Environment Diversification with Multi-head Neural Network for Invariant Learning

Bo-Wei Huang Keng-Te Liao Chang-Sheng Kao Shou-De Lin
Department of Computer Science and Information Engineering
National Taiwan University, Taiwan
{r10922007,d05922001,b07902046,sdlin}@csie.ntu.edu.tw

Abstract

Neural networks are often trained with empirical risk minimization; however, it has been shown that a shift between training and testing distributions can cause unpredictable performance degradation. On this issue, a research direction, invariant learning, has been proposed to extract causal features insensitive to the distributional changes. This work proposes EDNIL, an invariant learning framework containing a multi-head neural network to absorb data biases. We show that this framework does not require prior knowledge about environments or strong assumptions about the pre-trained model. We also reveal that the proposed algorithm has theoretical connections to recent studies discussing properties of variant and invariant features. Finally, we demonstrate that models trained with EDNIL are empirically more robust against distributional shifts.

1 Introduction

Ensuring model performance on unseen data is a common yet challenging task in machine learning. A widely adopted solution would be empirical risk minimization (ERM), where training and testing data are assumed to be independent and identically distributed. However, data in real-world applications can come with undesired biases, causing a shift between training and testing distributions. It has been known that the distributional shifts can severely harm ERM model performance and even cause the trained model to be worse than random predictions [10]. In this work, we focus on Invariant Learning, which aims at learning causal features expected to be robust against distributional changes. Invariant Risk Minimization (IRM) [1] has been proposed as a popular solution for invariant learning. Specifically, IRM is based on an assumption that training data are collected from multiple sources or environments having distinct data distributions. The learning objective is then designed as a standard ERM loss function with a penalty term constraining the trained model (e.g. classifier) to be optimal in all the environments.

IRM has shown to be effective; however, we note that IRM and many invariant learning methods rely on strict assumptions which limit the practical impacts. The limitations are summarized as follows.

Prior Knowledge of Environments IRM assumes training data are collected from environments, and the environment labels (i.e. which data instance belongs to which environment) are given. However, the environment labels are often unavailable. Moreover, the definition of environments can be implicit, making human labeling more difficult and expensive. To find environments without supervision, Creager et al. [6] propose EILL, a min-max optimization framework training the model of interest and inferring the environment labels. Another work, HRM [22] (or the extension, KerHRM [23]), parameterizes the environments and proposes clustering-based approaches to estimate the parameters. A recent method, ZIN [20], also learns to label data. However, it relies on carefully chosen features satisfying a series of theoretical constraints, and thus human efforts are still required.
Delicate Initialization  
EIIL is able to infer environments but requires an ERM model for initialization. Crucially, the ERM model should heavily depend on spurious correlations. Creager et al. reveal that, for example, slightly underfitted ERM models may encode more spurious relationships in some cases. However, as the distributional shifts are assumed to be unknown in the training stage, appropriate initialization might be difficult to guarantee.

Efficiency Issue  
HRM and KerHRM, though do not possess the above two limitations, suffer from the efficiency issue. Specifically, HRM is assumed to be trained with low dimensional data. As for KerHRM, although it extends HRM to avoid the issue of dimensions by adopting kernel methods, the computational costs of the proposed method can be very high if the data or model size is large.

This work proposes a novel framework, Environment Diversification with multi-head neural Network for Invariant Learning (EDNIL). EDNIL is able to infer environment labels without supervision and achieve joint optimization of environment inference and invariant learning models. The underlying multi-head neural network explicitly diversifies the inferred environments, which is consistent with recent studies revealing the benefits of diverse environments. Notably, the proposed neural network is functionally similar to a multi-class classifier and can be optimized efficiently. The advantages of EDNIL are summarized as:

- We implement this framework using various pre-trained models such as Resnet and DistilBert, and evaluate it with diverse data types and varied biases. The results show that EDNIL can constantly outperform the existing state-of-the-art methods.

- The learning algorithm of EDNIL has theoretical connections to recent studies discussing conditions of ideal environments.

- EDNIL does not have the three limitations discussed above. The comparisons between EDNIL and other methods are shown in Table 1.

<table>
<thead>
<tr>
<th>Unsupervised</th>
<th>Insensitive Initialization</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRM [1]</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>ZIN [20]</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>EIIL [6]</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>HRM [22]</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>KerHRM [23]</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>EDNIL (Ours)</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### 2 Preliminaries and Related Works

The goal of EDNIL is to tackle out-of-distribution problems with invariant learning in the absence of manual environment labels. In Section 2.1, background knowledge about out-of-distribution generalization and invariant learning are introduced. In Section 2.2, we discuss recent studies investigating ideal environments. In Section 2.3, we introduce the existing unsupervised methods inferring environments.

#### 2.1 Out-of-distribution Generalization and Invariant Learning

Following [1, 22], we consider a dataset $D = \{D^e\}_{e \in \text{supp}(E_{\text{tr}})}$ with different sources $D^e = \{(x_{te}^i, y_{te}^i)\}_{i=1}^{n_e}$ collected under multiple training environments $e \in \text{supp}(E_{\text{tr}})$. Random variable $E_{\text{tr}}$ indicates the training environment labels. For simplicity, $X^e, Y^e$ and $P^e$ denote data, target label and distribution in environment $e$ respectively.

With $E_{\text{all}}$ containing all possible environments such that $\text{supp}(E_{\text{all}}) \supset \text{supp}(E_{\text{tr}})$, the goal of out-of-distribution generalization is to learn a predictor $f(\cdot) : X \to Y$ as Equation [1] where $R^e(f) = \mathbb{E}_{X^e, Y^e}[(f(X^e), Y^e)] = \mathbb{E}_{Y^e}[(f(X^e), Y^e)]$ measures the risk under environment $e$ with any loss.

\[ \text{A method is \textit{unsupervised} if it does not require extra human efforts to obtain environments.} \]
function \( l(\cdot, \cdot) \). In general, for \( e \in \text{supp}(\mathcal{E}_e) \) and \( e' \in \text{supp}(\mathcal{E}_{e'}) \setminus \text{supp}(\mathcal{E}_e) \), \( P^{e'}(X, Y) \) is rather different from \( P^e(X, Y) \).

\[
C = \arg \max_{c} \sum_{e \in \text{supp}(\mathcal{E}_e)} R^e(c) + \lambda \| \nabla w |w=1.0 \| R^e(w \circ \Phi) \| 2 \quad (2)
\]

As \( w \) is a dummy classifier, the encoder \( \Phi \) is also regarded as a predictor.

### 2.2 Ideal Environments

As \( \mathcal{E}_e \) is unavailable or sub-optimal in most applications, learning to find appropriate environments (denoted by \( \mathcal{E}_{\text{learn}} \)) is attractive. However, the challenge is the lack of knowledge of valid environments. Recently, Lin et al. \[20\] have proposed Equation \[3\] and \[4\] as the conditions of ideal environments, where \( H \) is conditional entropy. To satisfy the conditions, Lin et al. \[20\] propose leveraging auxiliary information for model training. However, the method still requires extra human efforts to collect and verify the additional information.

\[
H(Y | X_e) = H(Y | X_e, \mathcal{E}_{\text{learn}}) \quad (3)
\]

\[
H(Y | X_e) - H(Y | X_e, \mathcal{E}_{\text{learn}}) > 0 \quad (4)
\]

Particularly, Equation \[4\] can be implied by empirical studies \[5\] where diversity of environments is recognized as the key to obtaining effective IRM models. To be more precise, large discrepancy of spurious correlations, or \( P^e(Y | X_e) \), between environments is favored. As the environments give a clear indication of distributional shifts, IRM can easily identify and eliminate variant features. Beyond IRM, HRM \[22\] and KerHRM \[23\] can also be viewed as optimizing diversity via clustering-based methods specifically.

### 2.3 Unsupervised Environment Inference

Here we provide a detailed introduction of the existing environment inference methods that do not require extra human efforts. The general idea is to integrate environment inference with invariant learning algorithms who require provided environments.

EIIL \[8\] proposes formulating invariant learning as a min-max optimization problem. Specifically, EIIL is composed of two objectives, \( \text{Environment Inference (EI)} \) and \( \text{Invariant Learning (IL)} \), where \( EI \) is optimized by maximizing the training penalty via labeling the data, and \( IL \) is optimized by minimizing the training loss given the data labeled by \( EI \). The two-stage framework bypasses the difficulty of defining environments; however, the training result heavily relies on the initialization of the \( EI \) optimization. Specifically, the initialization demands a strongly biased ERM reference model; otherwise, EIIL can have a significantly weaker performance.

Another method, HRM \[22\], proposes a clustering-based method for learning plausible environments. HRM assumes that spurious correlations in each environment can be modeled by a parameterized function and the dataset is generated by the mixture of the functions. The parameters are then estimated by employing EM algorithm. Additionally, HRM equips a joint learning framework which alternatively learns invariant predictors and improves quality of clustering results. However, a known
issue of HRM is an assumption that the data are represented by raw features. Data such as images and texts requiring non-linear neural networks to obtain representations are beyond the capability.

To extend HRM to a broader class of applications and improve the model performance, Liu et al. [23] propose KerHRM. The main idea is to adopt the Neural Tangent Kernel [15] method which transforms non-linear neural network training into a linear regression problem on the proposed Neural Tangent Features space. As a result, KerHRM elegantly resolves the shortcomings of HRM and is shown to be more effective. However, the proposed method and its implementations bring additional computational costs depending on data and model capacity. For applications favoring large datasets and pre-trained models, such as Resnet [14] and BERT [7], KerHRM may not be an affordable option at the present stage.

3 Methodology

In this section, we propose a general framework to learn invariant model without manual environment labels. As shown in Figure 1, our proposed method consists of two models, $M_{EI}$ and $M_{IL}$. Given the pooled data $(X, Y)$, $M_{EI}$ infers environments $E_{learn}$ satisfying Condition 3 and 4, and $M_{IL}$ is an invariant model trained with the inferred environments. Our framework is jointly optimized with alternating updates. The learned $M_{IL}$ can provide information of invariant features to $M_{EI}$, so that Condition 3 and 4 can be fulfilled simultaneously. Note that $M_{EI}$ only serves at train time to provide environments for invariant learning. At test time, only $M_{IL}$ is needed to perform invariant predictions.

3.1 The Environment Inference Model

The target of environment inference is to partition data with environment labels $E_{learn}$ satisfying Condition 3 and 4. In this regard, we propose a graphical model, which is a sufficient condition for Condition 3 and 4 (the proof is in Appendix A), as our foundation of inference model and learning objectives. The graph is shown in Figure 2, where the data generation process follows the proposed example in [1].

The inference model, $M_{EI}$, is a neural network learning to realize the underlying graphical model. Following the idea of parameterizing environments from HRM [22] and KerHRM [23], we assume the distinct mapping relation between $X$ and $Y$ in environment $e$ can be modeled by a function $f^e(\Psi(X))$, where $\Psi(X)$ is learned representations expected to encode variant features $X$, and $f^e$ is an environmental function responsible for predicting $Y$. Instead of employing clusters, we propose building a multi-head neural network as shown in Figure 1b, particularly, a single-head network in $M_{EI}$ with shared parameters corresponds to a cluster center in HRM or KerHRM. The training procedure of $M_{EI}$ can be divided into inference stage and learning stage.
3.1.1 Inference Stage of $M_{EI}$

The goal is to infer an environment label for each training data. As in the graphical model, $\mathcal{E}_{\text{learn}}$ is associated with variant relationships. Inspired by multi-class classification problem, we propose Equation (5) where the probability $P(e \mid X, Y)$ is estimated via a softmax of negative $l(f^e(\Psi(X)), Y)$ divided by a constant temperature $\tau$. The function $l$ is expected to be the commonly used cross entropy or mean squared error that measures the discrepancy between $Y$ and $f^e(\Psi(X))$ for each environment $e$. Intuitively, each data prefers the environment whose model has better prediction.

$$P(\mathcal{E}_{\text{learn}} = e \mid X, Y) = \frac{\exp (-l(f^e(\Psi(X)), Y)/\tau)}{\sum_{e' \in \text{supp}(\mathcal{E}_{\text{learn}})} \exp (-l(f^{e'}(\Psi(X)), Y)/\tau)} \quad (5)$$

3.1.2 Learning Stage of $M_{EI}$

The goal is to update the neural network to improve the quality of inference. Based on the structure of the graphical model, three losses are designed for minimization, i.e. **Environment Diversification Loss ($L_{\text{ED}}$)**, **Label Independence Loss ($L_{\text{L1}}$)** and **Invariance Preserving Loss ($L_{\text{IP}}$)**. In particular, $L_{\text{ED}}$ and $L_{\text{IP}}$ correspond to the concepts of Condition 1 and 2 respectively.

Environment Diversification Loss ($L_{\text{ED}}$) We consider maximizing $H(Y \mid X) - H(Y \mid X, \mathcal{E}_{\text{learn}})$ to satisfy Condition 1 and capture more diverse variant relationships. Given the estimated $P(\mathcal{E}_{\text{learn}} \mid X, Y)$, $L_{\text{ED}}$ selects the most probable environment and its corresponding network for optimization:

$$L_{\text{ED}} = -\sum_{i} w_i \max_{e} \log P(e \mid x_i, y_i) \quad (6)$$

For each data $(x_i, y_i)$, although only one environment $e_i = \arg \max_e P(e \mid x_i, y_i)$ is selected for the minimization, the softmax simultaneously propagates gradient to maximize $l(f^e(\Psi(x_i)), y_i)$ for $e' \neq e_i$. The network learns to maximize the dependency between $\mathcal{E}_{\text{learn}}$ and $Y$ given variant representations. In terms of spurious correlations, the distinction between environments is expected to become clearer accordingly. In practice, we utilize scaling weight $w_i$ inversely proportional to the size of $e_i$. The importance of smaller environments will be thus enhanced within the summation.

Label Independence Loss ($L_{\text{L1}}$) With d-separation [25], $\mathcal{E}_{\text{learn}}$ is independent of $Y$ in the graphical model. Hence, $L_{\text{L1}}$ constrains their dependency measured by the mutual information $I(Y; \mathcal{E}_{\text{learn}})$. Empirically, $L_{\text{L1}}$ prevents a trivial solution that environments are determined purely by target labels $Y$ regardless of input features $X$, which is undesirable for invariant learning. To minimize $I(Y; \mathcal{E}_{\text{learn}})$, it can be verified that it is equivalent to minimizing Equation (7) given $P(Y)$ is known.

$$L_{\text{L1}} = \mathbb{E}_{e \sim P(\mathcal{E}_{\text{learn}})} \left[ \sum_{y} P(y \mid e) \log P(y \mid e) \right] \quad (7)$$

Invariance Preserving Loss ($L_{\text{IP}}$) For Condition 3 as $M_{IL}$ learns some invariant relationships after several training steps (Section 3.2), $L_{\text{IP}}$ can be considered to exclude invariant features from the diversification. Specifically, designed as the contrary of $L_{\text{ED}}$, $L_{\text{IP}}$ limits the variance of expected loss from invariant predictor $\Phi$ (in $M_{IL}$) across environments (Equation 8). Similar idea can be found in [17]. However, instead of training invariant model given known environments, we freeze the invariant predictor and regularize the adjustment of $\mathcal{E}_{\text{learn}}$ (i.e. the updates of $M_{EI}$) here.

$$L_{\text{IP}} = \text{Var}_{e \sim P(\mathcal{E}_{\text{learn}})} \left[ \mathbb{E}_{e} [l(\Phi(X^e), Y^e)] \right] \quad (8)$$

In summary, the training loss of $M_{EI}$ can be summarized as **Environment Inference Loss ($L_{\text{EI}}$)**. The regularization strengths of $L_{\text{L1}}$ and $L_{\text{IP}}$ can be controlled by hyper-parameters $\beta$ and $\gamma$ respectively:

$$L_{\text{EI}} = L_{\text{ED}} + \beta L_{\text{L1}} + \gamma L_{\text{IP}} \quad (9)$$

In addition, before minimizing $L_{\text{EI}}$, we pre-train our $\Psi$ and one arbitrary $f^e$ with ERM. In general, it empirically facilitates better feature extraction. Unlike EII [6] taking ERM as a reference model heavily relying on variant features, EDNIL performs more consistently under various choices of ERM. Namely, the initialization of EDNIL can be more arbitrary than that of EII. We verify the argument in Section 3.4.
3.2 The Invariant Learning Model

To identify invariance across environments, IRM [1] is selected as our base algorithm optimizing the invariant predictor \( \Phi \) in our model \( M_{IL} \). As for the required environment partitions during training, we assign environment label \( e \in \text{supp}(E_{\text{learn}}) \) with largest \( P(e|x_i, y_i) \), inferred by \( M_{EL} \) (Section 3.1.1), to each data \((x_i, y_i)\). However, it is inevitable that there exist some noises in automatically inferred environments, especially in the beginning of joint optimization. To reduce the impact of immature environments on invariant learning, we calculate the confidence score \( c_e \) for each environment \( e \in \text{supp}(E_{\text{learn}}) \), i.e. \( \mathbb{E}[P(e|X^e, Y^e)] \). Our training objective is modified to minimize Invariant Learning Loss \( L_{IL} \) that considers the weighted average of environmental losses:

\[
L_{IL} = \sum_{e \in \text{supp}(E_{\text{learn}})} w_e \cdot \left[ R^e(\Phi) + \lambda \left\| \nabla_{w|w=1.0} R^e(w \circ \Phi) \right\|^2 \right]
\]

\[
w_e = \frac{c_e}{\sum_{e' \in \text{supp}(E_{\text{learn}})} c_{e'}}
\]

4 Experiments

We empirically validate the proposed method on biased datasets, Adult-Confounded, CMNIST, Waterbirds and SNLI. The generation of spurious correlations mainly follow the protocols proposed by [1, 6, 9, 29]. In Section 4.1, Adult-Confounded and CMNIST are tested with Multilayer Perceptron (MLP). In Section 4.2, two more complex datasets, Waterbirds and SNLI, are taken for evaluating the integration of transfer learning. Deep pre-trained models will be fine-tuned to discover variant and invariant representations.

The following four methods are selected as our competitors: Empirical Risk Minimization (ERM), Environment Inference for Invariant Learning (EIIL [6]), Kernelized Heterogeneous Risk Minimization (KerHRM [23]) and Invariant Risk Minimization (IRM [1], Equation 2). EIIL and KerHRM are invariant learning methods with unsupervised environment inference, which share the same settings as EDNIL. HRM [22] is replaced by KerHRM for non-linearity. For IRM who requires environment partitions, we re-label \( E_{\text{oracle}} \) on each given biased training set, which diversifies the spurious relationships to elicit upper-bound performance of IRM.

For hyper-parameter tuning, we split 10% of training data to construct an in-distribution validation set. In each dataset, several testing environments with different distributions are listed to evaluate the robustness of each method, and we mainly take worst-case performance for assessment. As all tasks in this section are classification problems, accuracy is adopted as the evaluation metric.

Besides, more experimental details are revealed in Appendix B. We also discuss additional experiments in Appendix C including solutions to regression problem and model stability given different spurious strengths at train time.

4.1 Simple Datasets with MLP

This section includes two simple datasets, Adult-Confounded and CMNIST, where spurious correlations are synthetically produced with the predefined strengths. For all competitors, MLP is taken as the base model and full-batch training is implemented. Since KerHRM performs unstably over random seeds, we first average the results after 10 runs as its first score, and select top 5 among them as the second one, which will be marked with an asterisk (*) in each table.

4.1.1 Discussions on Adult-Confounded

We take UCI Adult [16] to predict binarized income levels (above or below $50,000 per year)\(^3\). Following [6], individuals are re-sampled according to sensitive features race and sex to simulate spurious correlations. Specifically, with binarized race (Black/Non-Black) and sex (Female/Male), four possible subgroups are constructed: Non-black Males (SG1), Non-black Females (SG2), Black Males (SG3), and Black Females (SG4). Keeping original train/test split and subgroup sizes from

\(^3\)UCI Adult dataset is widely used in algorithmic fairness papers. However, a recent study [8] discusses some limitations of this dataset, such as the choice of income threshold.
We report our evaluation on a noisy digit recognition dataset, CMNIST. Following with Ablation Study for worst-case performance.

Table 2: $P(Y = 1|SG)$ for Adult-Confounded. IID shares spurious correlations with the train set. IND has no bias on race and sex. OOD defines the worst-case performance.

<table>
<thead>
<tr>
<th>SG1</th>
<th>SG2</th>
<th>SG3</th>
<th>SG4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>0.5</td>
<td>0.9</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9</td>
<td>0.5</td>
<td>0.1</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 3: Testing accuracy (%) on Adult Confounded. Three subsets are defined in Table 2, $\beta = 0$ and $\gamma = 0$ indicate the removal of $L_1$ and $L_1$ when training $M_{II}$.

<table>
<thead>
<tr>
<th>IID</th>
<th>IND</th>
<th>OOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>92.4 ± 0.1</td>
<td>66.8 ± 0.3</td>
</tr>
<tr>
<td>EIIIL</td>
<td>76.2 ± 0.4</td>
<td>73.5 ± 0.5</td>
</tr>
<tr>
<td>KerHRM</td>
<td>82.4 ± 3.9</td>
<td>75.1 ± 4.0</td>
</tr>
<tr>
<td>KerHRM*</td>
<td>81.2 ± 1.8</td>
<td>78.5 ± 0.3</td>
</tr>
<tr>
<td>EDNIL</td>
<td>80.7 ± 0.4</td>
<td>79.1 ± 0.4</td>
</tr>
<tr>
<td>EDNIL$_{\beta=0}$</td>
<td>91.8 ± 0.0</td>
<td>67.6 ± 0.1</td>
</tr>
<tr>
<td>EDNIL$_{\gamma=0}$</td>
<td>78.2 ± 2.4</td>
<td>75.4 ± 1.6</td>
</tr>
<tr>
<td>IRM</td>
<td>79.9 ± 0.4</td>
<td>79.3 ± 0.3</td>
</tr>
</tbody>
</table>

UCI Adult, we sample data based on the given target distributions in each sensitive subgroup as shown in Table 2. In particular, OOD contributes the worst-case performance to validate if the predictions rely on group information. In this task, MLP with one hidden layer of 96 neurons is considered. For IRM, four environments comprise $E_{\text{oracle}}$, where the correlations between variant features $(\text{race}, \text{sex})$ and target $Y$ are distributed without overlapping. More details are provided in Appendix B.

Results  The results are shown in Table 3. With strong spurious correlations at train time, ERM obtains high accuracy as the correlations remain aligned; however, its generalization to other testing distributions is limited. Among all invariant learning methods without prior environment labels, EDNIL can perfectly identify variant features and infer ideally diversified environments. Therefore, it achieves the most invariant performance over different testing distributions. EIIIL can improve consistency to some degree, but not as strong as EDNIL. A possible reason is that empirically trained reference model is not guaranteed to be purely variant. A possible reason is that empirically trained reference model is not guaranteed to be purely variant. For KerHRM, it performs inconsistently across random seeds, which is reflected in the large standard deviation. In some cases, the performance hardly improves over iterations, as observed by Liu et al. [23].

Ablation Study for $L_{II}$  We first claim the importance of $L_1$, which constraints label dependency, by setting the coefficient $\beta$ to zero. As discussed in Section 3, the resulting environments are determined purely by target labels, and thus lead to inferior performance for invariant learning as shown in Table 2. Next, we demonstrate the effectiveness of joint optimization in Figure 3a. The regularization $L_1$ promotes environment inference, so that the worst-case performance improves and remains stable over iterations. According to Table 3, if the coefficient $\gamma$ is turned off, feedback generated by $M_{II}$ will be ignored and the effect of invariant learning risks being undesirable.

4.1.2 Discussions on CMNIST

We report our evaluation on a noisy digit recognition dataset, CMNIST. Following 1, we first assign $Y = 1$ to those whose digits are smaller than 5 and $Y = 0$ to the others. Next, we apply label noise by randomly flipping $Y$ with probability 0.2. Finally, the digits are colored with color label $C$, which is generated by randomly flipping $Y$ with probability $e$. For training, two equal-sized environments with $e = 0.1$ and $e = 0.2$ are merged, which is equivalent to one with $e = 0.15$ on average. For testing, three situations are considered when $e$ is 0.1, 0.5 or 0.9, respectively. Note that when $e = 0.1$, the spurious correlation is much aligned with the training set. On the other hand, $e = 0.9$ defines the most challenging case since the spurious correlation shifts most dramatically from training.

For all competitors except KerHRM, we select MLP with two hidden layers of 390 neurons, and consider the whole dataset (50,000 samples) for training. For KerHRM who requires massive computing resources, we follow the settings recommended by 23. Specifically, we randomly select 5,000 samples and train MLP with one hidden layer of 1024 neurons. To construct ideally diversified $E_{\text{oracle}}$ for IRM, we pack all examples with $C = \bar{Y}$ into one environments, and $C \neq Y$ into the other.

Results  The results are shown in Table 4. First of all, not surprisingly ERM still adopts poorly to distributional shifts. Among all invariant learning methods without manual environment labels,
EDNIL gets closest to IRM with $E_{\text{oracle}}$, achieving consistent and robust performance in this dataset. As shown in Figure 3c, EDNIL provides almost ideally diversified $E_{\text{learn}}$ for invariant learning.

**Table 4:** Testing accuracy (%) of CMNIST, where color noise 0.9 makes the worst-case environment.

<table>
<thead>
<tr>
<th>Color Noise</th>
<th>0.1</th>
<th>0.5</th>
<th>0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>88.4 ± 0.3</td>
<td>55.0 ± 0.5</td>
<td>21.7 ± 0.8</td>
</tr>
<tr>
<td>EIIL</td>
<td>79.6 ± 0.3</td>
<td>71.7 ± 0.7</td>
<td>63.1 ± 0.5</td>
</tr>
<tr>
<td>KerHRM</td>
<td>74.3 ± 0.7</td>
<td>66.2 ± 1.7</td>
<td>58.0 ± 11.5</td>
</tr>
<tr>
<td>KerHRM*</td>
<td>71.3 ± 0.7</td>
<td>68.5 ± 0.5</td>
<td>66.1 ± 0.7</td>
</tr>
<tr>
<td>EDNIL</td>
<td>77.7 ± 0.4</td>
<td>76.8 ± 0.3</td>
<td>75.2 ± 0.4</td>
</tr>
<tr>
<td>IRM</td>
<td>77.8 ± 0.4</td>
<td>76.8 ± 0.4</td>
<td>75.2 ± 0.3</td>
</tr>
</tbody>
</table>

**4.2 Complex Datasets with Pre-trained Deep Learning Models**

This section extends MLP to deep learning models with pre-trained weights for more complex data. With mini-batch fine-tuning, we consider all competitors but exclude KerHRM due to the efficiency issue. In Section 4.2.1, image dataset, Waterbirds [29], with controlled spurious correlations is selected for evaluating the generalization on more high-dimensional images. In Section 4.2.2, a real-world NLP dataset, SNLI [3], is considered. The biases in SNLI are naturally derived from the procedure of data collection, and we define biased subsets for evaluation following Dranker et al. [9].

**4.2.1 Discussions on Waterbirds**

In Waterbirds [29], each bird photograph, from CUB dataset [31], is combined with one background image, from Places dataset [33]. Both birds and backgrounds are either from land or water, and our target is to predict the binarized species of birds. At train time, landbirds and waterbirds frequently present in land and water backgrounds respectively. Therefore, empirically trained models are prone to learn context features, and fail to generalize as background varies [2, 6, 10, 21, 29].

To split a validation set whose overall distribution is i.i.d. to the training set, we merge original training and validation data and split 10% for hyper-parameter tuning. For testing, we observe all four combinations of birds and backgrounds in the original testing set. Among them, the minor subgroup (waterbirds on land) contributes the most challenging case. In this task, Resnet-34 [14] is chosen for mini-batch fine-tuning. Given two binary labels (target, background), we distribute $target = background$ and $target \neq background$ into two different environments and apply balanced class weights for the oracle settings of IRM.

**Results** The results are shown in Table 5. As observed in [6, 29], ERM suffers in the hardest case (i.e. waterbirds on land). EIIL also performs poorly in this case. With a more sophisticated learning
Table 5: Testing accuracy (%) of Waterbirds, where Y and BG means target and background respectively. The subgroup (Water, Land) contributes the worst-case performance.

<table>
<thead>
<tr>
<th>(Y, BG)</th>
<th>(Land, Land)</th>
<th>(Water, Water)</th>
<th>(Land, Water)</th>
<th>(Water, Land)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>99.4 ± 0.0</td>
<td>91.4 ± 0.2</td>
<td>90.9 ± 0.8</td>
<td>72.8 ± 1.0</td>
</tr>
<tr>
<td>EIIL</td>
<td>99.4 ± 0.3</td>
<td>90.5 ± 1.8</td>
<td>89.3 ± 3.7</td>
<td>68.6 ± 5.4</td>
</tr>
<tr>
<td>EDNIL</td>
<td>98.5 ± 0.6</td>
<td>89.9 ± 1.3</td>
<td>90.3 ± 3.0</td>
<td>78.6 ± 4.3</td>
</tr>
<tr>
<td>IRM</td>
<td>98.0 ± 0.5</td>
<td>90.6 ± 1.1</td>
<td>89.5 ± 1.7</td>
<td>83.2 ± 2.2</td>
</tr>
</tbody>
</table>

Choice of Initialization  Both EIIL and EDNIL take ERM as initialization. As mentioned in Section 1, heavy dependency on initialization is risky when testing distribution is unavailable. Therefore, we take ERM with different training steps for EIIL and EDNIL to verify the stability. The results are shown in Figure 56. As suggested by 6, EIIL works only with underfitted reference model in this case. If the reference model is more well-trained, the performance of EIIL will greatly decline since ERM might get distracted from variant features. One is prone to be misled into an undesirable choice for EIIL when seeking hyper-parameters without prior knowledge of distributional shifts. For instance, the validation score of EIIL with 500-step reference model (95.8%) is higher than that with 100-step (94.7%), which is not consistent with their performances on testing. In comparison, EDNIL performs more consistent across different pre-training steps, which accentuates our strength of less sensitive initialization.

4.2.2 Discussions on SNLI

The target of SNLI 3 is to predict the relation between two given sentences, premise and hypothesis. Recent studies 12, 24, 27 reveal hypothesis bias in SNLI, which is characterized by patterns in hypothesis sentences highly correlated with a specific label. One can achieve low empirical risk without considering premises during prediction. However, as the bias no longer holds, the performance degradation occurs 11, 24.

We sample 100,000 examples and consider all classes, entailment, neutral and contradiction, for our experiment. Following 29, we define three subsets, unbiased, bias aligned and bias misaligned, by training a biased model with hypothesis as its only input. The specification of the subsets is as follows:

- Unbiased: Examples whose predictions from the biased model are ambiguous
- Aligned: Examples that the biased model can predict correctly with high confidence
- Misaligned: Examples that the biased model can predict incorrectly with high confidence

The proportions of the three subsets are 17%, 67% and 16%, respectively. Due to the minority, the bias misaligned subset is more likely to be ignored and thus defines the worst-case performance.

For all methods, DistilBERT 30 is taken as the pre-trained model for further mini-batch fine-tuning. For Eoracle, we assign the bias aligned subset to the first environment, and the bias misaligned subset to the second. In order to make bias prevalence equal in the two environments, unbiased samples are scattered proportionally to the two environments.

Results  The results are shown in Table 6. As reported by 32, ERM receives higher score on the bias aligned subset, but it fails in the bias misaligned case. Among all invariant learning methods without environment labels, only EDNIL improves on the bias misaligned subset. Namely, even though the definitions of biases are at a high level, EDNIL is still capable of encoding and diversifying possible variant features.

Table 6: Testing accuracy (%) on SNLI, where the misaligned defines the worst-case performance.

<table>
<thead>
<tr>
<th>Subset</th>
<th>Unbiased</th>
<th>Aligned</th>
<th>Misaligned</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>74.6 ± 0.3</td>
<td>94.7 ± 0.2</td>
<td>52.6 ± 0.9</td>
</tr>
<tr>
<td>EIIL</td>
<td>74.2 ± 0.3</td>
<td>95.0 ± 0.1</td>
<td>51.7 ± 1.3</td>
</tr>
<tr>
<td>EDNIL</td>
<td>74.3 ± 0.8</td>
<td>94.2 ± 0.2</td>
<td>54.5 ± 1.0</td>
</tr>
<tr>
<td>IRM</td>
<td>74.0 ± 0.9</td>
<td>92.3 ± 0.5</td>
<td>56.9 ± 1.1</td>
</tr>
</tbody>
</table>
5 Limitations

Our learning algorithm for environment inference is based on the graphical model plotted in Figure 2. As data are not necessarily generated by the presumed process, there can exist biases that cannot be captured by the proposed neural network. In the paper, we provide empirical studies of effectiveness on diverse datasets, while we are still aware that a stronger guarantee of performance is required.

6 Conclusions and Societal Impacts

This work proposes EDNIL for training models invariant to distributional shifts. To infer environments without supervision, we propose a multi-head neural network structure to identify and diversify plausible environments. With joint optimization, the resulting invariant models are shown to be more robust than existing solutions on data having distinct characteristics and different strengths of biases. We attribute the effectiveness to the underlying learning objectives, which are consistent with recent studies of ideal environments. Additionally, we note that the classifier-like structure of environment inference model makes EDNIL easy to combine with off-the-shelf pre-trained models and trained more efficiently.

Our contributions to invariant learning have broader societal impacts on numerous domains. For instance, it can encourage further research and real-world applications on debiasing machine learning systems. To be more specific, the identification and elimination of potential biases can facilitate more robust model training. It can be beneficial to many real applications where distributional shifts commonly occur, such as autonomous driving, social media and healthcare.

Furthermore, as discussed by [6], invariant learning can promote algorithmic fairness in some ways. In particular, our empirical achievements on Adult-Confounded can prevent discrimination against sensitive demographic subgroups in decision-making process. It shows that EDNIL has a potential to learn a fair predictor without prior knowledge of sensitive attributes, which is related to [13, 12, 52]. We expect that one can extend our work to more fairness benchmarks and criteria in the future.

Last but not least, it is worth mentioning some cautions, however. Since the invariant learning algorithm claims to find invariant relationships, one might cast more attention on feature importance of the invariant model and even incorporate the results into further research or applications. Nevertheless, the results are reliable only when the model is trained appropriately. Insufficient data collection or careless training process, for example, can certainly affect the identification of invariant features, and thus mislead experimental findings. As a result, we believe that adequate and careful preparations and analyses are essential before drawing conclusions from the inferred invariant relationships.

Acknowledgments and Disclosure of Funding

We would like to thank the anonymous reviewers for their helpful suggestions. This material is based upon work supported by National Science and Technology Council, ROC under grant number 111-2221-E-002 -146 -MY3 and 110-2634-F-002-050 -. 
References


