risk Minimization algorithm named Error-Control Invariant Risk Minimization, which allows learning different domain-invariant features via controlling the training classification error, leading to a new algorithm that works well on both our toy synthetic dataset and the real-world datasets.

Index Terms— Domain generalization, multiple domain-invariant features, classification error constraints.

1. INTRODUCTION

The machine learning theory and algorithms have overwhelmingly relied on the assumption that the training and the test data are independent and identically distributed, *i.e.*, the data samples and labels from the training set and the test set must follow the same underlying distribution. However, the independent and identically distributed assumption may not hold in practice due to a ubiquitous phenomenon usually referred to as *distribution shift* for which the distributions of data and/or labels on the training set and the test set are not identical. Distribution shift is the primary cause of the performance deterioration of traditional learning algorithms, for example, it has been observed that the Empirical Risk Minimization algorithm (ERM) fails when facing samples from a different domain, even under a mild distribution shift [1]. Mitigating the problem caused by the distribution shift is the primary goal of the *Domain Generalization* (DG) problem, where a model is trained using data from several *seen* domains but later needs to be applied to *unseen* (unknown but related) domains with different data and/or label distributions.

A vast of DG theories and algorithms are mainly based on the theme of domain invariant representation learning, *i.e.*, training a representation function and its corresponding classifier to learn the *domain-invariant features* that are unchanged and stable across domains [1–4]. Over the past few years, the Invariant Risk Minimization algorithm (IRM) [1] has appeared as the state-of-the-art DG method which is capable of effectively learning the domain invariant features while avoiding the spurious (fake) invariant features. For given data from multiple training (seen) domains, IRM allows a mechanism to find the invariant features by looking for a representation function and its corresponding optimal classifier such that this optimal classifier works equally well on the representation features from all training domains. In other words, under IRM settings, features are called domain invariant if one can design a corresponding *invariant classifier* (based on these features) such that it is simultaneously optimum regardless of training domains[1]. Even though the Invariant Risk Minimization algorithm is based on a compelling idea, we show that it is easily vulnerable in our toy example where there exist two domain invariant features, each owning a corresponding optimal classifier that matches all training domains. To handle this failure situation, we propose a simple fix but allow IRM works well on both our toy example and the real-world datasets.

We summarize our contributions as follows:

1. We introduce a new synthetic dataset named Equally-CMNIST (E-CMNIST) and numerically show that the well-known IRM algorithm will fail on E-CMNIST where multiple domain invariant features with different levels of classification error simultaneously exist.

2. We propose an effective modification that allows IRM to work well on E-CMNIST by imposing a constraint on the training classification error, leading to a new algorithm that is capable of selecting distinct domain invariant features.

[1] Of course, IRM only works under an implicit assumption that such invariant features and their corresponding invariant classifiers exist.

979-8-3503-2411-2/23/$31.00 ©2023 IEEE

3. Our numerical results demonstrate the effectiveness of the proposed algorithm not only in our synthetic dataset but also in the real-world DG benchmark datasets.

The remainder of this paper is structured as follows. In Sec. 2, we provide a short survey of recent works in DG, specifically, the works that are closely related to the IRM algorithm. In Sec. 3, we formally define the DG problem and summarize the main idea of the IRM framework. Based on the failure case of IRM observed in Sec. 4, we provide an effective fix in Sec. 5, leading to a new DG algorithm. The effectiveness of the proposed algorithm is demonstrated in Sec. 6. Finally, we discuss the limitations of our proposed approach in Sec. 7, and conclude in Sec. 8.

2. RELATED WORK

Over the past decade, domain-invariant learning has become the most widely used and successful method in DG. Domain-invariant learning decomposes the learning process into two parts: (1) learning representation functions that output the domain-invariant features, and (2) designing optimal classifiers based on these invariant features [5–8][2].

Invariant Risk Minimization algorithm (IRM) was introduced in [1] and was extended or served as the basis for the development of other algorithms in [9–14]. For given data from multiple training (seen) domains, IRM finds the invariant features by looking for a representation function and its corresponding optimal classifier such that this optimal classifier must work equally well *i.e.*, invariant across all training domains. In other words, by looking for invariant classifiers, IRM targets learning such features having the conditional distribution between labels given features stable regardless of training domains. Although the approach seems promising, IRM is vulnerable under particular settings. Kamath *et al.* [15] construct a dataset where there exist some undesired predictors but are considered invariant under IRM settings, leading to the failure of the learned model on the unseen domain. Guo *et al.* [9] consider a scenario where the correlation between spurious features and labels is much stronger than the correlation between true invariant features and labels, thus, enforcing the model to learn the spurious features to achieve high classification accuracy on seen domains while ignoring to learn the true invariant features. Other failure cases of IRM can be observed from [16] and [17] where the pseudo-invariant features and geometric skews exist, thus, the classifiers will utilize both the true and the pseudo-invariant features, leading to the failure when applying these classifiers on the unseen domains.

Since all proposed algorithms in [9–14] mainly rely on the theme of IRM, if IRM fails, there is a high chance that its descendants will fail. Therefore, determining the failure situations of IRM and finding the corresponding effective fix is important. In this paper, we construct a synthetic dataset having multiple domain invariant features, leading to the failure of IRM. To overcome this failure, an effective modification that aims to control the training classification error is introduced which not only allows better performances on our proposed synthetic dataset but also on other real-world datasets.

[2]Note that domain-invariant learning only works under implicit assumptions such that the domain-invariant features exist and are useful to predict the labels.

3. PROBLEM FORMULATION

In this section, we formally define the Domain Generalization (DG) problem and re-capture the main idea of the Invariant Risk Minimization (IRM) algorithm.

3.1. Domain generalization

Under DG settings, one can observe the data from a set of K observed (seen) domains $\mathcal{D} = \{D^{(1)}, D^{(2)}, \dots, D^{(K)}\}$. Note that the data/label distributions of these K domains are not necessarily independent and identically distributed. DG algorithms aim to train a predictor from the input space to the label space composed by: (1) a representation function $f : \mathcal{X} \to \mathcal{Z}$ from the input space $\mathcal{X}$ to the representation space $\mathcal{Z}$, and (2) a classifier $g : \mathcal{Z} \to \mathcal{Y}$ from the representation space $\mathcal{Z}$ to the label space $\mathcal{Y}$ on the data from K seen domains such that this predictor induces a small classification error on a new (unseen) domain $D^{(u)} \notin \mathcal{D}$. Formally, DG algorithms want to handle the following optimization problem:

$$\min_{f,\, g} \;\; R^{(u)}(g \circ f) \tag{1}$$

where $R^{(u)}(g \circ f)$ denotes the classification risk (error) induced by f and g on the unseen domain $D^{(u)}$.

3.2. Invariant risk minimization

The Invariant Risk Minimization (IRM) algorithm [1] solves (1) by finding a representation function f such that there exists an invariant classifier g which is simultaneously optimal for representation features outputted by f across all training domains. Here, the implicit assumption is that such representation function and classifier must exist. To find f and g, IRM optimizes the following bi-level optimization problem:

$$\min_{f,g} \sum_{i=1}^{K} R^{(i)}(g \circ f), \text{ s.t. } g \in \underset{\bar{g}:\mathcal{Z}\to\mathcal{Y}}{\arg\min} \; R^{(i)}(\bar{g} \circ f), \; \forall i \tag{2}$$

where $R^{(i)}(g \circ f)$ denotes a classification error induced by f and g in domain i, $i = 1, 2, \dots, K$. In practice, $R^{(i)}(g \circ f)$ is usually approximated by the cross-entropy loss.

Since the above bi-level optimization problem is hard to solve, Arjovsky *et al.* only considers a class of linear classifiers and further argues that linear classifier can be finally replaced by a scalar classifier [1], leading to the following objective function:

$$\min_{f} \sum_{i=1}^{K} R^{(i)}(f), \text{ s.t. } \nabla_{g|g=1} R^{(i)}(g \cdot f) = 0, \; \forall i, \tag{3}$$

where $\nabla_{g|g=1} R^{(i)}(g \cdot f)$ denotes the gradients of the classification error in different domains, and g is a scalar classifier.

Finally, to break the bi-level optimization, Arjovsky *et al.* proposes the below practical objective function:

$$\min_{f} \sum_{i=1}^{K} \Big[R^{(i)}(f) + \alpha \, \|\nabla_{g|g=1} R^{(i)}(g \cdot f)\|^2 \Big], \qquad (4)$$

where α is a hyper-parameter that controls the balance between two terms in the objective function, and $\|.\|^2$ denotes the squared Euclidean norm. From this point, we stick with the IRM objective function in (4). The first term in (4) controls the classification error on training (seen) domains to guarantee the useful features will be extracted while the second term in (4) ensures that the classifier works equally well (approximately invariant) for all training (seen) domains. Under quite restricted assumptions (linear classifiers, the classification error function is convex and differentiable) Theorem 4 in [1] shows that minimizing the proposed IRM loss in (4) will yield an invariant classifier/predictor over all training domains.

4. A FAILURE CASE OF INVARIANT RISK MINIMIZATION ALGORITHM

The IRM algorithm allows learning an optimal classifier that is invariant across all training (seen) domains, thus, having a high chance to accurately adapt to a new (unseen) domain. Although the approach seems promising, IRM will fail if there exist multiple representation features and each has one corresponding optimal classifier that equally matches for all domains. To illustrate this argument, we construct a toy example derived from Colored-MNIST (CMNIST) dataset [1], called Equally-CMNIST (E-CMNIST). While both the colors and digits in E-CMNIST and the original CMNIST contain information about the labels, the main difference between E-CMNIST and CMNIST is that: in E-CMNIST, both colors and digits possess their own optimal classifier that *equally* matches for all domains (thus the same values of the second term in (4) can be approximately achieved by learning colors or digits), however, a classifier using digits will induce lower classification error than a classifier using colors (thus the lower value of the first term in (4) can be achieved by learning the digits). Consequently, the IRM algorithm which minimizes the loss in (4) will learn the digits and ignore the colors, thus, will attain a low classification accuracy if, on the unseen domain, the colors are more useful to predict the labels than the digits. Our data construction is described in detail as follows.

4.1. Construction of Equally-CMNIST dataset

E-CMNIST is a variant of the CMNIST dataset proposed in [1]. There are two training (seen) domains each containing 25,000 images and one test unseen domain contains 10,000 images. Let $i = 1, 2, 3$ denote three domains, $X_g^{(i)}$ denote the gray image in domain i, $Y_g^{(i)}$ denote the label of $X_g^{(i)}$, $C^{(i)}$ denote the color will be later added to gray image $X_g^{(i)}$, finally, $X^{(i)}$ and $Y^{(i)}$ denote the colored image and its corresponding label that will be used for the classification tasks. The graphical model to construct E-CMNIST dataset is illustrated in Fig. 1 which contains the following steps:

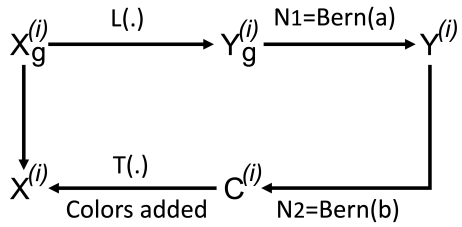

Fig. 1. Graphical model for our proposed Equally-CMNIST (E-CMNIST) dataset.

1. $Y_g^{(i)} \leftarrow L(X_g^{(i)})$: from gray images containing digits from "0" to "9" in the MNIST dataset [18], we construct a binary classification problem by labeling $Y_g^{(i)} = 0$ if the digits are strictly less than "5", and labeling $Y_g^{(i)} = 1$, otherwise.

2. $Y^{(i)} \leftarrow Y_g^{(i)} \oplus N_1$, where $N_1 = \text{Bern}(a)$: adding noise to the label via a Bernoulli function with hyper-parameter $a \in [0, 1]$ to create the final label. Due to the added noise, using digits to predict the final label $Y^{(i)}$ can only achieve the minimum error rate of a, or equivalently, the maximum accuracy rate of $1 - a$.

3. $C^{(i)} \leftarrow Y^{(i)} \oplus N_2$, where $N_2 = \text{Bern}(b)$: the index color $C^{(i)}$ is selected via Bernoulli function with hyper-parameter $b \in [0, 1]$. Here, we only use two index colors: green and red, $C^{(i)} \in \{\text{green, red}\}$. If $Y^{(i)} = 0$, the image is colored green with a probability of b and red with a probability of $1 - b$, If $Y^{(i)} = 1$, the image is colored green with a probability of $1 - b$ and red with a probability of b. Thus, using colors to predict the final label $Y^{(i)}$ can only achieve the minimum error rate of b, or equivalently, the maximum accuracy rate of $1 - b$.

4. $X^{(i)} \leftarrow T(X_g^{(i)}, C^{(i)})$: the gray image $X_g^{(i)}$ is colored with the index color $C^{(i)}$ to produce the final colored image $X^{(i)}$.

5. Selecting different values of (a, b) and repeating the above process for $i = 1, 2, 3$, we have two training domains ($i = 1, 2$) and one test domain ($i = 3$). The colored image $X^{(i)}$ and its label $Y^{(i)}$ are used for classification tasks.

For two training (seen) domains, we set $(a = 0, b = 0.1)$. For the test (unseen) domain, we select $(a = 0.9, b = 0.1)$. By construction, both colors and digits are stable on two training

Algorithms	Training accuracy	Test accuracy
IRM [1]	96.69%	32.33%

Table 1. The best test accuracy and its corresponding training accuracy of IRM [1] on E-CMNIST dataset after 10 times repeating the whole experiment.

domains, each owns a classifier that performs equally well on two training domains, thus, yields a very similar value for the second term in the IRM loss in (4). However, since $a = 0 < b = 0.1$, the digits produce lower classification error (the first term in (4)) than the colors on two training domains, thus, optimizing the IRM loss in (4) will enforce the models to learn the digits and ignore the colors. Next, by increasing $a = 0.9$, but keeping the same $b = 0.1$, only colors are useful in predicting the labels from the unseen domain, therefore, the selected model on training domains that prefers the digits will definitely fail on the unseen domain.

4.2. Failure of IRM on E-CMNIST

To verify the failure of IRM on the E-CMNIST dataset, we follow the experiment established in [1]. Particularly, the neural network is composed of three linear layers with feature map dimensions of 196, 390, and 390, each linear layer is followed by a Rectified Linear Unit (ReLU) activation. The last layer is a linear layer that aims to classify the label of images to "0" or "1". We use Adam optimizer for training with a learning rate of 5×10^{-4}, and the total number of steps is set to 500. We repeat the whole experiment 10 times by selecting 10 random seeds, for each random seed, the whole training and test processes are repeated from scratch. Note that all these hyper-parameters are exactly the same as the ones used in [1] which can be viewed from this link[3].

IRM's training and test accuracy on the E-CMNIST dataset can be viewed in Table 1. As seen, the best test accuracy on the unseen domain selected from running the whole experiment 10 times is only 32.33% even though the training accuracy can reach 96.69%. We conclude that under particular settings in the E-CMNIST dataset where there exist multiple optimal classifiers that match all training domains, the well-known IRM algorithm will fail.

In the next section, we provide a modification to the traditional IRM algorithm that allows us to learn different features (digits or colors) and its corresponding optimal classifiers which finally allows us to achieve better test accuracy on the unseen domain.

5. AN EFFECTIVE FIX FOR INVARIANT RISK MINIMIZATION ALGORITHM

In this section, we provide an effective fix that helps IRM performs well on the E-CMNIST dataset. Indeed, by construction, setting $a = 0.9$ and $b = 0.1$ makes colors more strongly correlated with labels than digits on the unseen domain, thus, the IRM algorithm will fail if it is not able to catch the colors. However, as previous analysis, by optimizing the loss function in (4), the traditional IRM algorithm will definitely select digits while ignoring colors.

To handle this situation, we propose a method that is capable of learning colors or digits by *controlling the classification error* induced from the training phase. To do that, we introduce a modification of the IRM algorithm named Error-Control-IRM (EC-IRM). In practice, EC-IRM optimizes the loss function below:

$$\min_{f,g} \sum_{i=1}^{K} \Big[\underbrace{\text{ReLU}\Big(\beta - R^{(i)}(f)\Big)}_{\text{new term}} + \underbrace{R^{(i)}(f) + \alpha \|\nabla_{g|g=1} R^{(i)}(g \cdot f)\|^2}_{\text{Original IRM terms}} \Big] \tag{5}$$

which differs from the original IRM loss in (4) by adding a new term $\text{ReLU}\big(\beta - R^{(i)}(f)\big)$ where $\text{ReLU}\big(.\big)$ denotes the Rectified Linear Unit function defined by:

$$\text{ReLU}\Big(\beta - R^{(i)}(f)\Big) = \begin{cases} 0, & \text{if } \beta \leq R^{(i)}(f) \\ \beta - R^{(i)}(f), & \text{otherwise.} \end{cases}$$

Indeed, $\text{ReLU}\big(\beta - R^{(i)}(f)\big)$ is parameterized by β will penalize if the classification error $R^{(i)}(f) < \beta$. Thus, the new term $\text{ReLU}\big(\beta - R^{(i)}(f)\big)$ acts as a constraint to control the classification error of the learned models. By varying the values of β, it is possible to learn the invariant features with different levels of classification error. Since learning different features (colors or digits) can vary the values of classification error in the E-CMNIST dataset, adding this constraint helps the model discern and then is able to select different features. This is why the proposed algorithm is called Error-Control-IRM (EC-IRM) algorithm.

To illustrate how EC-IRM works, we consider the E-CMNIST dataset again. In E-CMNIST, by construction, both colors and digits are domain invariant features *i.e.*, each corresponds to an optimal classifier that matches all seen domains. For this reason, both colors and digits approximately yield the same value for the second terms in the traditional IRM loss function in (4). Thus, to optimize (4), the network will prefer the digits to achieve a lower value for the first term (classification error) in (4), leading to the failure on the test phase where the colors are more useful than the digits on the unseen domain. On the other hand, our EC-IRM algorithm is able to select different features with different levels of classification error by varying the value of β. For example, by selecting a large enough value for β, the model will prefer the invariant features with a high classification error, thus, will pay more attention to learning colors.

In the next section, we will demonstrate that the proposed EC-IRM algorithm not only performs well on the synthetic E-CMNIST dataset but also on other real-world datasets.

[3]https://github.com/facebookresearch/InvariantRiskMinimization

Algorithms	Training accuracy	Test accuracy
EC-IRM ($\beta = 0.1$)	95.66%	39.97%
EC-IRM ($\beta = 0.2$)	92.64%	63.85%
EC-IRM ($\beta = 0.3$)	89.86%	89.75%
EC-IRM ($\beta = 0.4$)	89.88%	**90.07**%
EC-IRM ($\beta = 0.5$)	90.03%	90.00%
EC-IRM ($\beta = 0.6$)	51.08%	49.95%
EC-IRM ($\beta = 0.7$)	53.14%	54.16%
EC-IRM ($\beta = 0.8$)	52.31%	55.56%
EC-IRM ($\beta = 0.9$)	49.61%	50.96%
IRM [1]	96.69%	32.33%

Table 2. The best test accuracy and its corresponding training accuracy of EC-IRM (ours) and IRM on E-CMNIST dataset selected from 10 times repeating the whole experiment. For EC-IRM, we report the accuracy with different values of hyper-parameter $\beta \in [0.1, 0.9]$ that controls the constraint on training classification error.

Algorithms	PACS	Office-Home
EC-IRM ($\beta = 0.01$)	84.5%	65.2%
EC-IRM ($\beta = 0.02$)	84.6%	64.9%
EC-IRM ($\beta = 0.03$)	85.1%	65.1%
EC-IRM ($\beta = 0.04$)	85.4%	65.1%
EC-IRM ($\beta = 0.05$)	85.6%	65.9%
EC-IRM ($\beta = 0.06$)	85.7%	65.9%
EC-IRM ($\beta = 0.07$)	85.7%	66.4%
EC-IRM ($\beta = 0.08$)	**85.8**%	66.4%
EC-IRM ($\beta = 0.09$)	85.6%	**66.5**%
IRM [1]	83.5%	64.3%

Table 3. The test accuracy on the unseen domain of EC-IRM (ours) and IRM on PACS and Office-Home datasets. For EC-IRM, we report the accuracy with different values of hyper-parameter $\beta \in [0.01, 0.09]$ that controls the constraint on classification error.

6. NUMERICAL RESULTS

6.1. Synthetic dataset

We use the same experimental setting as described in Section 4 when testing our proposed EC-IRM algorithm on the E-CMNIST dataset. The EC-IRM algorithm is compared against its original version, the IRM algorithm. Table 2 provides the accuracies of EC-IRM with different values of hyper-parameter $\beta \in [0.1, 0.9]$ that controls the constraint on the training classification error. As seen, EC-IRM consistently outperforms its baseline IRM for any value of $\beta \in [0.1, 0.9]$. The best accuracy of EC-IRM on the test (unseen) domain is 90.07% achieved at $\beta = 0.4$ which is approximately the theoretical maximum value induced by colors on the test domain when setting $b = 0.1$.

6.2. Real-world datasets

Even though EC-IRM significantly outperforms the traditional IRM on our synthetic dataset, we still want to verify whether our method works well on a real-world dataset. To do so, we examine EC-IRM on two datasets: PACS [19] and Office-Home [20]. We follow the settings in SWAD [21] to perform our experiments.

Due to limited resources, we only report the performance of EC-IRM for $\beta = [0.01, 0.09]$ while the accuracy for the baseline IRM is directly adopted from [22]. As seen, from Table 3, EC-IRM outperforms the traditional IRM for all values of β between 0.01 and 0.09. Specifically, the largest gain achieved by EC-IRM over IRM is 2.3% on the PACS dataset and 2.2% on the Office-Home dataset. Based on our numerical results, we recommend using a small value of β between 0.05 and 0.10 for real-world datasets. Finally, our implementation is available to download at this link[4].

7. DISCUSSIONS AND LIMITATIONS

In [8], Lyu *et al.* pointed out that there is a fundamental trade-off while minimizing a class of common objective functions in DG that usually includes two terms: (1) the classification error, and (2) the domain discrepancy (for learning domain-invariant features). Particularly, Theorem 1 in [8] showed that one can not perfectly minimize both two terms in DG's objective function, *i.e.*, if the algorithm focuses too much on minimizing one term, it will definitely enlarge the other term and subsequently may fail on the test phase on the unseen domain. We believe that IRM algorithm is not an exception and will follow this trade-off. Applying the result in [8] into IRM implies a crucial fact that a model having the lowest training classification error may not be the best model on the unseen domain. Interestingly, this is coincident with the idea of the proposed EC-IRM algorithm where we propose a mechanism to control the classification error, thus, enabling one to select the model with different values of training error. Indeed, by using $\beta > 0$, EC-IRM enforces the network to avoid the model with the lowest training accuracy, thus, may be able to achieve lower values of domain discrepancy and precisely learn the domain-invariant features. Even though our focus in this paper is just the IRM algorithm, our idea as well as the theory in [8] are not limited to IRM only and can be extended to other DG algorithms by adding one more term that manages the classification error on the training process.

For the limitations, we believe that the proposed method will not work under some particular settings. For example, if the distributions of data from seen and unseen domains are identical, then one should select a model that has the lowest

[4] https://drive.google.com/file/d/1Qkym-G_U4gS2IX9AUlbZkBgTv28I0NEY/view?usp=sharing

classification error on seen domains to apply to the unseen domain. In this case, adding a classification error constraint will definitely harm the generalization performance.

8. CONCLUSIONS

In this paper, we proposed a new synthetic dataset named E-CMNIST where multiple domain invariant features exist and each possesses its optimal classifier that matches all training domains, leading to the failure of the well-known IRM algorithm. To make IRM works on E-CMNIST, we modified its original objective function by adding a constraint on the training classification error, leading to a new algorithm named EC-IRM that does not only outperform the traditional IRM on our synthetic dataset but also on other real-world datasets.

9. ACKNOWLEDGMENT

The authors would like to acknowledge funding provided by AFOSR grant # FA9550-18-1-0465.

10. REFERENCES

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz, "Invariant risk minimization," *stat*, vol. 1050, pp. 27, 2020.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al., "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, pp. 137, 2007.

[3] Baochen Sun and Kate Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[4] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5400–5409.

[5] Yingjun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees GM Snoek, and Ling Shao, "Learning to learn with variational information bottleneck for domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 200–216.

[6] Fan Zhou, Zhuqing Jiang, Changjian Shui, Boyu Wang, and Brahim Chaib-draa, "Domain generalization with optimal transport and metric learning," *arXiv preprint arXiv:2007.10573*, 2020.

[7] Divyat Mahajan, Shruti Tople, and Amit Sharma, "Domain generalization using causal matching," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7313–7324.

[8] Boyang Lyu, Thuan Nguyen, Matthias Scheutz, Prakash Ishwar, and Shuchin Aeron, "A principled approach to model validation in domain generalization," *arXiv preprint arXiv:2304.00629*, 2023.

[9] Ruocheng Guo, Pengchuan Zhang, Hao Liu, and Emre Kiciman, "Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix," *arXiv preprint arXiv:2101.07732*, 2021.

[10] Kartik Ahuja, Ethan Caballero, Dinghuai Zhang, Jean-Christophe Gagnon-Audet, Yoshua Bengio, Ioannis Mitliagkas, and Irina Rish, "Invariance principle meets information bottleneck for out-of-distribution generalization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3438–3450, 2021.

[11] Bo Li, Yifei Shen, Yezhen Wang, Wenzhen Zhu, Dongsheng Li, Kurt Keutzer, and Han Zhao, "Invariant information bottleneck for domain generalization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, vol. 36, pp. 7399–7407.

[12] Thuan Nguyen, Boyang Lyu, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron, "Conditional entropy minimization principle for learning domain invariant representation features," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 3000–3006.

[13] Thuan Nguyen, Boyang Lyu, Prakash Ishwar, Matthias Scheutz, and Shuchin Aeron, "Joint covariate-alignment and concept-alignment: a framework for domain generalization," in *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2022, pp. 1–6.

[14] Yong Lin, Hanze Dong, Hao Wang, and Tong Zhang, "Bayesian invariant risk minimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16021–16030.

[15] Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro, "Does invariant risk minimization capture invariance?," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 4069–4077.

[16] Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski, "The risks of invariant risk minimization," in *International Conference on Learning Representations*, 2021.

[17] Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur, "Understanding the failure modes of out-of-distribution generalization," in *International Conference on Learning Representations*, 2021.

[18] Yann LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[19] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5542–5550.

[20] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5018–5027.

[21] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park, "Swad: Domain generalization by seeking flat minima," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22405–22418, 2021.

[22] Ishaan Gulrajani and David Lopez-Paz, "In search of lost domain generalization," in *International Conference on Learning Representations*, 2021.