Environment Diversification with Multi-head Neural Network for Invariant Learning

Anonymous Author(s)
Affiliation
Address
email

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 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 the environment or strong assumptions about the pre-train 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 empirically models trained with this framework are more robust against distributional shift.

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 be worse than random predictions [9]. In this work, we focus on *Invariant Learning*, which aims at learning causality 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 can be partitioned into multiple subsets 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 partitioned by 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 EIIL, a min-max optimization framework training the model of interest and inferring the environment labels. Another work, HRM [17] (or the extension, KerHRM [18]), parameterizes the environments and proposes clustering-based approaches to estimate the parameters. A recent method, ZIN [15], 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.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Delicate initialization: EIIL is able to infer environments but requires an ERM model for initialization. Crucially, the ERM model has to be underfitted and heavily depend on spurious correlations. As the distributional shift is assumed to be unknown in the training stage, appropriate initialization is difficult to guarantee.

Scalability issue: HRM and KerHRM, though do not possess the above two limitations, suffer from the scalability issue. Specifically, HRM is assumed to be trained by low dimensional data. KerHRM as the extension of HRM avoids the issue of dimensions by adopting kernel methods. However, the training cost of the kernel methods can be very high if the data or pre-trained model size is large.

This work proposes a novel framework, **E**nvironment **D**iversification with multi-head **N**eural network for **I**nvariant **L**earning (EDNIL). EDNIL is able to infer environment labels without supervision and be jointly optimized with an invariant learning model. The underlying neural network explicitly diversifies the inferred environments, which is consistent with a recent study [5] revealing the benefits of diverse environments. Notably, the structure of the proposed neural network is 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 [12] and DistilBert [24], and evaluate it with diverse data types and varied biases. The results show that EDNIL can constantly outperform the existing state-of-the-art models.
- The learning algorithm of EDNIL has theoretical connections to recent studies [5, 15, 17, 18] discussing conditions of ideal environments.
- EDNIL does not have the three limitations discussed above. The comparison between EDNIL and other models is shown in Table 1.

Table 1: A summary of the advantages of invariant learning models.

	No need of prior knowledge of environments	No need of delicate initialization	No scalability issue
IRM [1]	✗	✓	✓
ZIN [15]	✗	✓	✓
EIIL [6]	✓	✗	✓
HRM [17]	✓	✓	✗
KerHRM [18]	✓	✓	✗
Ours	✓	✓	✓

2 Preliminaries and related works

The goal of EDNIL is to tackle out-of-distribution problem 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 the assumptions of variant features and desired environments for invariant learning. In Section 2.3, we introduce the existing invariant learning methods that do not require human labelling.

2.1 Out-of-distribution Generalization and Invariant Learning

Following [1, 17, 18], we consider a dataset $D = \{D^e\}_{e\in\text{supp}(\mathcal{E}_{\text{tr}})}$ with different sources $D^e = \{(x_i^e, y_i^e)\}_{i=1}^{n_e}$ collected under multiple training environments $e \in \text{supp}(\mathcal{E}_{\text{tr}})$. Random variable $\mathcal{E}_{tr}$ represents the training environment labels. For simplicity, X^e, Y^e and P^e denote data, target and distribution in environment e respectively.

With $\mathcal{E}_{all}$ containing all possible indices of environments such that $\mathcal{E}_{all} \supset \mathcal{E}_{tr}$, the goal of out-of-distribution generalization is to learn a predictor $f(\cdot) : \mathcal{X} \to \mathcal{Y}$ as Equation 1. $R^e := E^e[l(f(X^e), Y^e)]$ measures the risk under environment e, where $l(\cdot,\cdot)$ is the loss function. In general, for $e \in \text{supp}(\mathcal{E}_{\text{tr}})$ and $e' \in \text{supp}(\mathcal{E}_{\text{all}}) \setminus \text{supp}(\mathcal{E}_{\text{tr}})$, $P^{e'}(X^{e'}, Y^{e'})$ is different from $P^e(X^e, Y^e)$.

$$f = \arg\min_f \max_{e\in\text{supp}(\mathcal{E}_{\text{all}})} R^e(f) \tag{1}$$

Recently, several studies [1, 20, 22, 14, 4] attempt to tackle the generalization problem by discovering invariant relationships across $\mathrm{supp}(\mathcal{E}_{\mathrm{all}})$. A commonly proposed invariance assumption is as follows:

Assumption 2.1 *There exists a random variable X_c such that $P^e(Y|X_c) = P^{e'}(Y|X_c)\ \ \forall e, e' \in \mathrm{supp}(\mathcal{E}_{\mathrm{all}})$.*

X_c, which is called *invariant features*, induces consistent conditional distributions over target Y across environments $e \in \mathrm{supp}(\mathcal{E}_{\mathrm{all}})$. To learn invariant predictor Φ that purely encodes the information of X_c, IRM [1] introduces a regularization term enforcing simultaneous optimality of the classifier $w \circ \Phi$ with $\mathcal{E}_{tr}$, where dummy classifier $w = 1.0$ is a fixed multiplier for each output dimension:

$$\sum_{e \in \mathrm{supp}(\mathcal{E}_{\mathrm{tr}})} R^e(\Phi) + \lambda ||\nabla_{w|w=1.0} R^e(w \circ \Phi)||^2 \tag{2}$$

2.2 Environment Inference

Following [17, 18, 15], the existence of variant features X_v is assumed as follows:

Assumption 2.2 *Given X and X_c satisfying Assumption 2.1, there exists a random variable X_v such that $X = h(X_c, X_v)$, where $h(\cdot)$ is a transformation function. The distribution $P^e(Y|X_v)$ can vary across environments $e \in \mathrm{supp}(\mathcal{E}_{\mathrm{all}})$.*

Given a mixture of training data D, $\mathcal{E}_{tr}$ is unavailable or sub-optimal for invariant learning in most applications [17], which limits the training of IRM. To facilitate invariant learning, our goal is to infer environments $\mathcal{E}_{learn}$ for IRM to elicit better invariant predictor. With Assumption 2.1 and 2.2, X_c and X_v are more identifiable when the dependency between X_c and Y is consistent across $\mathcal{E}_{learn}$, and that between X_v and Y is diversified [17, 18, 15]. Therefore, our task is defined as Problem 2.1.

Problem 2.1 *Given a dataset $D = \{D^e\}_{e \in \mathrm{supp}(\mathcal{E}_{\mathrm{tr}})}$, the task is to generate $\mathcal{E}_{learn}$ satisfying the following two conditions regarding conditional entropy H and learn an invariant predictor with Equation 2 for better out-of-distribution generalization:*

$$H(Y|X_c) = H(Y|X_c, \mathcal{E}_{learn}) \tag{3}$$

$$H(Y|X_v) - H(Y|X_v, \mathcal{E}_{learn}) > 0 \tag{4}$$

2.3 Invariant learning methods with unsupervised environment inference

Here we provide a more detailed introduction of the existing unsupervised invariant learning methods: EIIL [6], HRM [17] and KerHRM [18]. EIIL proposes formulating invariant learning as a min-max optimization problem. Specifically, EIIL is composed of two objectives, EI and IL, where EI is optimized by maximizing the training loss via labeling the data, and IL is optimized by minimizing the training loss given the data labeled by EI. The two-stage algorithm bypasses the difficulty of defining environments; however, the training result heavily relies on the initialization of the EI optimization. Empirically, the initialization demands an underfitted and biased ERM model, or EIIL can have a significantly weaker performance than ERM.

Another method, HRM, proposes a clustering-based method for learning plausible environments. HRM assumes that data 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 jointly learning framework which alternatively learns invariant predictors and improves quality of clustering results. A known issue of HRM is an assumption that the data are represented by raw features; namely, data such as images and text 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. [18] proposes KerHRM. The main idea is to adopt the Neural Tangent Kernel [13] 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 shortcoming of HRM and is shown to be more robust and effective. However, the kernel method also brings additional training cost depending on data size and pre-trained models. In particular, for applications favoring large pre-trained models such as Resnet [12] and BERT [7], KerHRM may not be an affordable option.

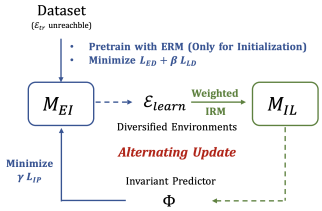

Figure 1: The jointly learning framework of EDNIL.

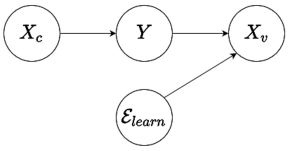

Figure 2: The underlying graphical model of EDNIL.

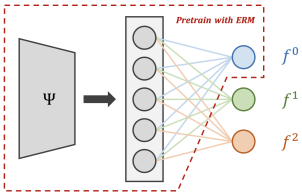

Figure 3: The multi-head neural network M_{EI}.

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} for environment inference and M_{IL} for invariant learning. Given the pooled data (X, Y), M_{EI} generates environments $\mathcal{E}_{learn}$ satisfying Condition 3 and 4, and M_{IL} is an invariant model trained with inferred environments. Our framework is jointly optimized with alternating updates. The learned invariant model can provide information of X_c to M_{EI}, so that Condition 3 and 4 can be fulfilled simultaneously.

3.1 The graphical model and learning objectives

We propose a graphical model which is a sufficient condition (the proof is in the supplementary file Section A) for Condition 3 and 4, and is the foundation of M_{EI} in EDNIL. The graph is shown in Figure 2. Based on the structure of the graphical model, we propose two objectives for learning to find effective environments $\mathcal{E}_{learn}$.

Label dependency objective: The target Y and inferred environments $\mathcal{E}_{learn}$ are independent.

Environment diversification objective: $H(Y \mid X_v) - H(Y \mid X_v, \mathcal{E}_{learn}) > 0$. (i.e. Condition 4)

Additionally, if jointly learning invariant representation $\Phi(X) \in X_c$ and environment $\mathcal{E}_{learn}$ is desired, we propose including and optimizing the third objective:

Invariance preserving objective: $H(Y \mid \Phi(X)) - H(Y \mid \Phi(X), \mathcal{E}_{learn}) = 0$. (i.e. Condition 3)

3.2 The environment inference network

The inference model, M_{EI}, is a deep neural network learning to optimize the objectives proposed in Section 3.1. Following the idea of HRM [17] and KerHRM [18], we assume the mapping relation between X and Y in environment e can be modelled by a function $f^e(\Psi(X))$, where $\Psi(X)$ is learned representations expected to encode variant features X_v. Instead of taking $f^e(\Psi)$ as a center of a cluster, we propose combining the environmental functions to build a multi-head network as shown in Figure 3. Specifically, a cluster in HRM [17] or KerHRM [18] corresponds to a single-head neural network in M_{EI} with shared parameters. The training procedure of M_{EI} can be divided into *inference stage* and *learning stage*.

Inference stage The goal is to infer the environment of each training data. Equation 5 is the proposed method, where the probability $P(e \mid X, Y)$ is estimated via a softmax of negative $l(f^e(\Psi(X), Y))$ with a hyper-parameter τ. The function l is expected to be the commonly used cross entropy or mean squared error loss, measuring the discrepancy between $f^e(\Psi(X))$ and Y.

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

Learning stage The goal is to update the neural network to improve the quality of inference. Three losses are minimized for our three objectives. The strength of each loss can be controlled by hyper-parameters β and γ.

$$L_{EI} = L_{ED} + \beta L_{LD} + \gamma L_{IP} \tag{6}$$

Given the estimated $P(\mathcal{E}_{learn}|X,Y)$, L_{ED} selects the most probable environment and its corresponding network for optimization:

$$L_{ED} = -\sum_i w_i \log \max_{e \in \text{supp}(\mathcal{E}_{\text{learn}})} P(e \mid x_i, y_i) \tag{7}$$

Although $\max_{e \in \text{supp}(\mathcal{E}_{\text{learn}})} P(e \mid X, Y)$ selects only one environment for the minimization, the softmax simultaneously propagates gradient to maximize $l(f^{e'}(\Psi(X), Y))$. As a result, the network learns to predict Y by relying on both $\Psi(X)$ and $\mathcal{E}_{learn}$, consistent with the environment diversification objective that $H(Y \mid X_v) - H(Y \mid X_v, \mathcal{E}_{learn}) > 0$. In practice, we utilize rescaling weight w_i inversely proportional to the size of e_i. The importance of smaller environments will be thus enhanced within summation.

The second term, L_{LD}, constraints the mutual information $I(Y; \mathcal{E}_{learn})$ measuring the dependency between Y and $\mathcal{E}_{learn}$. With satisfying the label dependency objective, L_{LD} also prevents a naive solution that environments are purely split by labels without considering X_v, which is undesirable for invariant learning. To minimize $I(Y; \mathcal{E}_{learn})$, it can be verified that it is equivalent to minimizing Equation 8 given $P(Y)$ is known.

$$L_{LD} = \mathbb{E}_{e \sim P(\mathcal{E}_{learn})}[\sum_y P(y|e) \log P(y|e)] \tag{8}$$

The last term, L_{IP}, promotes $\mathcal{E}_{learn}$ to invariance preserving objective as M_{IL} jointly learns invariant features to some extent. To prevent existing invariant relationship from being diversified by L_{ED}, it restricts the variance of expected loss from invariant model Φ between environments.

$$L_{IP} = Var_{e \in \text{supp}(\mathcal{E}_{\text{learn}})}[\mathbb{E}(l(\Phi(X^e), Y^e))] \tag{9}$$

In addition, before minimizing L_{EI}, we pre-train our Ψ and one arbitrary f^e with ERM. In general, it empirically facilitates better feature extraction. Unlike [6] taking ERM as reference model heavily relying on spurious features, EDNIL performs consistently under various choices of ERM. In particular, it is not restricted to under-fitted ERM training. We verify the argument in Section 4.

3.3 The invariant learning model

Invariant Learning identifies invariance across given environments. For our learning model, M_{IL}, IRM [1] is selected as our base algorithm. We assign environment label $e \in \text{supp}(\mathcal{E}_{\text{learn}})$ with largest $P(e|x_i, y_i)$ to each data (x_i, y_i) for training. 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 $e \in \text{supp}(\mathcal{E}_{\text{learn}})$, i.e. $P(e|X^e, Y^e)$. Our training objective is modified to minimize the weighted average of the environment risks:

$$L_{IL} = \sum_{e \in \text{supp}(\mathcal{E}_{\text{learn}})} c^e \cdot [R^e(\Phi) + \lambda ||\nabla_{w|w=1.0} R^e(w \circ \Phi)||^2] \tag{10}$$

$$c^e = \sum_i P(e|x_i^e, y_i^e) \tag{11}$$

4 Experiments

We empirically validate the proposed methods on four biased datasets, Adult-Confounded, CMNIST, Waterbirds and SNLI. The generation of spurious correlations mainly follow the protocols proposed by [6, 18, 23, 8]. 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 representation learning. Modern deep models with pretrained weights will serve as the encoders of variant and invariant features.

For hyper-parameter tuning, we split 10% of training data as in-distribution validation set. In each dataset, several testing subsets with different distributions are listed to evaluate the robustness of each method, and we mainly take worst-case performance for assessment. † will be marked if the difference is statistically significant (p-value less than 0.05) comparing with the 2nd-best result.

- **Empirical Risk Minimization (ERM):** ERM chooses model parameters that minimize the average loss on the training set.
- **Invariant Risk Minimization (IRM [1]):** Given environments, IRM seeks the invariance relationship with Equation 2. As indicated in [5, 18, 8], larger discrepancy of spurious correlations between environments benefits invariant learning more. Therefore, unlike [6, 17, 18] who take sub-optimal $\mathcal{E}_{tr}$ for IRM, we re-label $\mathcal{E}_{oracle}$ on given biased training set, which extremely diversifies the correlations between variant features and target, to provide upper-bound performance with IRM.
- **Environment Inference for Invariant Learning (EIIL [6]):** EIIL partitions environments by maximizing Equation 2 with ERM as reference model. Subsequently, IRM is taken to learn invariant predictor with inferred environments.
- **Kernelized Heterogeneous Risk Minimization (KerHRM [18]):** KerHRM jointly explores heterogeneity by clustering algorithm and learns invariant parameters in kernel space.

4.1 Simple datasets with MLP

This section includes two simple datasets, Adult-Confounded and CMNIST, where spurious correlations are synthetically produced with the pre-defined strength. For all competitors, MLP is taken as the base model and full-batch training is implemented. Since KerHRM performs unstable 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 a star (*) in each table.

4.1.1 Discussion on Adult-Confounded dataset

We take UCI Adult dataset[1] to predict binarized income levels (above or below \$50,000 per year) based on demographic information. 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 size from UCI Adult, we sample data for the given label distributions in each sensitive subgroup as shown in Table 2. In this task, MLP with one hidden layer of 96 neurons is considered. For IRM, four environments comprise $\mathcal{E}_{oracle}$, where the correlations between variant features $(race, sex)$ and target Y are distributed without overlapping. More details are provided in the supplementary file Section B.

Results The results are shown in Table 3. First of all, not surprisingly we found ERM adopts poorly to distribution shift. Among all invariant learning methods without prior environment labels, EDNIL can perfectly identify variant features and generate diversified environments. Therefore, it achieves the most invariant performance over different testing distributions. EIIL can improve consistency to some degree, but not as strong as EDNIL. A possible explanation is that some predictable invariant features risk being diversified across environments and being discarded by invariant learning since empirically trained reference model is not guaranteed to be purely variant [6]. For KerHRM, it performs inconsistently across random seeds, which can be reflected on large standard deviation. In some cases, the performance hardly improves over iterations, as also observed in Liu et al. [18].

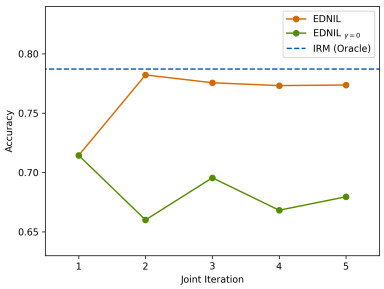

Figure 4: Ablation study of joint optimization. $\gamma = 0$ means the removal of L_{IP}.

Ablation Study for M_{EI} We first claim the importance of constraining label dependency by removing the coefficient β of L_{LD}. As discussed in Section 3, the resulting environments are purely split by labels, which leads to inferior performance for invariant learning as shown in Table 3. Figure 4 shows the effectiveness of joint optimization. The regularization L_{IP} promotes environment inference, so that the worst-case performance improves and remains stable over iterations. If the coefficient γ is turned off, feedback generated by M_{IL} will be ignored and the effect of invariant

[1] https://archive.ics.uci.edu/ml/datasets/adult

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

	Train	Test (IID)	Test (IND)	Test (OOD)
SG1	0.9	0.9	0.5	0.1
SG2	0.1	0.1	0.5	0.9
SG3	0.9	0.9	0.5	0.1
SG4	0.1	0.1	0.5	0.9

Table 3: Testing accuracy on Adult Confounded. Three subsets are defined in Table 2. $\beta = 0$ and $\gamma = 0$ mean the removal of L_{LD} and L_{IP} when training EDNIL.

	IID	IND	**OOD**
ERM	**92.4** $\pm$ **0.1**	66.8 $\pm$ 0.3	40.7 $\pm$ 0.5
EIIL	75.8 $\pm$ 0.4	73.3 $\pm$ 0.5	70.4 $\pm$ 1.7
KerHRM	82.4 $\pm$ 3.9	75.1 $\pm$ 4.0	67.9 $\pm$ 9.3
KerHRM*	81.2 $\pm$ 1.8	78.5 $\pm$ 0.3	75.6 $\pm$ 1.9
EDNIL	80.7 $\pm$ 0.4	**79.1** $\pm$ **0.4**	**77.5** $\pm$ **0.3**†
EDNIL$_{\beta=0}$	91.8 $\pm$ 0.0	66.7 $\pm$ 0.1	41.3 $\pm$ 0.7
EDNIL$_{\gamma=0}$	78.2 $\pm$ 2.4	75.4 $\pm$ 1.6	72.5 $\pm$ 3.3
IRM	79.9 $\pm$ 0.4	79.3 $\pm$ 0.3	78.8 $\pm$ 0.4

learning risks being undesirable. In Table 3, the degradation occurs on all subsets. It indicates that some invariant features are eliminated since Condition 3 cannot be realized in the splitting of environments.

4.1.2 Discussion on CMNIST dataset

We report our evaluation on a noisy digit recognition dataset, CMNIST. Following [1], we first assign $Y = 0$ to those whose digits are larger than 5 and $Y = 1$ 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, e is set to 0.15, which is equivalent to merging two training environments from [1]. 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 training set. On the other hand, $e = 0.9$ defines the most challenging scenario since the 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. Due to massive computing resources required by KerHRM, we follow the settings recommended by [18]. Specifically, we randomly select 5,000 samples and train MLP with one layer of 1024 neurons. To construct most diversified $\mathcal{E}_{oracle}$ for IRM, we pack all examples with $C = Y$ into one environments, and $C \neq Y$ into the other.

Table 4: Testing accuracy of CMNIST, where color noise 0.9 is the worst-case subset.

Color	0.1	0.5	**0.9**
ERM	**88.4** $\pm$ **0.3**	55.0 $\pm$ 0.5	21.7 $\pm$ 0.8
EIIL	76.5 $\pm$ 0.6	73.9 $\pm$ 0.4	71.0 $\pm$ 0.5
KerHRM	74.3 $\pm$ 0.7	66.2 $\pm$ 1.7	58.0 $\pm$ 11.5
KerHRM*	71.3 $\pm$ 0.7	68.5 $\pm$ 0.5	66.1 $\pm$ 0.7
EDNIL	77.7 $\pm$ 0.4	**76.8** $\pm$ **0.3**	**75.2** $\pm$ **0.4**†
IRM	77.8 $\pm$ 0.4	76.8 $\pm$ 0.4	75.2 $\pm$ 0.3

Results The results are shown in Table 4. With large spurious correlation at train time, ERM obtains high accuracy as the correlation remains aligned; however, its generalization to other testing distribution is limited. Among all invariant learning methods without manual environment labels, EDNIL gets closest to IRM with $\mathcal{E}_{oracle}$, achieving consistent and robust performance in this dataset.

Number of Environments As shown in Figure 5, EDNIL is not sensitive to predefined number of environments. Specifically, when the environment number is larger than the oracle (i.e. 2), some environment classifiers become redundant. Each of them owns a high constant loss, taking up a fixed and ignorable space in softmax function. The comparison between $\mathcal{E}_{oracle}$ and $\mathcal{E}_{learn}$ with size of 5 is visualized in Figure 6. Additionally, training EDNIL with more environments is much more efficient than clustering-based methods, such as the one proposed in HRM and KerHRM.

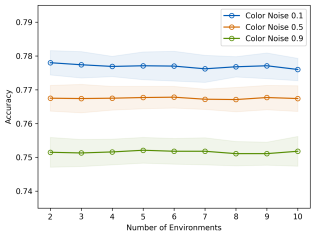

Figure 5: EDNIL with different number of environments.

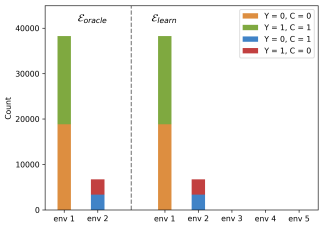

Figure 6: Comparison between $\mathcal{E}_{learn}$ and $\mathcal{E}_{oracle}$.

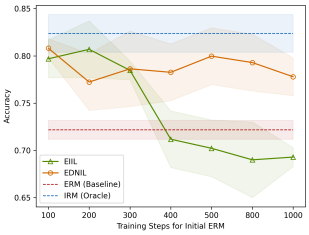

Figure 7: Testing reliance on initialization.

4.2 Complex datasets with pre-trained deep learning models

This section extends MLP to deep learning models with pre-trained weights on more complex data. With mini-batch fine-tuning, we consider all competitors except KerHRM due to its limitation to deep representation learning [18]. In Section 4.2.1, image dataset, Waterbirds [23], 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. [8].

4.2.1 Discussion on Waterbirds dataset

In Waterbirds [23], each bird photograph, from CUB dataset [25], is combined with one background image, from Places dataset [26]. Both birds and backgrounds are either from land or water, and our target is to predict the species of birds. At training time, landbirds and waterbirds frequently present in land and water backgrounds respectively. Therefore, empirically trained models are prone to learn context features, and fails to generalize as background varies [2, 6, 9, 16, 23].

To split validation set so that its distribution is i.i.d. to 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 scenario. In this task, *Resnet-34* [12] is chosen with 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, 23], ERM suffers in the hardest scenario (i.e. waterbirds on land). EIIL also performs poorly in this scenario. With more sophisticated learning framework, EDNIL narrows the gaps between subgroups and uplifts the worst-case performance. The results show that EDNIL is more resistant to distribution shifting.

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 7. As suggested by [6], EIIL works only with underfitted reference model (training steps fewer than 400). If the reference model is well-fitted, the performance of EIIL will greatly decline since ERM learns beyond variant features. One can be misled into undesirable choice for EIIL when seeking hyper-parameters with in-distribution validation set. For instance, the validation score of EIIL with 500-step reference model (96.9%) is much higher than that with 100-step (94.9%), which is not consistent with their performances on testing. In comparison, EDNIL remains consistent across different pre-trained steps, which accentuates our strength of *effortless initialization*.

4.2.2 Discussion on SNLI dataset

The target of SNLI [3] is to predict the relation between two given sentences, premise and hypothesis. Recent studies [11, 19, 21] reveal hypothesis bias in SNLI, which is characterized by patterns in hypothesis highly correlated with a specific label. One can achieve low empirical risk without

Table 5: Testing accuracy of Waterbirds where Y and BG means target and background respectively. The subgroup (Water, Land) contributes the worst-case performance.

(Y, BG)	(Land, Land)	(Water, Water)	(Land, Water)	**(Water, Land)**
ERM	$\mathbf{99.4 \pm 0.0}$	$\mathbf{91.3 \pm 0.2}$	$\mathbf{91.0 \pm 0.7}$	72.2 ± 0.6
EIIL	99.3 ± 0.3	91.3 ± 1.3	88.6 ± 2.8	70.2 ± 3.4
EDNIL	98.3 ± 0.7	91.1 ± 0.8	89.2 ± 2.1	$\mathbf{80.0 \pm 3.3}^{\dagger}$
IRM	98.3 ± 0.4	90.4 ± 1.1	89.8 ± 1.9	82.4 ± 2.2

considering premises during prediction. However, as the bias no longer holds, the performance degradation occurs [10, 19].

We sample 50,000 examples and select two major classes, *entailment* and *contradiction*, for our experiment. Following [8], we define three subsets, unbiased, bias aligned and bias misaligned, by training a biased model with hypothesis as its only input. Specification of the subsets is as follows:

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

The proportions of the three subsets are 22%, 67% and 11% respectively. Due to the minority of misaligned bias subset, it is more likely to be ignored and thus defines the worst-case performance.

For all methods, *DistilBERT* [24] is taken as the pre-trained model for further mini-batch fine-tuning. For $\mathcal{E}_{oracle}$, we assign the aligned bias subset to the first environment, and the misaligned bias subset to the second. In order to make bias prevalence equal in the two environments, samples in the unbiased subset are scattered according to their sizes.

Table 6: Testing accuracy on SNLI, where bias misaligned subset defines the worst-case performance.

Subset	Unbiased	Aligned	**Misaligned**
ERM	$\mathbf{90.3 \pm 0.4}$	$\mathbf{97.7 \pm 0.2}$	73.4 ± 0.8
EIIL	88.6 ± 0.7	97.2 ± 0.2	72.1 ± 0.3
EDNIL	89.9 ± 0.4	97.4 ± 0.1	$\mathbf{74.6 \pm 0.5}^{\dagger}$
IRM	89.9 ± 0.3	96.6 ± 0.1	76.1 ± 0.8

Results Table 6 shows our results with the three subsets. As reported by [8], ERM receives remarkable score on the bias aligned subset, but it fails on the bias misaligned case. Although both EIIL and EDNIL degrade on the unbiased and bias aligned subsets, only EDNIL improves significantly on the bias misaligned subset. Even though the definition of biases are in high level, EDNIL is still capable of encoding and diversifying possible variant features.

5 Limitation

The learning algorithm of EDNIL 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 model effectiveness on diverse datasets, while we are aware that a stronger guarantee of performance is required.

6 Conclusion

This work proposes EDNIL for training models invariant to distributional shifts. To infer environments without supervision, we propose a multi-head neural network structure with a two-stage learning procedure to identify and diversify plausible environments. With joint optimization, the trained 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 consistent with recent studies of ideal environments. Additionally, we note that the classifier-like structure makes EDNIL easy to combine with off-the-shelf pre-trained models and trained efficiently.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv*, 2019. URL https://arxiv.org/abs/1907.02893.

[2] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *ECCV*, 2018.

[3] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, sep 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

[4] Shiyu Chang, Yang Zhang, Mo Yu, and T. Jaakkola. Invariant rationalization. In *ICML*, 2020.

[5] Yo Joong Choe, Jiyeon Ham, and Kyubyong Park. An empirical study of invariant risk minimization. *ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning*, 2020.

[6] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning (ICML)*, 2021.

[7] Devlin, Jacob, Chang, Ming-Wei, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[8] Yana Dranker, He He, and Yonatan Belinkov. IRM—when it works and when it doesn't: A test case of natural language inference. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=KtvHbjCF4v.

[9] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. URL https://doi.org/10.1038%2Fs42256-020-00257-z.

[10] Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2103. URL https://aclanthology.org/P18-2103.

[11] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *NAACL*, 2018.

[12] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

[14] Kun Kuang, Ruoxuan Xiong, Peng Cui, Susan Athey, and Bo Li. Stable prediction with model misspecification and agnostic distribution shift. In *AAAI*, 2020.

[15] Yong Lin, Shengyu Zhu, and Peng Cui. Zin: When and how to learn invariance by environment inference? *arXiv*, 2022. URL https://arxiv.org/abs/2203.05818.

[16] Evan Zheran Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning (ICML)*, 2021.

[17] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Heterogeneous risk minimization. In *International Conference on Machine Learning (ICML)*, 2021.

[18] Jiasuho Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Kernelized heterogeneous risk minimization. In *NeurIPS*, 2021.

[19] Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL `https://aclanthology.org/P19-1334`.

[20] J. Peters, Peter Buhlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78, 2015.

[21] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana, jun 2018. Association for Computational Linguistics. doi: 10.18653/v1/S18-2023. URL `https://aclanthology.org/S18-2023`.

[22] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard E. Turner, and J. Peters. Invariant models for causal transfer learning. *J. Mach. Learn. Res.*, 19:36:1–36:34, 2018.

[23] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *In International Conference on Learning Representations (ICLR)*, 2020.

[24] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[25] P Welinder, S Branson, T Mita, C Wah, F Schroff, S Belongie, and P Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, Caltech, 2010.

[26] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. doi: 10.1109/TPAMI.2017.2723009.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] in supplementary file Section B
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]

(c) Did you include any new assets either in the supplemental material or as a URL? [No]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]