Conditional entropy minimization principle for learning domain invariant representation features

Thuan Nguyen*, Boyang Lyu*, Prakash Ishwar†, Matthias Scheutz‡ and Shuchin Aeron*
*Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155
†Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215
‡Department of Computer Science, Tufts University, Medford, MA 02155
Email: *Thuan.Nguyen@tufts.edu, Boyang.Lyu@tufts.edu, pi@bu.edu, Matthias.Scheutz@tufts.edu, Shuchin@ece.tufts.edu*

Abstract—**Invariance-principle-based methods such as Invariant Risk Minimization (IRM), have recently emerged as promising approaches for Domain Generalization (DG). Despite promising theory, such approaches fail in common classification tasks due to mixing of *true invariant features* and *spurious invariant features*[1]. To address this, we propose a framework based on the conditional entropy minimization (CEM) principle to filter-out the spurious invariant features leading to a new algorithm with a better generalization capability. We show that our proposed approach is closely related to the well-known Information Bottleneck (IB) framework and prove that under certain assumptions, entropy minimization can exactly recover the true invariant features. Our approach provides competitive classification accuracy compared to recent theoretically-principled state-of-the-art alternatives across several DG datasets.**

I. Introduction

A fundamental assumption in most statistical machine learning algorithms is that the training data and the test data are independently and identically distributed (i.i.d). However, it is usually violated in practice due to a phenomenon often referred to as the domain distribution shift where the training domain and the test domain distributions are not the same. This leads to an increased risk/error of the trained classifier on the test domain. Mitigating this issue is the subject of the area broadly referred to as Domain Generalization (DG).

Over the past decade, many methods have been proposed for DG, under different settings [3] [4]. Among these, Invariant Risk Minimization (IRM) [5] [6] has emerged as one of the promising methods. IRM is constructed based on a widely accepted assumption that the representations are general and transferable if the feature representations remain invariant from domain to domain. However, this approach is shown to fail in some simple settings where spurious invariant features exist [2] [7] [8] [9]. A particular example is the problem of classifying cow and camel images [10] [11] where the label is a deterministic function of the invariant features, for example, the shape of animals, and does not depend on the spurious features such as the background. However, because cows usually appear in a picture with a greenfield while the camels live in a desert with a yellow background, the background color could be incorrectly learned as a spurious invariant feature. This can lead to classification errors, for example, if the cow is placed in a yellow field, then it may be misclassified as a camel. Therefore, even though an invariance-principle-based approach can learn invariant features, it may still fail in a classification task if the extracted features contain not only the true invariant features but also spurious invariant features. These spurious features could be eliminated if one can observe a sufficiently large number of domains [1] [2]. For example, if the seen domain contains a picture of a cow walking in a desert. However collecting labeled data from all possible domains is impractical.

Several frameworks have been proposed to deal with the presence of spurious invariant features. For example, in [12] the entropy of the extracted features is minimized to filter out spurious features. However, only linear classifiers are considered and although the approach is motivated by the Information Bottleneck (IB) framework [13], IB is not directly utilized in the learning objective. A similar approach directly based on the IB objective function for eliminating spurious invariant features appears in [14] [15]. Although numerical results in [14] [15] significantly outperform the state-of-the-art methods, the methods are heuristically motivated and lack theoretical justification.

In contrast to previous works, the key contributions of this paper are the following:

- We propose a new objective function that is motivated by the conditional entropy minimization (CEM) principle and show that it is explicitly related to the Deterministic Information Bottleneck (DIB) principle [16].
- We theoretically show that under some suitable assumptions, minimizing the proposed objective function will filter-out spurious features.
- Our approach is general in the sense that it is able to handle non-linear classifiers and may be extended to other DG methods that employ the invariance-principle.

The key idea of our approach is to adopt the IRM framework for learning a good representation function that can capture both the true invariant features and the spurious invariant features, but penalize the conditional entropy of representations given labels to filter-out the spurious invariant features.

The remainder of this paper is structured as follows. In Section II, we summarize relevant work on DG and briefly introduce the IRM algorithm and the IB framework. In Section III we formally define the problem and state and discuss the

[1] We use the terms "spurious invariant features" or just "spurious features" to denote features that are invariant across all seen domains, but change in the unseen domain [1] [2].

main assumptions underlying our theoretical analysis. Section IV provides the main theoretical results which motivate our practical approach proposed in Section V. Experiments and their results are described in Section VI with concluding remarks presented in Section VII.

II. Related Work

A. Domain Generalization

Numerous DG methods have been proposed in the past ten years which can be broadly categorized into some major directions, chiefly "data manipulation", representation learning, and meta-learning. The performance of a learning model often relies on the quantity and diversity of the training data and data manipulation is one of the cheapest methods to generate samples from a given set of limited data. Data manipulation can be employed via data augmentation [17] [18], domain randomization [19], or adversarial data augmentation [20], [21]. The representation learning approach aims to learn a good representation feature by decomposing the prediction function into a representation function followed by a classifier. Over the past decade, many methods are emerged for better representation learning which can be categorized into two different learning principles: domain-invariant representation learning and feature disentanglement. Domain-invariant representation learning is based on the assumption that the representations are general and transferable to different domains if the representation features remain invariant from domain to domain [22]. Notably, domain-invariant representation learning has emerged as one of the most common and efficient approaches in DG and provided many promising results [5] [6] [12] [14] [15] [23]–[27]. Finally, meta-learning methods aim to learn the algorithm itself by learning from previous experience or tasks, i.e., learning-to-learn. Even though meta-learning is a general learning framework, it has recently been applied to DG tasks [15] [28] [29]. For more details, we refer the reader to the recent surveys on DG in [3] and [4].

B. Information Bottleneck and Invariant Risk Minimization

In this section we review the IB framework [13] [16] and the IRM algorithm [5] which are directly related to our proposed method. We use $f : \mathcal{X} \to \mathcal{Z}$ to denote a (potentially stochastic) representation mapping from the input data space $\mathcal{X}$ to the representation space $\mathcal{Z}$ and $g : \mathcal{Z} \to \mathcal{Y}$ to denote a classifier/labeling function from the representation space $\mathcal{Z}$ to the label space $\mathcal{Y}$.

1) Information Bottleneck Principle: The IB method aims to find the best trade-off between accuracy and complexity (compression) when summarizing a random variable [13]. Particularly, IB aims to find a good (stochastic) representation function f^* by solving the following optimization problem:

$$f^* = \arg\min_f I(X;Z) - \theta I(Y;Z), \tag{1}$$

where $I(X;Z)$ denotes the mutual information between the random variable X corresponding to input data and its representation $Z = f(X)$, $I(Y;Z)$ denotes the mutual information between the random variable Y corresponding to the label and Z, and θ is a positive hyper-parameter that controls the trade off between maximizing $I(Y;Z)$ and minimizing $I(X;Z)$. Mutual information is a nonnegative statistical measure of dependence between random variables with larger values corresponding to stronger dependence and value zero corresponding to independence. Thus, the IB framework aims to find a representation Z that is weakly dependent on input X, but is strongly dependent on the prediction label Y. Indirect rate-distortion source coding in information-theory provides an alternative interpretation of the IB objective with Z viewed as a "compressed" encoding of X, $I(X;Z)$ is the number of "bits" needed to compress X to Z, and $I(Y;Z)$ is a measure of how well the label Y can be decoded from Z, i.e., a measure of prediction accuracy or "inverse-distortion". The IB problem can then be stated as a Lagrangian formulation of minimizing the number of bits needed to compress X to Z while being able to recover Y from Z to a desired accuracy.

The Deterministic Information Bottleneck (DIB) [16] problem aims to find f by solving the following optimization problem which is closely related to (1):

$$f^* = \arg\min_f H(Z) - \theta I(Y;Z). \tag{2}$$

For $\theta = 1$, $H(Z) - \theta I(Y;Z) = H(Z|Y)$ which is the conditional entropy of the representation variable Z given the label Y. Thus, minimizing $H(Z|Y)$ is a special case of DIB where the aims of compression, i.e., minimizing $H(Z)$, and accuracy, i.e., maximizing $I(Y;Z)$, are equally weighted (balanced).

2) Invariant Risk Minimization Algorithm: The IRM algorithm [5] aims to find the representation $Z = f(X)$ for which the optimum classifier g is invariant across all domains. The implicit assumption is that such representations and optimum domain-invariant classifiers exist. In practice, this is approximately realized by solving the following optimization problem [5]:

$$\min_{h \in \mathcal{G} \circ \mathcal{F}} L_{IRM}(h, \alpha) := \sum_{i=1}^{m} \left[R^{(i)}(h) + \alpha \, \|\nabla_{t|t=1.0} R^{(i)}(t \cdot h)\|^2 \right], \tag{3}$$

where $\mathcal{F}$ is a family of representation functions (typically parameterized by weights of a neural network with a given architecture), $\mathcal{G}$ a family of *linear* classifiers (typically the last fully connected classification layer of a classification neural network), $R^{(i)}(g \circ f) := \mathbb{E}_{(X,Y) \sim D_i}[\ell(g(f(X)), Y)]$ denotes a classification risk (e.g., error or cross-entropy loss) of using a representation function f followed by a classifier g in domain i when using loss function ℓ, and α is a hyper-parameter associated with the squared Euclidean norm of the gradients (denoted by ∇) of the risks in different domains. When restricted to the family of linear classifiers and convex differentiable risk functions, Theorem 4 of [5] shows (under certain technical assumptions) that minimizing L_{IRM} will yield a predictor that not only (approximately) minimizes the cumulative risk across all domains (the first term in L_{IRM}), but is also approximately optimum simultaneously across all domains, i.e., approximately invariant, and this is captured by the sum of squared risk gradients across all domains.

In this paper, we rely on the IRM algorithm [5] to extract the invariant features and use the CEM principle to filter out the spurious invariant features. We note, however, that our approach is applicable to any method that can learn invariant features. We chose IRM due to its popularity and good empirical performance.

III. Problem Formulation

In this section, we formulate the minimum conditional entropy principle, which is a special case of the DIB principle, and show that it can be used to filter out spurious features. To do this we first introduce three modeling assumptions underlying our proposed approach. Our assumptions embrace two key ideas (i) the learned features are a linear mixture (superposition) of "true" domain-invariant features and "spurious" domain-specific features, and (ii) the invariant features are conditionally independent of spurious features given the label.

A. Notation

Consider a classification task where the learning algorithm has access to i.i.d. data from the set of m domains $\mathbb{D} = \{D_1, D_2, \dots, D_m\}$. The DG task is to learn a representation function $f : \mathcal{X} \to \mathcal{Z}$ from the input data space $\mathcal{X}$ to the representation space $\mathcal{Z}$, and a classifier $g : \mathcal{Z} \to \mathcal{Y}$ from the representation space $\mathcal{Z}$ to the label space $\mathcal{Y}$ that generalizes well to an unseen domain $D_s \notin \mathbb{D}$.

Let X denote the data random variable in input space, Y the label random variable in label space, and Z the extracted feature random variable in representation space. Let the invariant and spurious features be denoted by Z_{inv} and Z_{sp}, respectively. We denote expectation, variance, discrete/differential entropy, and mutual information by $\mathbb{E}[\cdot]$, $\mathit{Var}(\cdot)$, $H(\cdot)$, and $I(\cdot)$, respectively.

B. Assumptions

Ideally, we want to learn a representation function f such that $f(X) = Z_{\text{inv}}$. However, due to a finite number of observed domains, it is possible that the learned features might contain spurious invariant features which are invariant for all observed domains, but change in the unseen domain [1] [2]. We model this situation by assuming that the representation function extracts features that are (approximately) composed of two elements: the (true) invariant features and the spurious invariant features:

$$f(X) = Z = \Theta(Z_{\text{inv}}, Z_{\text{sp}}).$$

Next, we state three assumptions on $Z_{\text{inv}}, Z_{\text{sp}}$ and Θ that we will use in Section IV to derive our theoretical results.

Assumption 1. *The (true) invariant features Z_{inv} are independent of the spurious invariant features Z_{sp} for a given label Y. Formally, $Z_{\text{inv}} \perp\!\!\!\perp Z_{\text{sp}} | Y$.*

Assumption 1 is widely accepted in the DG literature [10] [2] [12] [30]. For example, in the construction of the binary-MNIST dataset [10], the class (label) is first selected, then the color (spurious feature) is independently added to the handwritten digit (invariant feature) picked from the selected class, making $Z_{\text{inv}} \perp\!\!\!\perp Z_{\text{sp}} | Y$. For more details, we refer the reader to the third constraint in Section 3, page 5 of [10]. In [2] [12] and [30], this assumption is used but not explicitly stated. It is, however, implicit in Fig. 2 in [12], Fig. 3.1 in [2], and the discussion below Fig. 2 in [30].

Assumption 2. *The uncertainty of the invariant features is lower than the uncertainty of the spurious features when the label is known. Formally, we assume $H(Z_{\text{inv}}|Y) < H(Z_{\text{sp}}|Y)$.*

Assumption 2 has the following interesting clustering interpretation: invariant features are better clustered together in each class (have smaller variability) than spurious features. If additionally, $H(Z_{\text{inv}}) = H(Z_{\text{sp}})$, then $I(Z_{\text{inv}}; Y) = H(Z_{\text{inv}}) - H(Z_{\text{inv}}|Y) > H(Z_{\text{sp}}) - H(Z_{\text{sp}}|Y) = I(Z_{\text{sp}}; Y)$, implying that the invariant features Z_{inv} are more strongly related to the label Y than the spurious features Z_{sp}.

Assumption 3. $f(X) = Z = \Theta(Z_{\text{inv}}, Z_{\text{sp}}) = aZ_{\text{inv}} + bZ_{\text{sp}}$ *and* $\mathit{Var}(Z|Y) = \mathit{Var}(Z_{\text{inv}}|Y) = \mathit{Var}(Z_{\text{sp}}|Y) = 1$.

Assumption 3 states that the extracted (learned) features are a linear combination of invariant features and spurious features, i.e., $Z = \Theta(Z_{\text{inv}}, Z_{\text{sp}}) = aZ_{\text{inv}} + bZ_{\text{sp}}$. This is similar in spirit to the settings in [5] [12] and is inspired by methods for Blind Source Separation (BSS) such as Independent Component Analysis (ICA) [31]–[33] which aim to separate-out statistically independent latent component sources, say $S_1, S_2, S_1 \perp\!\!\!\perp S_2$, from observations of their *linear combination* $M = a_1S_1 + a_2S_2$. Our focus on a simple linear combination model enables us to derive some insightful theoretical results in the next section and translate them into a practical algorithm for filtering-out spurious features in the context of domain generalization which provides substantial performance improvements over competing alternatives. A general non-linear dependence relationship $Z = \Theta(Z_{\text{inv}}, Z_{\text{sp}})$ could potentially be handled using techniques such as non-linear ICA [34] or non-linear IRM [6] to filter out the spurious features. But we leave this to future work.

The assumption $\mathit{Var}(Z|Y) = \mathit{Var}(Z_{\text{inv}}|Y) = \mathit{Var}(Z_{\text{sp}}|Y) = 1$ is also motivated by an identical constraint in ICA needed to overcome the so-called *scaling* ambiguity: if $S_1 \perp\!\!\!\perp S_2$ and $M = a_1S_1 + a_2S_2$, then both (S_1, S_2) and (a_1S_1, a_2S_2) are pairs of independent component sources whose linear combination is M. Finally, it is worth noting that Assumption 1 and Assumption 3 together imply that $a^2 + b^2 = 1$ (see proof of Lemma 1).

IV. Main Results

Our proposed approach is based on two fundamental steps. The first step is to extract all the invariant features Z from source domains. These extracted invariant features may include both the true invariant features Z_{inv} and the spurious invariant features Z_{sp}. The next step is to remove the spurious features in order to construct a classifier that purely relies on the true invariant features Z_{inv}. For example, in the "cow-camel setting", the first step is to learn all extracted invariant features which might contain the color of the background. However, this spurious feature needs to be removed in the second step. In this section, we show that the CEM principle, i.e., minimizing $H(Z|Y)$, supports filtering-out the spurious invariant features.

Assumption 4. *Let*

$$\begin{aligned} f^* &= \arg\min_f L_{\text{invariant}}(f) \\ s.t.\quad H(f(X)|Y) &\leq \gamma. \end{aligned}$$

where $L_{\text{invariant}}$ is the loss function of an invariant representation learning algorithm. We assume that $L_{\text{invariant}}$ is such that for all γ, $Z = f^*(X)$ is a linear superposition of both the invariant feature Z_{inv} and the spurious feature Z_{sp}.

Under Assumption 4, our key idea to "eliminate" the contribution of Z_{sp} from Z by minimizing $L_{\text{invariant}}$ subject to a suitable bound on the uncertainty of Z given Y, i.e., designing a suitable value of γ. Indeed, we will show that there exists a suitable choice for γ for which f^* will extract only the (true) invariant feature Z_{inv} and filter-out Z_{sp}. The key result needed to show this is the following lemma.

Lemma 1. *If Assumptions 1, 2, 3 hold, then*

$$H(Z|Y) = H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y) \geq H(Z_{\text{inv}}|Y) \tag{4}$$

and equality holds in (4) if, and only if, $a = 1$ and $b = 0$.

Proof. Our proof of Lemma 1 is for differential entropy, but it can be easily extended to discrete entropy (recall that we use $H(\cdot)$ to denote discrete or differential entropy). Under Assumptions 1 and 3, we first show that $a^2 + b^2 = 1$. Indeed,

$$\begin{aligned} 1 &= \mathit{Var}(Z|Y) = \mathit{Var}(aZ_{\text{inv}} + bZ_{\text{sp}}|Y) \\ &= a^2\, \mathit{Var}(Z_{\text{inv}}|Y) + b^2\, \mathit{Var}(Z_{\text{sp}}|Y) \qquad (5) \\ &= a^2 + b^2, \qquad (6) \end{aligned}$$

where (5) is because $Z_{\text{inv}} \perp\!\!\!\perp Z_{\text{sp}}|Y$ and (6) is due to the assumption that $\mathit{Var}(Z_{\text{inv}}|Y) = \mathit{Var}(Z_{\text{sp}}|Y) = 1$.

Next, we utilize the result in Lemma 1 of [35] which states that for any two random variables R_1, R_2, and any two scalars a, b, if $R_1 \perp\!\!\!\perp R_2$ and $a^2 + b^2 = 1$, then:

$$H(aR_1 + bR_2) \geq a^2 H(R_1) + b^2 H(R_2). \tag{7}$$

Now, for a given $Y = y \in \mathcal{Y}$, we have:

$$\begin{aligned} &H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y = y) \\ &\quad\geq a^2 H(Z_{\text{inv}}|Y = y) + b^2 H(Z_{\text{sp}}|Y = y) \qquad (8) \\ &\quad= a^2 H(Z_{\text{inv}}|Y = y) + b^2 H(Z_{\text{inv}}|Y = y) \\ &\quad+ b^2 H(Z_{\text{sp}}|Y = y) - b^2 H(Z_{\text{inv}}|Y = y) \\ &\quad= H(Z_{\text{inv}}|Y = y) \\ &\quad+ b^2\big(H(Z_{\text{sp}}|Y = y) - H(Z_{\text{inv}}|Y = y)\big), \qquad (9) \end{aligned}$$

where (8) is due to (7) and $a^2 + b^2 = 1$ and (9) is because $a^2 + b^2 = 1$. Next,

$$\begin{aligned} &H(Z|Y) = H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y) \\ &= \int_{y\in\mathcal{Y}} p(y) H(aZ_{\text{inv}} + bZ_{\text{sp}}|Y = y)\, dy \\ &\geq \int_{y\in\mathcal{Y}} p(y) H(Z_{\text{inv}}|Y = y)\, dy \qquad (10) \\ &+ \int_{y\in\mathcal{Y}} p(y) b^2 \big(H(Z_{\text{sp}}|Y = y) - H(Z_{\text{inv}}|Y = y)\big)\, dy \\ &= H(Z_{\text{inv}}|Y) + b^2\big(H(Z_{\text{sp}}|Y) - H(Z_{\text{inv}}|Y)\big) \qquad (11) \\ &\geq H(Z_{\text{inv}}|Y) \qquad (12) \end{aligned}$$

where (10) follows from (9) and (12) from $H(Z_{\text{sp}}|Y) > H(Z_{\text{inv}}|Y)$ (Assumption 2). If $a = 1$ and $b = 0$ then $Z = Z_{\text{inv}}$ and equality holds. Conversely, if equality holds then $a = 1$ and $b = 0$ must hold, because otherwise we would have $b^2 > 0$ which together with $H(Z_{\text{sp}}|Y) > H(Z_{\text{inv}}|Y)$ and (11) would imply that $H(Z|Y)$ is strictly larger than $H(Z_{\text{inv}}|Y)$. Thus, equality $H(Z|Y) = H(Z_{\text{inv}}|Y)$ occurs if, and only if, $a = 1$ and $b = 0$, or equivalently, if, and only if $Z = Z_{\text{inv}}$. □

Lemma 1 shows that $H(Z|Y)$ is always lower bounded by $H(Z_{\text{inv}}|Y)$ and equality occurs if, and only if, $Z = Z_{\text{inv}}$. We use Lemma 1 to prove Theorem 1 which states that the CEM principle can be used to extract the (true) invariant features Z_{inv}.

Theorem 1. *If Assumptions 1, 2, 3, and 4 hold, then there exits a γ^* such that $f^*(X) = Z_{\text{inv}}$.*

Proof. From Assumption 4, for any γ, minimizing $L_{\text{invariant}}$ yields $Z = aZ_{\text{inv}} + bZ_{\text{sp}}$ for some values of a, b that depend on γ. We also have $\gamma \geq H(Z|Y) \geq H(Z_{\text{inv}}|Y)$, where the first inequality is due to the constraint in the minimization of $L_{\text{invariant}}$ and the second is from Lemma 1. If we choose $\gamma = \gamma^* := H(Z_{\text{inv}}|Y)$, then $H(Z|Y) = H(Z_{\text{inv}}|Y)$. From Lemma 1, $H(Z|Y) = H(Z_{\text{inv}}|Y)$ if, and only if, $b = 0$. Thus, selecting $\gamma^* = H(Z_{\text{inv}}|Y)$ will lead to a representation function f^* such that $f^*(X) = Z = Z_{\text{inv}}$. □

V. Practical Approach

We propose to find invariant features by solving the following CEM optimization problem:

$$\min_{h\in\mathcal{G}\circ\mathcal{F}} L_{CE-IRM}(h, \alpha, \beta) := L_{IRM}(h, \alpha) + \beta H(f(X)|Y). \tag{13}$$

This can be interpreted as the Lagrangian form of the optimization problem in Assumption 4 with $L_{\text{invariant}}$ replaced by the IRM loss function L_{IRM} in (3) and the conditional entropy constraint in Assumption 4 appearing as the second term with Lagrange multiplier β. The two hyper-parameters α and β control the trade-off between minimizing the Invariant Risk loss and minimizing the conditional entropy loss. Here, Y denotes the label, $h = g \circ f$ acts as an invariant predictor with $f \in \mathcal{F}$, $g \in \mathcal{G}$, and $Z = f(X)$ is the output of the penultimate layer of the end-to-end neural network that implements $h = g \circ f$, i.e., the layer just before the output layer. We note that Z and Y represent, respectively, the latent representations and the labels corresponding to the input data X from all seen domains *combined*.

In order to practically solve the optimization problem in (13), we leverage the implementations in [12] and [36]. Since

$$H(Z|Y) = H(Z) + H(Y|Z) - H(Y)$$

and $H(Y)$ is a data-dependent constant independent of $h = g \circ f$, the CEM optimization problem in (13) is equivalent to the following one

$$\min_{h \in \mathcal{G} \circ \mathcal{F}} L_{IRM}(h, \alpha) + \beta H(f(X)) + \beta H(Y|f(X)). \quad (14)$$

The first two terms of the objective function in (14) are identical to the objective function proposed in [12]. We therefore adapt the implementation in [12], which can be found at this link [2], to minimize the first two terms in (14). In order to optimize the third conditional entropy term $H(Y|Z)$, we adopt the variational characterization of conditional entropy described in [36]. A simple implementation of the variational method in [36] for minimizing of conditional entropy is available at this link[3].

VI. Experiments

In this section, we evaluate the efficacy of our proposed method on some DG datasets that contain spurious features.

A. Datasets

AC-CMNIST [5]. The Anti-causal-CMNIST dataset is a synthetic binary classification dataset derived from the MNIST dataset. It was proposed in [5] and is also used in [12]. There are three domains in AC-CMNIST: two training domains containing 25,000 data points each, and one test domain containing 10,000 data points. Similar to the CMNIST dataset [38], the images in AC-CMNIST are colored red or green in such a way that the color correlates strongly with the binary label in the two seen (training) domains, but is weakly correlated with the label in the unseen test domain. The goal is to identify whether the colored digit is less than five or more than five (binary label). Thus, in this dataset color is designed to be a spurious invariant feature. For a fair comparison, we utilize the same construction of AC-CMNIST dataset as in [5] [12].

CS-CMNIST [39]. The Covariate-Shift-CMNIST dataset is a synthetic classification dataset derived from CMNIST dataset. It was proposed in [39] and used in [12]. This dataset has three domains: two training domains containing 20,000 data points each and one test domain also containing 20,000 data points. We follow the construction method of [12] to set up a ten-class classification task, where the ten classes are the ten digits from 0 to 9, and each digit class is assigned a color that is strongly correlated with the label in the two seen training domains and is independent of the label in the unseen test domain. Details of the CS-CMNIST and the model for generating this dataset can be found in Section 7.2.1.A of [39]. For a fair comparison, we utilize the same construction methodology of the CS-CMNIST dataset as in [12].

Linear unit dataset (LNU-3/3S) [7]. The linear unit (LNU) dataset is a synthesic dataset that is constructed from a linear low-dimensional model for evaluating out-of-distribution generalization algorithms under the effect of spurious invariant features [7]. There are six sub-datasets in the LNU dataset, each sub-dataset consists of three or six domains, and each domain contains 10,000 data points. Due to limited time and space, we selected two sub-datasets from the LNU dataset named LNU-3 and LNU-3S to perform the evaluation. From the numerical results in [12], we note that LNU-3 and LNU-3S are the most challenging sub-datasets in the LNU dataset.

B. Methods Compared

We compare our proposed method, named Conditional Entropy and Invariant Risk Minimization (CE-IRM) against the following competing alternatives: (i) Empirical Risk Minimization (ERM) [37] as a simple baseline, (ii) the original Invariant Risk Minimization (IRM) algorithm in [5], (iii) the Information Bottleneck Empirical Risk Minimization (IB-ERM) algorithm in [12], and (iv) the Information Bottleneck Invariant Risk Minimization (IB-IRM) algorithm in [12]. We omit comparison with the algorithm proposed in [14] since their implementation was not available at the time our paper was submitted. Moreover, with the exception of the CS-CMNIST dataset where our method improves over theirs about 10% points, they do not report results for the other datasets that we used.

C. Implementation Details

We use the training-domain validation set tuning procedure in [12] for tuning all hyper-parameters. To construct the validation set, we split the seen data into a training set and a validation set in the ratio of 95% to 5% and select the model that maximizes classification accuracy on the validation set.

For AC-CMNIST, we utilize the learning model in [12] which is based on a simple Multi-Layer Perceptron (MLP) with two fully connected layers each having an output size 256 followed by an output layer of size two which aims to identify whether the digit is less than 5 or more than 5. The Adam optimizer is used for training with a learning rate of 10^{-4}, batch size of 64, and the number of epochs set to 500. To find the best representation, we search for the best values of weights of the Invariant Risk term and the Conditional Entropy term, i.e., α, β, respectively, among the following choices: $0.1, 1, 10, 10^2, 10^3, 10^4$.

For CS-CMNIST, we follow the learning model in [12] which is composed of three convolutional layers with feature map dimensions of 256, 128, and 64. Each convolutional layer is followed by a ReLU activation and batch normalization layer. The last layer is a linear layer that aims to classify the digit to 10 classes. We use the SGD optimizer for training with a batch size of 128, learning rate of 10^{-1} with decay over every 600 steps, and the total number of steps set to 2,000. Similarly to AC-CMNIST, we perform a search for the

[2] https://github.com/ahujak/IB-IRM

[3] https://github.com/1Konny/VIB-pytorch

TABLE I: Average accuracy in percentage (%) of compared methods. The number of classes in LNU-3/3S and AC-CMNIST datasets is 2 while the number of classes in CS-CMNIST dataset is 10. "#Domains" denotes the number of domains in the dataset.

Datasets	**#Domains**	**ERM** [37]	**IRM** [5]	**IB-ERM** [12]	**IB-IRM** [12]	**CE-IRM (proposed)**
CS-CMNIST	3	60.3 ± 1.2	61.5 ± 1.5	71.8 ± 0.7	71.8 ± 0.7	**85.7 ± 0.9**
LNU-3	6	67.0 ± 18.0	**86.0 ± 18.0**	74.0 ± 20.0	81.0 ± 19.0	84.0 ± 19.0
LNU-3S	6	64.0 ± 19.0	86.0 ± 18.0	73.0 ± 20.0	81.0 ± 19.0	**90.0 ± 17.0**
LNU-3	3	**52.0 ± 7.0**	**52.0 ± 7.0**	51.0 ± 6.0	**52.0 ± 7.0**	**52.0 ± 7.0**
LNU-3S	3	51.0 ± 6.0	51.0 ± 7.0	51.0 ± 6.0	51.0 ± 7.0	**52.0 ± 7.0**
AC-CMNIST	3	17.2 ± 0.6	16.5 ± 2.5	17.7 ± 0.5	**18.4 ± 1.4**	17.5 ± 1.3

weights of Invariant Risk and Conditional Entropy terms with $\alpha, \beta \in \{0.1, 1, 10, 10^2, 10^3, 10^4\}$.

For the LNU dataset, we follow the procedure described in [12]. Particularly, 20 pairs of α in the range $[1-10^{-0.3}, 1-10^{-3}]$, β in the range $[1-10^{0}, 1-10^{-2}]$, learning rate in the range $[10^{-4}, 10^{-2}]$, and weight of decay in the range $[10^{-6}, 10^{-2}]$ are randomly sampled and trained. The best model is selected based on the training-domain validation set tuning procedure.

We repeat the whole experiment five times by selecting five random seeds, where for each random seed, the whole process of tuning hyper-parameters and selecting models is repeated. Finally, the average accuracy and standard deviation values are reported. The source code of our proposed algorithm is available at this link.[4]

D. Results and Discussion

The results of all our computer experiments are shown in Table I. The numerical results of ERM, IRM, IB-ERM, and IB-IRM reported in Table I are taken from [12]. On the CS-CMNIST dataset, the four competing algorithms we tested achieve a classification accuracy in the range $60\% - 72\%$. But, our proposed CE-IRM algorithm vastly improves over the best alternative by almost 14% points. This can be explained by the way the CS-CMNIST is generated. Indeed, by construction, the colors (spurious features) are added independently into the digits (invariant features) for a given label. Therefore our assumption $Z_{\text{sp}} \perp\!\!\!\perp Z_{\text{inv}}|Y$ holds for the CS-CMNIST dataset.

For the LNU dataset, we followed the procedures in [12] to compute the classification error (equivalently accuracy) of the tested algorithms. We report the average accuracy together with its standard deviation in Table I. Similarly to [12], we compare all algorithms on the LNU-3 dataset and the LNU-3S dataset with the number of domains set to 6 or 3 (we used the same 3 domains as in [12]). For six domains, our CE-IRM algorithm outperforms all four competing methods by more than 4% points on the LNU-3S dataset, but is only second-best on the LNU-3 dataset about 2% point below the IRM algorithm. For three domains, the performance of all methods is very similar on both the LNU-3 and LNU-3S datasets. The results for the LNU-3 and LNU-3S datasets show that having more domains during training can improve the test accuracy of all algorithms.

Compared to the CS-CMNIST and the LNU-3/3S datasets, our results indicate that the AC-CMNIST is, by far, the most challenging dataset where none of the methods work well. Indeed, by construction, the AC-CMNIST contains strong spurious correlations between data and label leading to the failure of all tested algorithms. These results are consistent with those reported in [5], [12], and [14].

VII. Conclusions

We proposed a new DG approach based on the CEM principle for filtering-out spurious features. Our practical implementation combines the well-known IRM algorithm and the CEM principle to achieve competitive or better performance compared to the state-of-the-art DG methods. In addition, we showed that our objective function is closely related to the DIB method, and theoretically proved that under certain conditions, our method can truly extract the invariant features. We focused on the simple model where the features learned by an IRM algorithm are a linear combination of true and spurious invariant features. Our future work will focus on combining the non-linear IRM algorithm [6] with a nonlinear Blind Source Separation method, e.g., non-linear ICA [34], to accommodate non-linear mixture models of invariant features and spurious features.

VIII. Acknowledgment

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

[4] https://github.com/thuan2412/Conditional_entropy_minimization_for_Domain_generalization

References

[1] Y. Chen, E. Rosenfeld, M. Sellke, T. Ma, and A. Risteski, "Iterative feature matching: Toward provable domain generalization with logarithmic environments," *arXiv preprint arXiv:2106.09913*, 2021.

[2] E. Rosenfeld, P. Ravikumar, and A. Risteski, "The risks of invariant risk minimization," in *International Conference on Learning Representations*, vol. 9, 2021.

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

[4] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *arXiv preprint arXiv:2103.02503*, 2021.

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

[6] C. Lu, Y. Wu, J. M. Hernández-Lobato, and B. Schölkopf, "Nonlinear invariant risk minimization: A causal approach," *arXiv preprint arXiv:2102.12353*, 2021.

[7] B. Aubin, A. Słowik, M. Arjovsky, L. Bottou, and D. Lopez-Paz, "Linear unit-tests for invariance discovery," *arXiv preprint arXiv:2102.10867*, 2021.

[8] P. Kamath, A. Tangella, D. Sutherland, and N. Srebro, "Does invariant risk minimization capture invariance?" in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 4069–4077.

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

[10] V. Nagarajan, A. Andreassen, and B. Neyshabur, "Understanding the failure modes of out-of-distribution generalization," in *International Conference on Learning Representations*, 2020.

[11] M.-H. Bui, T. Tran, A. Tran, and D. Phung, "Exploiting domain-specific features to enhance domain generalization," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[12] K. Ahuja, E. Caballero, D. Zhang, J.-C. Gagnon-Audet, Y. Bengio, I. Mitliagkas, and I. Rish, "Invariance principle meets information bottleneck for out-of-distribution generalization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 3438–3450, 2021.

[13] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[14] B. Li, Y. Shen, Y. Wang, W. Zhu, C. J. Reed, J. Zhang, D. Li, K. Keutzer, and H. Zhao, "Invariant information bottleneck for domain generalization," *arXiv preprint arXiv:2106.06333*, 2021.

[15] Y. Du, J. Xu, H. Xiong, Q. Qiu, X. Zhen, C. G. Snoek, and L. Shao, "Learning to learn with variational information bottleneck for domain generalization," in *European Conference on Computer Vision*. Springer, 2020, pp. 200–216.

[16] D. Strouse and D. J. Schwab, "The deterministic information bottleneck," *Neural computation*, vol. 29, no. 6, pp. 1611–1630, 2017.

[17] N. H. Nazari and A. Kovashka, "Domain generalization using shape representation," in *European Conference on Computer Vision*. Springer, 2020, pp. 666–670.

[18] F. C. Borlino, A. D'Innocente, and T. Tommasi, "Rethinking domain generalization baselines," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9227–9233.

[19] R. Khirodkar, D. Yoo, and K. Kitani, "Domain randomization for scene-specific car detection and pose estimation," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1932–1940.

[20] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, "Deep domain-adversarial image generation for domain generalisation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 025–13 032.

[21] F.-E. Yang, Y.-C. Cheng, Z.-Y. Shiau, and Y.-C. F. Wang, "Adversarial teacher-student representation learning for domain generalization," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[22] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira *et al.*, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, p. 137, 2007.

[23] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

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

[25] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-draa, "Domain generalization with optimal transport and metric learning," *arXiv preprint arXiv:2007.10573*, 2020.

[26] B. Lyu, T. Nguyen, P. Ishwar, M. Scheutz, and S. Aeron, "Barycentric-alignment and invertibility for domain generalization," *arXiv preprint arXiv:2109.01902*, 2021.

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

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

[29] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, "Metareg: Towards domain generalization using meta-regularization," *Advances in Neural Information Processing Systems*, vol. 31, pp. 998–1008, 2018.

[30] E. C. Neto, "Causality-aware counterfactual confounding adjustment for feature representations learned by deep models," *arXiv preprint arXiv:2004.09466*, 2020.

[31] E. Oja and A. Hyvarinen, "Independent component analysis: A tutorial," *Helsinki University of Technology, Helsinki*, 2004.

[32] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[33] G. R. Naik and D. K. Kumar, "An overview of independent component analysis and its applications," *Informatica*, vol. 35, no. 1, 2011.

[34] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural networks*, vol. 12, no. 3, pp. 429–439, 1999.

[35] S. Verdú and D. Guo, "A simple proof of the entropy-power inequality," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 2165–2166, 2006.

[36] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[37] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

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

[39] K. Ahuja, J. Wang, A. Dhurandhar, K. Shanmugam, and K. R. Varshney, "Empirical or invariant risk minimization? a sample complexity perspective," in *International Conference on Learning Representations*, 2020.