Bayesian Invariant Risk Minimization

Yong Lin[1*] Hanze Dong[1*] Hao Wang[2] Tong Zhang[1†]
[1]The Hong Kong University of Science and Technology [2]Rutgers University
{ylindf,hdongaj}@ust.hk hw488@cs.rutgers.edu

Abstract

Generalization under distributional shift is an open challenge for machine learning. Invariant Risk Minimization (IRM) is a promising framework to tackle this issue by extracting invariant features. However, despite the potential and popularity of IRM, recent works have reported negative results of it on deep models. We argue that the failure can be primarily attributed to deep models' tendency to overfit the data. Specifically, our theoretical analysis shows that IRM degenerates to empirical risk minimization (ERM) when overfitting occurs. Our empirical evidence also provides supports: IRM methods that work well in typical settings significantly deteriorate even if we slightly enlarge the model size or lessen the training data. To alleviate this issue, we propose Bayesian Invariant Risk Minimization (BIRM) by introducing Bayesian inference into the IRM. The key motivation is to estimate the penalty of IRM based on the posterior distribution of classifiers (as opposed to a single classifier), which is much less prone to overfitting. Extensive experimental results on four datasets demonstrate that BIRM consistently outperforms the existing IRM baselines significantly.

1. Introduction

The past decade has witnessed a great success of machine learning technology, boosting the development in computer vision [25,31], speech recognition [23] and many other areas [6,19,36,57,58]. However, more in-depth studies have recently revealed the failure of these models due to the existence of spurious features or shortcuts [7,14,20,55]; [7] raised an example: models could rely on the background (pastures or deserts) to distinguish cows and camels. In this case, background, a spurious feature, is non-invariant and can change arbitrarily in different domains.

The common foundation of machine learning based on the i.i.d (independent and identically distributed) assumption does not always hold. The Empirical Risk Minimization (ERM) based models can deteriorate dramatically if the testing distribution is different from the training one. This is also known as the out-of-distribution (OOD) generalization problem. To relax the i.i.d. assumption, [40] propose to exploit the "invariance principle" to obtain a better OOD generalization ability. The invariance principle aims to utilize invariant features that are stable even in the case of distributional shifts. In the aforementioned cow and camel example [7], the shape of the animal is an invariant feature.

Invariant Risk Minimization (IRM) [4] extends the invariance principle to neural networks. Specifically, IRM considers that training data is collected from multiple environments (domains) and the correlation of spurious features with the labels differs in different environments while the correlation of invariant ones remain stable. IRM regularizes neural networks to extract invariant features and discard spurious features. Hopefully, the model relying on invariant features only can generalize to unseen environments well.

IRM has gained its popularity for its potential and inspires a line of excellent works [1–3,11,32,41,51,56]. IRM is guaranteed to identify the invariant features given enough environments with linear model [4, 41]. However, recent empirical findings in [24,34] indicate the ineffectiveness of IRM methods on deep models. We argue that this failure can be mainly attributed to deep models' tendency to overfit data. From a theoretical perspective, we show that IRM can ultimately degenerate to ERM when overfitting occurs. The theoretical findings are verified by extensive empirical evidence: (1) the model trained by ERM can also minimize IRM penalty (Section 3.3); (2) IRM-trained model can still contain spurious features while the IRM penalty vanishes (Section 5); (3) the IRM methods deteriorate quickly with an enlarged model or lessened data (Section 5).

Motivated by both the theoretical and empirical findings, we propose Bayesian Invariant Risk Minimization (BIRM) as a Bayesian treatment [8] of IRM to substantially alleviate overfitting, making IRM practical in deep models. Suppose the prediction model consists of a feature extractor and a classifier [16, 17]. Given the learned feature representation, BIRM estimates the posterior distribution of the classifier (as opposed to a single classifier) for each environment. If

[*] Equal contribution. This work is supported by GRF 16201320.
[†] Jointly with Google Research

feature representation only contains invariant features, the estimated posterior in each environment should be almost the same. Otherwise, the posterior distribution will differ across environments, and to prevent this from happening, we then introduce an additional penalty term. Compared with existing IRM methods, BIRM estimates the invariance regularization on the posterior distribution of the classifier, which is less prone to overfitting [8].

Contributions.

- We formally identify overfitting as a crucial reason why IRM fails in large deep models. We provide empirical evidence with supporting theoretical analysis.
- We propose a Bayesian formulation of IRM to alleviate overfitting, along with an efficient algorithm, reducing the chance of failure for deep IRM models significantly.
- We verify the effectiveness of Bayesian IRM with extensive experiments and show that our method improves the existing baselines by a large margin.

2. Related Works

Invariant Risk Minimization. IRM is developed in [4] and became popular recently. Several IRM variants are proposed subsequently: [32, 51] suggests to penalize the variance of the risks among different environments; [11, 56] uses neural networks to estimate the violation of invariance; [52] extends this idea further by optimizing the worst case in a convex hull of classifiers; [2] presents IRM games by incorporating game theory. [1, 13, 35] consider a more challenging task where the explicit environmental index is not available. The theoretical properties of IRM are analyzed in [3, 12, 27, 41]. [3] studies the sample efficiency of IRM. [41] investigates IRM with a special non-linear function. [12] utilizes iterative methods to reduce the number of environments required by IRM. Despite the popularity of IRM, some recent works [24, 34] find IRM less effective on deep models. In this paper, we attribute this issue to the overfitting problem and upgrade IRM by incorporating the Bayesian principle.

Bayesian Inference. Bayesian inference is an essential method of statistical inference; it considers the uncertainty of model parameters [9, 21, 47, 48]. Bayesian inference has been widely adopted in many machine learning topics, e.g., uncertainty qualification [22, 39, 46, 48], reinforcement learning [45], etc. Nonetheless, the approximation of posterior distributions in Bayesian methods can be challenging. Fortunately, variational inference offers possibility to estimate these posterior distributions efficiently even on large models [10, 15, 29, 48, 53]. Recently, Bayesian inference has also been introduced to deep learning models to improve robustness [15, 26, 29, 30, 37, 38, 48, 49, 53], which also inspire our method.

3. IRM and Its Overfitting Pitfall

3.1. Invariant Risk Minimization

Preliminaries. Throughout the paper, upper-cased letters, X and Y, denote random variables; lower-cased letters, $\mathbf{x}$, y, and w, denote samples and parameters. We assume there is a set of multiple environments, $\mathcal{E}$, where the data can be drawn from. During training, we have access to a collection of environments, $\mathcal{E}_{tr} \subset \mathcal{E}$; each environment $e \in \mathcal{E}_{tr}$ contains n^e samples, denoted as $\mathcal{D}^e \triangleq \{(\mathbf{x}_i^e, y_i^e)\}_{i=1}^{n_e}$. Let $\mathcal{X}$ and $\mathcal{Y}$ be the space of X and Y. Our goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$, which predicts Y given X. Here f consists of a classifier $g_w(\cdot)$ and a feature extractor $h_u(\cdot)$ with parameters w and u, respectively. The task of out-of-distribution generalization aims to find the optimal w and u which minimizes the loss of the worst environment:

$$\min_{w,u} \sup_{e\in\mathcal{E}} \mathcal{R}^e(w,u), \tag{1}$$

where $\mathcal{R}^e(w,u)$ is the negative log likelihood of the data from e. Formally, we have

$$\mathcal{R}^e(w,u) = -\ln p(\mathcal{D}^e|w,u) = -\sum_{i=1}^{n_e} \ln p\big(y_i^e|w, h_u(\mathbf{x}_i^e)\big),$$

that is, we aim to learn the optimal w and u to maximize the likelihood of the *worst* environment from $\mathcal{E}$. We only consider the case that w, u are well-specified, such that $\mathcal{R}^e(w,u) \geq 0$ holds for all w, u.

Invariant Risk Minimization (IRM). IRM [4] aims to solve the following objective to achieve (1):

$$\begin{aligned} &\min_{w,u} \sum_{e\in\mathcal{E}_{tr}} \mathcal{R}^e(w,u), \\ &\text{s.t.} \quad w \in \arg\min_{w^e} \mathcal{R}^e(w^e,u), \forall e \in \mathcal{E}_{tr} \end{aligned} \tag{2}$$

IRM defined in Eq. (2) tries to learn a feature representation by $h_u(\cdot)$ that can induce a classifier $g_w(\cdot)$ which is simultaneously optimal for all training environments. To achieve this, $h_u(\cdot)$ should discard spurious features.

IRMv1. Since Eq. (2) is a challenging bi-level optimization problem, [4] proposes IRMv1 to approximate the solution of Eq. (2). IRMv1 is shown as following:

$$\min_{w,u} \sum_{e\in\mathcal{E}_{tr}} \mathcal{R}^e(w,u) + \lambda\|\nabla_w \mathcal{R}^e(w,u)\|^2 \tag{3}$$

Besides IRMv1, several other excellent variants of IRM emerged recently: InvRat [11] estimates the penalty by a mini-max procedure; REx [32] uses the variance of the losses in different environments as the penalty. Due to space constraints, we refer readers to the original works [2, 4, 11, 13, 51] for detailed description.

3.2. The Overfitting Pitfall

In this section, we theoretically analyze the behavior of IRM when overfitting occurs. Our results show that the invariant constraint of IRM in Eq. (2) holds trivially when the model memorizes training data. Then IRM will no longer provide any guarantee on learning invariant features. Our analysis works under the following assumptions:

Assumption 1 (Finite Sample Size)**.** *The number of training environments and samples are finite:* $|\mathcal{E}_{tr}| < \infty$ *and* $|\mathcal{D}_e| = n^e < \infty, \forall e \in \mathcal{E}_{tr}$*.*

Assumption 2 (Sufficient Capacity)**.** *The parameters* w *and* u *have sufficient capacity to fit the training data: there exist* $\bar{w}$ *and* $\bar{u}$*, such that* $\forall e \in \mathcal{E}_{tr}, \mathcal{R}^e(\bar{w}, \bar{u}) = 0$*.*

Assumption 1 holds in practice because we have access to only limited training data from several environments. Assumption 2 is also consistent with the recent findings on over-parameterized neural networks; for example, [54] shows that large neural networks can memorize all training data even in the presence of strong regularization.

We then proceed by defining the overfitting region.

Definition 1 (Overfitting Region)**.** *The overfitting region,* Ω*, is the collections of* $\bar{w}$ *and* $\bar{u}$ *that satisfies Assumption 2:*

$$\Omega := \{\bar{w}, \bar{u} | \mathcal{R}^e(\bar{w}, \bar{u}) = 0, \forall e \in \mathcal{E}_{tr}\}$$

Our main results go as following:

Proposition 1 (Failure of General IRM)**.** *Under Assumption 1 and 2, IRM degenerates to ERM in* Ω*. Furthermore, any element in* Ω *is a solution of IRM defined in Eq. (2).*

The full proof of Proposition 1 is deferred to Appendix A. Proposition 1 shows any model that overfits the training data is a solution of IRM in Eq. (2), no matter whether the model uses spurious feature or not. Such model may behave arbitrarily badly in an unseen test environment. Unfortunately, such an overfitting phenomenon is common for deep neural networks [54].

Connection to Existing Theory. Some theoretical properties of IRM are analyzed in [3,41]. [3] shows that the sample complexity of IRM is worse than ERM. [41] shows the difficulty of IRM for non-linear functions. Compared to [3, 41], our theory enjoys the following favorable properties.

- Our theory directly works on the definition of IRM, which is applicable to various variants of IRM [4, 11, 32, 51, 52, 56]. In contrast, [3, 41] focus on merely one variant of IRM, IRMv1 [4]. Whether their theories are applicable to other variants remains under-explored.
- [41] restricts the discussion within some special non-linear models where the function value jumps on the boundary of the high-density region. It is hard to verify whether this case is general enough to cover models used in practice, i.e. neural networks. In contrast, our theory works on very mild and verifiable assumptions.

Corollary 1 below implies that it is also difficult for IRMv1 to learn invariant features in the overfitting case.

Assumption 3 (Differentiability)**.** $\mathcal{R}^e(w, u)$ *is differentiable w.r.t.* w, u*.*

Corollary 1 (Failure of IRMv1)**.** *Under assumptions 1,2,3,* $\forall(\bar{w}, \bar{u}) \in \Omega$*, the following equality holds:*

$$(\bar{w}, \bar{u}) \in \arg\min_{w,u} \sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(w, u) + \lambda \|\nabla_w \mathcal{R}^e(w, u)\|^2$$

Corollary 1 indicates that any model with zero empirical loss is also an optimal solution of IRMv1. Notably, this model can still rely on spurious features. The proof of Corollary 1 is a direct consequence of Proposition 1. We can also prove similar failure cases of InvRat, REx. Due to space limitations, we leave them in the Appendix A.

3.3. Empirical Evidence

As we indicated above, IRM will fail if the model memorizes the data. To see this, we visualize the training procedure of ERM. The penalty of IRM is computed but not applied to the training objective. At the same time, we also estimate spurious features are contained in the model by the *non-invariant indicator*. Non-invariant indicator is defined as the percentage of testing samples whose predictions are vulnerable to the change of spurious features (refer to Appendix B for detailed explanation). Zero non-invariant indicator means the model completely ignores the spurious feature while larger one stands for more spurious feature usage. Figure 1 shows the IRM penalty and the non-invariant indicator when we train a ERM model with 3-layer-MLP on CMNIST [4]. At the beginning, the randomly initialized network does not contain spurious features so the non-invariant indicator and IRM penalty are at low level. As the training proceeds, the model learns the spurious features quickly, increasing of the non-invariant indicator and IRM penalty. Then as the model learns the invariant feature after spurious feature [44], the non-invariant indicator drops and plateaus until the end. The IRM penalty vanishes at the end as the model begins to memorize the data, however the non-invariant indicator remains 60% - 70%. In other

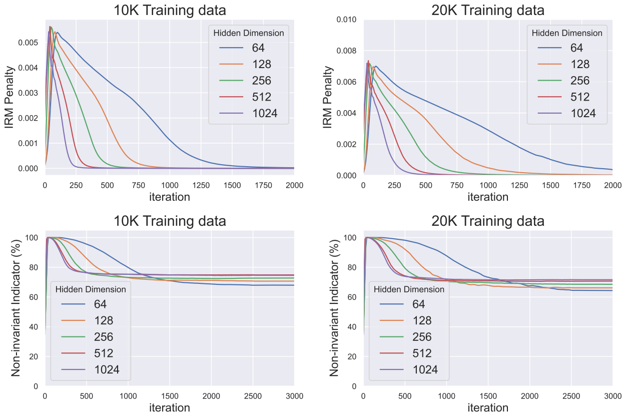


Figure 1. Illustration of training ERM on CMNIST [4] with 3-layer MLP of different hidden dimensions. The penalty of IRM (REx) is measured but not applied to the objective. As the training of ERM proceeds, the IRM penalty decays to zero while the non-invariant indicator shows the existence of large amount of spurious feature in the model. The IRM penalty vanishes faster with larger model and less training data.

words, the model still heavily relies on spurious features while IRM penalty can not detect it. Figure 1 further shows that the IRM penalty vanishes even faster as the capacity of model increases or the dataset size decreases. The empirical phenomenon is consistent with our theoretical results in Section 3.2: IRM fails when overfitting. More empirical supports are available in Section 5.

4. Bayesian Invariant Risk Minimization

In Section 3.2, we have shown that overfitting is harmful for IRM. Bayesian inference is a well known method to alleviate overfitting and it is proven to achieve the optimal sample complexity rate in the presence of model misspecification [5,33]. In this section, we propose Bayesian Invariant Risk Minimization (BIRM), a novel variant of IRM, by incorporating Bayesian principle. Extensive experimental results in Section 5 show superiority of BIRM.

4.1. Motivation and Formulation

To motivates our method, we set up a diagram in Figure 2 for the invariant learning problem. The node u stands for the feature extractor $h_u(\cdot)$. Let $\mathcal{D}_u^e$ be the data from environment e that is transformed by the extractor $h_u(\cdot)$: $\mathcal{D}_u^e \triangleq \{h_u(\mathbf{x}_i^e), y^e\}_{i=1}^n$. Let $\mathcal{D}_u \triangleq \bigcup_{e=1}^{\mathcal{E}_{tr}} \mathcal{D}_u^e$ denote the data collection from the mixture of the training environments. The nodes w^e and w in Figure 2 stand for $p(w^e|\mathcal{D}_u^e)$ and $p(w|\mathcal{D}_u)$, the posteriors of the classifiers given the feature representation, respectively. We add zebra stripes in Figure 2 to distinguish w and u from w^e because w and u are not dependent on a certain environment index. Following the common practice in typical mean-field variational inference [10], we assume the same prior $p_0(w)$ for all w^e and w.

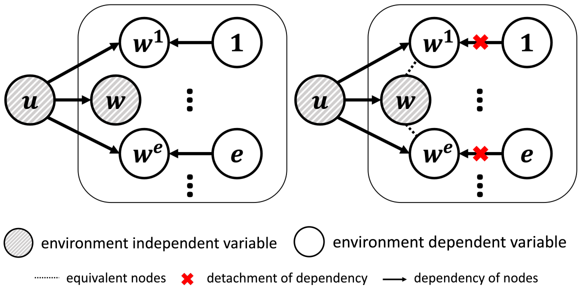


Figure 2. Diagram of models that learn invariant and non-invariant features. Node u represents the feature encoder $h_u(\cdot)$. Nodes w^e stand for the posteriors of the classifier parameters given $\mathcal{D}_u^e$, which is the data distribution of environment e transformed by $h_u(\cdot)$. Node w stands for the posterior given the data from the mixture of environments, $\mathcal{D}_u$. (left) when $h_u(\cdot)$ encodes non-invariant features, each environment has a unique posterior of classifier parameters, which has a dependency on on the environment index e; (right) when $h_u(\cdot)$ encodes invariant features, w^e has almost the same posterior with w, which is no longer dependent on the environment index e.

If the feature extractor $h_u(\cdot)$ learns non-invariant features, the data distribution of $\mathcal{D}_u^e$ differs with e. So the posterior $p(w^e|\mathcal{D}_u^e)$ would be different among environments. Then there is a dependency of w^e on e as illustrated in Figure 2 (left) . We further have $p(w^e|\mathcal{D}_u^e) \neq p(w|\mathcal{D}_u)$ because the data distribution of $\mathcal{D}_u^e$ is different from that of $\mathcal{D}_u$. In such the case, the model can not generalize to an unseen environment e' because the $\mathcal{D}_u^{e'}$ can be arbitrary.

The goal of invariant learning is to obtain a extractor $h_u(\cdot)$ that encodes invariant features. With the invariant representation, the data distribution of $\mathcal{D}_u^e$ will be the same for all e. Consequently, the posterior $p(w^e|\mathcal{D}_u^e)$ should be close for each environment and they are all further equivalent to the shared posterior: $p(w^e|\mathcal{D}_u^e) \approx p(w|\mathcal{D}_u)$. Figure 2 (right) illustrates this case by removing the dependency of node w^e on the node e.

Motivated by aforementioned intuition, we propose our Bayesian Invariant Risk Minimization (BIRM):

$$\max_u \sum_e \mathbb{E}_{q_u(w)}[\ln p(\mathcal{D}^e|w,u)] + \lambda\big(\mathbb{E}_{q_u(w)}[\ln p(\mathcal{D}^e|w,u)] - \mathbb{E}_{q_u^e(w^e)}[\ln p(\mathcal{D}^e|w^e,u)]\big), \tag{4}$$

where $q_u(w) \approx p(w|\mathcal{D}_u)$ and $q_u^e(w^e) \approx p(w^e|\mathcal{D}_u^e)$, are the approximate posterior distributions for the classifier given $\mathcal{D}_u$ and $\mathcal{D}_u^e$; the two terms,

$$\mathbb{E}_{q_u^e(w^e)}[\ln p(\mathcal{D}^e|w^e,u)] = \int \ln p(\mathcal{D}^e|w^e,u) q_u^e(w^e) dw^e,$$

$$\mathbb{E}_{q_u(w)}[\ln p(\mathcal{D}^e|w,u)] = \int \ln p(\mathcal{D}^e|w,u) q_u(w) dw$$

are the expected log likelihood of $q_u^e(w^e)$ and $q_u(w)$ on the data from environment e, respectively.

Note that the approximated posteriors $q_u(w)$ and $q_u^e(w^e)$ explicitly depend on u. The first term in Eq. (4) is maximizing the expected log likelihood of the shared posterior $q_u(w)$ of w by optimizing over u. It encourages u to retain as much information as possible to enable $q_u(w)$ to fit the data distribution. The second term in Eq. (4) requires u to learn invariant features. If $h_u(\cdot)$ encodes non-invariant features, the transformed distribution $\mathcal{D}_u^e$ varies among environments. Recall that $q_u^e(w^e)$ is the posterior given $\mathcal{D}_u^e$ and $q_u(w)$ is the posterior given $\mathcal{D}_u$. So $q_u^e(w^e)$ can achieve higher likelihood than $q_u(w)$ on $\mathcal{D}_u^e$. Then we impose a penalty to require $h_u(\cdot)$ to discard non-invariant features.

Note that the vanilla definition of IRM in Eq. (2) is based on a single point estimation of w, which can be highly unstable when data is insufficient. Rather than point estimation, BIRM is induced by the posterior distributions directly, which is less prone to overfitting [5, 8, 33].

Variational Inference. The estimation of the posterior distributions is non-trivial in large models. Here, we approximate them using $q_u^e(w^e)$ and $q_u(w)$ by variational inference. Given a distribution family $\mathcal{Q}$, we approximate the posterior distribution by finding the optimal $q \in \mathcal{Q}$ that maximizes the evidence lower bound (ELBO). The objective function to estimate $q_u^e(w^e)$ is:

$$q_u^e(w^e) = \arg\max_{q' \in \mathcal{Q}} \mathbb{E}_{q'}\big[\ln p(\mathcal{D}^e|w,u) \\ - \mathrm{KL}(q' \| p_0(w))\big], \tag{5}$$

where the first term is to maximize the expected log likelihood of the posterior distribution, and the second term aims to keep q' close to the prior $p_0(w)$. Similarly, the objective function to obtain $q_u(w)$ is:

$$q_u(w) = \arg\max_{q' \in \mathcal{Q}} \sum_e \mathbb{E}_{q'}\big[\ln p(\mathcal{D}^e|w,u) \\ - \mathrm{KL}(q' \| p_0(w))\big]. \tag{6}$$

Following common practice in variational inference (mean field approximation) [10], we choose factorized Gaussian distributions, i.e., $\mathcal{Q} = \{\mathcal{N}(\mu, \Sigma) : \mu = [w_1, ..., w_d]^\top, \Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_d)\}$, where d is the dimension of the classifier parameter w. The prior $p_0(w)$ is set to a Gaussian distribution with zero mean: $\mathcal{N}(0, \sigma I)$. The estimated posteriors by Eq. (5) and (6) are denoted as $q_u(w) = \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$ and $q_u^e(w^e) = \mathcal{N}(\tilde{\mu}^e, \tilde{\Sigma}^e)$.

With the help of variational inference, we are finally able to optimize Eq. (4). Specifically, the training process will iterate among solving Eq. (5), (6), and (4).

The following proposition characterizes the behavior of $q_u^e(w^e)$ and $q_u(w)$ when we learn an invariant u.

Proposition 2. *If $h_u(\cdot)$ does not extract spurious features, as $n_e \to \infty$, $q_u^e(w^e) \xrightarrow{\mathcal{D}} q_u(w)$ and*

$$\mathbb{E}_{q_u(w)}[\ln p(\mathcal{D}^e|w,u)] - \mathbb{E}_{q_u^e(w^e)}[\ln p(\mathcal{D}^e|w,u)] \to 0,$$

where $\xrightarrow{\mathcal{D}}$ indicates convergence in distribution.

The proof of Proposition 2 is in Appendix A. Proposition 2 indicates that if $h_u(\cdot)$ does not extract spurious feature, the penalty will be zero and BIRM only consider the empirical risk of the model. Otherwise, a penalty will be induced to encourage $h_u(\cdot)$ to discard spurious features.

4.2. Variance Reduced Reparameterization

Note that we use Monte Carlo samples from $q_u(w)$ and $q_u^e(w^e)$ to estimate the penalty term in Eq. (4). A common practice is to draw samples by reparameterization trick [29]:

$$w = \tilde{\mu} + \epsilon\tilde{\Sigma}, \quad w^e = \tilde{\mu}^e + \epsilon^e\tilde{\Sigma}^e, \forall e, \tag{7}$$

where $\epsilon, \epsilon^e \sim \mathcal{N}(0, I), \epsilon \perp \epsilon^e, \forall e \in \mathcal{E}_{tr}$. However, in Proposition 2, these two expectation terms are close but conventional reparameterization method may induce high variance during training. Consider that we collect K samples to estimate the expectations, $w_{u,1}, \cdots, w_{u,K}$ from $q_u(w)$ and $w_{u,1}^e, \cdots, w_{u,K}^e$ from $q_u^e(w^e)$; the estimated penalty is computed as:

$$J_K(u) = \frac{1}{K}\sum_{i=1}^{K}\sum_e -\ln p(\mathcal{D}^e|w_{u,i}, u) + \ln p(\mathcal{D}^e|w_{u,i}^e, u). \tag{8}$$

The variance of $J_K(u)$ is characterized as following.

Proposition 3. *By conventional reparamterization in Eq. (7), as $n_e \to \infty$, $\mathbb{V}[J_K] \to c/K$, where c is a constant and $\mathbb{V}[J_K]$ is the variance of J_K.*

Proposition 3 indicates that the variance of estimated penalty J_K is a constant when given K. In this case, we need a large K to make the training algorithm stable. Besides, near the end of training, the expectation of the penalty is close to zero (according to Proposition 2), meaning that the variance can dominate the penalty.

To fix this, we propose the variance reduced reparameterization trick. Our main intuition is to use shared auxiliary noise variable ϵ_s for both w and w^e so that the randomness of sampling can cancel each other after the subtraction. Specifically, we sample $\epsilon_s \sim \mathcal{N}(0, I)$, and use it to parameterize both w_u and w_u^e:

$$w = \tilde{\mu} + \epsilon_s\tilde{\Sigma} \quad w^e = \tilde{\mu}^e + \epsilon_s\tilde{\Sigma}^e, \forall e \tag{9}$$

We name the reparameterization in Eq. (9) the *variance reduced reparameterization trick*. The following proposition characterizes the advantage of this method.

Proposition 4. *By the variance reduced reparameterization in Eq.* (9)*, as $n_e \to \infty$, $\mathbb{V}[J_K] \to 0$, where $\mathbb{V}[J_K]$ is the variance of J_K.*

Comparing Proposition 4 with Proposition 3, we can see that the variance reduced reparameterization can achieve much smaller variance than the conventional method.

4.3. Fast Adaptation

Although the introduction of Bayesian posterior is straightforward and reasonable, it is computation-exhausted to find the ELBO solution of Eq. (5) on different environments at each step. We further borrow ideas of fast adaptation from MAML [18] to estimate $q_u^e(w^e)$ in a more efficient way. Proposition 2 shows that as the training procedure proceeds, $q_u^e(w^e)$ will be closer to $q_u(w)$ when less spurious features are extracted by $h_u(\cdot)$. This makes it possible to perform fast estimation of $q_u^e(w)$ as following:

$$q_u^e(w) = \mathcal{N}(\mu - \nabla_\mu \mathbb{E}_{q_u(w)} \ln p(\mathcal{D}^e|w,u), \Sigma), \tag{10}$$

where $q_u(w) = \mathcal{N}(\mu, \Sigma)$. Here, the mean of $q_u^e(w^e)$, μ^e, is approximated by a step of gradient descent of μ on the data from environment e. The feasibility of fast adaptation is based on the nearness of $q_u^e(w^e)$ to $q_u(w)$ indicated by Proposition 2, which makes single step estimation plausible. By this method, we do not need to estimate $q_u^e(w^e)$ from scratch each time. Note that full algorithm of BIRM is included in Appendix B.

Remark. An existing work [50], proposes Domain-Invariant Learning with Uncertainty (DILU), which also estimates a distribution of classifiers for better OOD performance. Specifically, they randomly draw samples with the same label from each environment and match the outputs of the samples. However, existing IRM works typically consider an extremely challenging task where the labels are noisy [2, 4, 11, 32, 40, 51]. Due to the existence of label noise, DILU can force to align the prediction of samples from different classes, which will hinder the learning of causal features. Though DILU does not fall into the line of IRM methods, we add DILU as a baseline in the experiments in Section 5. The results in Section 5 show that our method outperforms DILU by a large margin.

5. Experiments

In this section, we demonstrate the effectiveness of BIRM on several datasets, one Synthetic dataset and three vision dataset. Details are summarized in Table 1.

Baselines. We compare BIRM with (1) standard Empirical Risk Minimization (**ERM**); (2) three existing IRM methods: **IRMv1** [4], **REx** [32], and **InvRat** [11]; (3) a related domain generalization method: **DILU** [50]; (4) ERM trained on the dataset without spurious feature (**Oracle**).

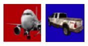
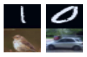
Dataset	Invariant	Spurious	Training	Testing
Synthetic*	X_1	X_2	$\rho^e = 1.0$	$\rho^e = 9.9$
CMNIST	Digit	Color		
ColoredObject	Object	Background		
CifarMnist	CIFAR	MNIST		

Table 1. Illustration of datasets. "Invariant" and "Spurious" stands for the invariant and spurious features. The spurious feature has a strong correlation with the label, as shown in "Training" samples. However, the correlation is reversed in the "Testing" samples to simulate the distributional shift. * For the Synthetic dataset, X_2 is generated with different ρ^e according to Eq. (11).

Sample size	5K	2K	1K	0.5K
Oracle	0.97	0.98	1.02	1.02
ERM	28.40	27.22	30.32	28.66
IRMv1	2.15	4.31	8.76	13.75
REx	5.55	8.65	15.40	15.12
InvRat	2.25	4.15	9.03	13.66
BIRM (Ours)	**1.82**	**2.90**	**3.17**	**3.86**

Table 2. Test MSE on the synthetic dataset. Sample size stands form the amount of training data.

5.1. Synthetic Dataset

The synthetic dataset considers a similar case with [4] where the spurious feature is induced by anti-causal effect. Specifically, the dataset is generated as following:

$$\begin{aligned} &X_1 \sim \mathcal{N}(0, \sigma^2 I), \quad Y = \mathbf{1}^\intercal X_1 + \mathcal{N}(0, \sigma I) \\ &X_2 = Y \cdot \mathbf{1} + \mathcal{N}(0, (\rho^e \sigma)^2 I), \end{aligned} \tag{11}$$

where $X_1, X_2 \in \mathbb{R}^2$ and $y \in \mathbb{R}$. X_1 and X_2 are the invariant and spurious features respectively. I is identity matrix and $\mathbf{1}$ is the vector of 1. ρ^e varies in different environments, indicating that the correlation between the spurious feature, X_2, and Y is unstable. The larger ρ^e is, the weaker correlation between X_2 and Y. The training dataset consists two environments, in which ρ^e is set to 0.5 and 1.0, respectively. In the testing dataset, ρ^e is set to 9.9. A model depending on the spurious feature is expected to perform poorly in the testing dataset. We fit a linear model to predict Y on X_1 and X_2. Then we evaluate the Mean Squared Error (MSE) between the predicted value $\hat{Y}$ and Y: $\mathbb{E}[(Y - \hat{Y})^2]$.

Table 2 shows the results of each method with different amount of training data. The poor performance of ERM on the test dataset indicates that ERM relies on the spurious feature X_2. IRM baselines performs well when the sample size is 5K. However, their performance deteriorates quickly with lessened data. Our BIRM consistently outperforms the baselines IRM, IRMv1, REx and InvRat in all settings. When the data is limited, BIRM improves upon the

Sample size	50K*	40K	30K	20K	15K	10K	5K
Oracle	72.45	71.61	70.19	69.45	68.11	66.99	64.15
ERM	10.80	11.03	11.08	13.58	16.22	18.20	21.04
DILU	50.22	52.31	45.31	44.21	48.92	43.14	43.83
IRMv1	67.45	65.25	63.46	58.67	49.51	35.60	26.19
InvRat	66.35	66.61	61.05	57.25	50.04	34.28	25.42
REx	**69.12**	**69.10**	66.94	63.35	56.50	43.17	32.55
BIRM (Ours)	**69.97**	**69.47**	**69.06**	**67.02**	**66.78**	**66.40**	**60.01**

Table 3. Test accuracy on CMNIST by MLP of hidden size 390 with varied training sample size. IRMv1, REx and InvRat deteriorate quickly with less data. BIRM improves baselines significantly in the data-starving case. *Standard sample size is 50K in [4, 32]

baselines by a significant margin: the test MSE of BIRM is 3.86 given 0.5K training data while the test MSE of other IRM baselines is all larger than 13!

5.2. Vision Datasets

In this section, we evaluate BIRM on three vision classification datasets with spurious features, CMNIST [4], ColoredObject [1, 55] and CifarMnist [34, 44]. Multi-layer-perceptron (MLP) is adopted for CMNIST and ResNet-18 is adopted for ColoredObject and CifarMnist. The datasets and experimental settings follow the conventions in IRM literature [1, 2, 4, 11, 32, 34, 35, 51, 55].

We use *bias ratio* to denote the correlation of the spurious feature with the label [55]. Each dataset contains two training environments and one testing environment. The bias ratio differs across environments, denoted as (r_1, r_2, r_3), where r_1 and r_2 are the training bias ratio, and r_3 is the testing one [55].

CMNIST [4]. CMNIST consists of digit images in 2 classes: 0 and 1. These images are repaint with colored background as a spurious feature. As defined before, the bias ratio is $(0.9, 0.8, 0.1)$ in CMNIST. To make it more challenging, 25% label noise is added to CMNIST [4, 55].

CifarMnist [34, 44]. Each image in CifarMnist is synthesized by concatenating two component images: CIFAR-10 (invariant) and MNIST (spurious). The bias ratio is $(0.999, 0.7, 0.1)$. The label noise ratio is 10% [4].

ColoredObject [1, 55]. ColoredObject is constructed by superimposing eight classes of objects extracted from MSCOCO on a colored background (spurious feature). The bias ratios are $(0.999, 0.7, 0.1)$. 10% label noise is injected.

5.2.1 Results

We summarize the results of CMNIST in Table 3. Note that CMNIST adopted in [4, 32] contains 50K image samples. In this paper, we further reduce the sample size by randomly subsampling. As Table 3 shows, the performance of IRMv1, REx and InvRat drops dramatically if the sample size decreases from 50K to 5K. For example, IRMv1 achieves 67.45% test accuracy when provided with 50K training data, whereas only preserving 26% test accuracy with 5K training samples. In contrast, BIRM can maintain a test accuracy of 60.01% with only 5K training data.

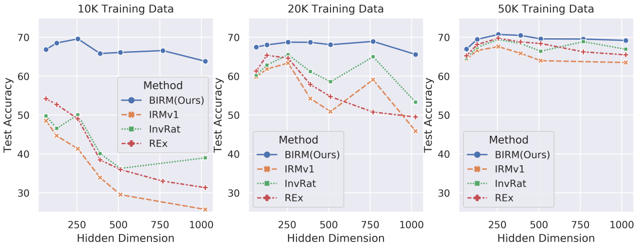


Figure 3. BIRM versus baselines with different model size

Method	ColoredObject	CifarMnist
Oracle	85.3±0.6	83.7±1.5
ERM	49.8±0.4	39.5±0.4
IRMGame	55.7±1.8	46.7±2.1
DILU	56.2±1.7	50.2±1.7
IRMv1	71.4±0.2	51.3±3.0
REx	73.2±2.9	50.1±2.2
InvRat	73.5±1.5	52.3±0.9
BIRM (Ours)	**78.1**±0.6	**59.3**±2.3

Table 4. Test accuracy on ColoredObject and CifarMnist

Figure 3 shows the results of each method on CMNIST with MLP of varied hidden dimension when 10K, 20K and 50K data is provided. We can see that the large IRM baseline models are more likely to fail with insufficient data. For example, in the 10K training data case, the performance of REx drops from 55.2% to 32.4% when the hidden dimension of the model increases from 64 to 1024. Compared with the other baselines, BIRM is more stable as the model hidden dimension increases, i.e., the performance only slightly drops from 67.4% to 63.5% in the 10K training data case according to Figure 3. On CMNIST(20K), BIRM consistently outperforms IRM baselines, surpassing the best of them by over 10% when the hidden dimension is 1024 . These experimental results also provide support for the theoretical findings in Section 3.2 that IRM can easily fail due to overfitting with lessened data or enlarged model.

Table 4 summarizes the results of all methods on ColoredObject and CifarMnist. BIRM outperforms all the baselines significantly. The performance of ERM is only 49.8%, indicating that it heavily relies on the spurious features. IRM baselines, IRMv1, REx, and InvRat, achieving 71.4%, 73.2%, and 73.5% test accuracy, are more robust than ERM. The proposed BIRM improves further to 78.5%.

On CifarMnist, Table 4 shows that ERM only achieves

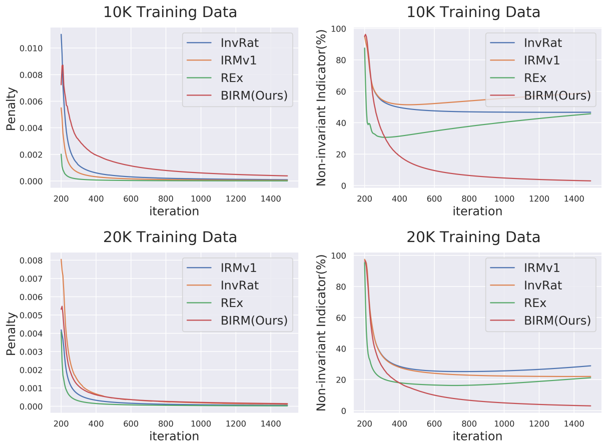


Figure 4. Illustration of training IRM methods on CMNIST [4]. As the training proceeds, the penalty of the IRM baselines, REx, InvRat and REx, vanishes quickly. However, large amount of spurious feature still exist in the feature representation according to the the non-invariant indicator (refer to Section 3.3 for definition). Only BIRM reduces the spurious feature to a low level.

Method	L_2	Early Stopping	Train (%)	Test (%)
IRMv1	0	✗	95.21	21.84
	1×10^{-3}	✗	86.39	35.60
	1×10^{-2}	✗	65.29	49.78
	1×10^{-1}	✗	50.47	50.44
	1×10^{-3}	✓	84.20	41.92
	1×10^{-2}	✓	63.61	50.69
BIRM	1×10^{-3}	✗	66.67	**66.40**

Table 5. Comparison on CMNIST (10K sample size) between BIRM and IRMv1 with L_2 weight decay and early stopping. 10^{-3} is the default weight decay ratio in this paper and [4, 32].

Model	IRMv1	BIRM
CMNIST (MLP)	1.0×	1.5×
COCO (ResNet-18)	1.0×	1.1×

Table 6. Relative training of BIRM VS IRMv1.

39.5% test accuracy. The test accuracy of IRM baselines is barely over 50%. BIRM achieves 59.3% test accuracy, exceeding the best baseline model by nearly 7.0%. Overall, we can see that the task of CifarMnist is harder than ColoredObject. Notably, BIRM also exceeds DILU by a large margin on both CifarMnist and ColoredObject.

5.2.2 More Analysis

The Overfitting of IRM Penalty. Figure 4 illustrates the trends of the IRM penalty and the non-invariant indicator (defined in Section 3.3) during the training with 10K and 20K training data. As the training proceeds, the penalty of all IRM baselines decay to zero. However, the non-invariant indicators are still 20%-40% at the end. This means that the penalty of the IRM is overfitted to zero even though large amount of spurious is retained in the model. In contrast, the non-invariant indicator of BIRM converges towards zero in a pace synchronized with its penalty. Figure 4 clearly shows the advantage of BIRM to alleviate the overfitting when compared with other baselines.

Comparison with Strongly Regularized IRM. We have already shown that Bayesian method improves the performance of IRM by avoiding overfitting. A natural question is whether stronger regularization, i.e., L_2 weight decay or early stopping [42], can also help. Table 5 compares BIRM with strongly regularized IRMv1 on CMNIST with 10K training samples. IRMv1 with 10^{-3} weight decay (the same as [4, 32]) has 35.60% testing accuracy. A weight decay of 10^{-2} can bring the testing accuracy to 49.78%. However, enlarging weight decay further to 10^{-1} can hinder the learning of invariant features because the training accuracy drops to nearly 50%. Early stopping can slightly improve the performance along with L_2 weight decay, nevertheless, it is still far behind BIRM. In conclusion, common regularization techniques cannot achieve comparable performance with Bayesian method on IRM.

Computational Overhead One possible concern for Bayesian inference is the computational overhead. However, the computational overhead is acceptable in BIRM: (1) *Training*. We estimate the posterior of the *classifiers* (NOT on the feature extraction layers); therefore, the relative computational overhead is small, especially for models with large feature extractors (as shown in Table 6). (2) *Inference*. With the robust feature extractor obtained from training, we *do NOT need to sample* from the posterior during inference; therefore there is no overhead.

6. Conclusions

In this paper, we investigated the failure of IRM formulation in overparameterized deep learning models, and showed that a key reason is that the IRM penalty degenerates due to the overfitting of large models. To remedy this problem, we proposed a Bayesian formulation, BIRM, which averages over the uncertain model parameter regimes to avoid overfitting. We showed that this method stabilizes invariant feature learning. We have conducted extensive experiments to demonstrate that BIRM improves the original IRM formulation for relatively large models.

Code Availability

Our codes are available at https://github.com/linyongver/Bayesian-Invariant-Risk-Minmization.

References

[1] Faruk Ahmed, Yoshua Bengio, Harm van Seijen, and Aaron Courville. Systematic generalisation with group invariant predictions. In *International Conference on Learning Representations*, 2020. 1, 2, 7

[2] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pages 145–155. PMLR, 2020. 1, 2, 3, 6, 7

[3] Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R Varshney. Empirical or invariant risk minimization? a sample complexity perspective. *arXiv preprint arXiv:2010.16412*, 2020. 1, 2, 3

[4] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. 2019. 1, 2, 3, 4, 6, 7, 8

[5] Jean-Yves Audibert. Progressive mixture rules are deviation suboptimal. *Advances in Neural Information Processing Systems*, 20:41–48, 2007. 4, 5

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015. 1

[7] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018. 1

[8] José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009. 1, 2, 5

[9] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006. 2

[10] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. 2, 4, 5

[11] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *International Conference on Machine Learning*, pages 1448–1458. PMLR, 2020. 1, 2, 3, 6, 7

[12] Yining Chen, Elan Rosenfeld, Mark Sellke, Tengyu Ma, and Andrej Risteski. Iterative feature matching: Toward provable domain generalization with logarithmic environments. *arXiv preprint arXiv:2106.09913*, 2021. 2

[13] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021. 2, 3

[14] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, pages 1–10, 2021. 1

[15] Harrison Edwards and Amos Storkey. Towards a neural statistician. In *International Conference on Learning Representations*, 2017. 2

[16] Cong Fang, Hanze Dong, and Tong Zhang. Over parameterized two-level neural networks can learn near optimal feature representations. *arXiv preprint arXiv:1910.11508*, 2019. 1

[17] Cong Fang, Hanze Dong, and Tong Zhang. Mathematical models of overparameterized neural networks. *Proceedings of the IEEE*, 109(5):683–703, 2021. 1

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 6

[19] Yanwei Fu, Xiaomei Wang, Hanze Dong, Yu-Gang Jiang, Meng Wang, Xiangyang Xue, and Leonid Sigal. Vocabulary-informed zero-shot and open-set learning. *IEEE transactions on pattern analysis and machine intelligence*, 42(12):3136–3152, 2019. 1

[20] 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. 1

[21] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, 2015. 2

[22] Roger Ghanem, David Higdon, and Houman Owhadi. *Handbook of uncertainty quantification*, volume 6. Springer, 2017. 2

[23] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013. 1

[24] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021. 1, 2, 3

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3

[26] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993. 2

[27] Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pages 4069–4077. PMLR, 2021. 2

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 3

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. 2, 5

[30] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International Conference on Machine Learning*, pages 5436–5446. PMLR, 2020. 2

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[32] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi

Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International Conference on Machine Learning*, pages 5815–5826. PMLR, 2021. 1, 2, 3, 6, 7, 8

[33] Guillaume Lecué. Suboptimality of penalized empirical risk minimization in classification. In *International Conference on Computational Learning Theory*, pages 142–156. Springer, 2007. 4, 5

[34] Yong Lin, Qing Lian, and Tong Zhang. An empirical study of invariant risk minimization on deep models. *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*, 2021. 1, 2, 7

[35] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Heterogeneous risk minimization. In *International Conference on Machine Learning*, pages 6804–6814. PMLR, 2021. 2, 7

[36] Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015. 1

[37] Chengzhi Mao, Mia Chiquier, Hao Wang, Junfeng Yang, and Carl Vondrick. Adversarial attacks are reversible with natural supervision. 2021. 2

[38] Chengzhi Mao, James Wang, Kevin Xia, Hao Wang, Junfeng Yang, Elias Bareinboim, and Carl Vondrick. Causal transportability for neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[39] Lu Mi, Hao Wang, Yonglong Tian, and Nir Shavit. Training-free uncertainty estimation for neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2

[40] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012, 2016. 1, 6

[41] Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2020. 1, 2, 3

[42] 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. *arXiv preprint arXiv:1911.08731*, 2019. 8

[43] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4

[44] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. volume 33, pages 9573–9585, 2020. 3, 7

[45] Malcolm Strens. A bayesian framework for reinforcement learning. In *International Conference on Machine Learning*, volume 2000, pages 943–950, 2000. 2

[46] Hao Wang, Yifei Ma, Hao Ding, and Yuyang Wang. Context uncertainty in contextual bandits with applications to recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 2

[47] Hao Wang, Chengzhi Mao, Hao He, Mingmin Zhao, Tommi S Jaakkola, and Dina Katabi. Bidirectional inference networks: A class of deep bayesian networks for health profiling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 766–773, 2019. 2

[48] Hao Wang, Xingjian Shi, and Dit-Yan Yeung. Natural-parameter networks: A class of probabilistic neural networks. volume 29, 2016. 2

[49] Hao Wang and Dit-Yan Yeung. A survey on bayesian deep learning. *ACM Computing Surveys (CSUR)*, 53(5):1–37, 2020. 2

[50] Zehao Xiao, Jiayi Shen, Xiantong Zhen, Ling Shao, and Cees Snoek. A bit more bayesian: Domain-invariant learning with uncertainty. In *International Conference on Machine Learning*, pages 11351–11361. PMLR, 2021. 6

[51] Chuanlong Xie, Fei Chen, Yue Liu, and Zhenguo Li. Risk variance penalization: From distributional robustness to causality. *arXiv e-prints*, pages arXiv–2006, 2020. 1, 2, 3, 6, 7

[52] Yilun Xu and Tommi Jaakkola. Learning representations that support robust transfer of predictors. *arXiv preprint arXiv:2110.09940*, 2021. 2, 3

[53] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7343–7353, 2018. 2

[54] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 3

[55] Dinghuai Zhang, Kartik Ahuja, Yilun Xu, Yisen Wang, and Aaron Courville. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pages 12356–12367. PMLR, 2021. 1, 7, 3

[56] Guojun Zhang, Han Zhao, Yaoliang Yu, and Pascal Poupart. Quantifying and improving transferability in domain generalization. *arXiv preprint arXiv:2106.03632*, 2021. 1, 2, 3

[57] Xiao Zhou, Weizhong Zhang, Zonghao Chen, Shizhe Diao, and Tong Zhang. Efficient neural network training via forward and backward propagation sparsification. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[58] Xiao Zhou, Weizhong Zhang, Hang Xu, and Tong Zhang. Effective sparsification of neural networks with global sparsity constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3599–3608, 2021. 1

Bayesian Invariant Risk Minimization

Supplementary Material

A. Theoretical Details

A.1. Proof of Proposition 1

Proof. First we prove that any element in Ω is a stationary point of IRM defined in (2). Let $(\bar{w}, \bar{u})$ be an arbitrary element in Ω. According to Definition 1, we have

$$\mathcal{R}^e(\bar{w}, \bar{u}) = 0, \forall e \in \mathcal{E}_{tr}. \tag{12}$$

Note that

$$\mathcal{R}^e(w, u) \geq 0, \forall e \in \mathcal{E}_{tr}, w, u \tag{13}$$

So it follows that

$$\mathcal{R}^e(w, \bar{u}) \geq 0, \forall e \in \mathcal{E}_{tr}, w$$

Then

$$\bar{w} \in \arg\min \mathcal{R}^e(w, \bar{u}) \geq 0, \forall e \in \mathcal{E}_{tr}$$

So $(\bar{w}, \bar{u})$ stratifies the constrain in (2). At the same time, (12) and (13) already suffice to show that $(\bar{w}, \bar{u})$ is the minimum of the objective. Then we conclude that $(\bar{w}, \bar{u})$ is a stationary point solution of IRM defined in (2).

The first argument that IRM degenerates to ERM in Ω follows directly from the proof above.

Suppose there exists another collections of (w', u') that matches the constrain in (2),

$$\Omega' := \{(w', u') | w \in \arg\min_{w} \mathcal{R}^e(w, u'), \forall e \in \mathcal{E}_{tr}, (w', u') \notin \Omega\}.$$

Note that the elements in Ω are excluded from Ω' for simplicity. So $\Omega \cap \Omega' = \emptyset$ and $\Omega \cup \Omega'$ includes all (w, u) that satisfies the constrain in (2). By (12) and (13) we know that

$$\begin{aligned} &\forall (w', u') \in \Omega', (\bar{w}, \bar{u}) \in \Omega, \exists e \in \mathcal{E}_{tr}, \\ &\quad \mathcal{R}^e(u', w') > \mathcal{R}^e(\bar{u}, \bar{w}) = 0 \end{aligned}$$

It follows that

$$\begin{aligned} &\forall (w', u') \in \Omega', (\bar{w}, \bar{u}) \in \Omega, \\ &\sum_e \mathcal{R}^e(u', w') > \sum_e \mathcal{R}^e(\bar{u}, \bar{w}), \end{aligned}$$

This means any element $(\bar{w}, \bar{u})$ in Ω has smaller objective than any element (w', u') in Ω'. This means IRM in (2) will not pick any element in Ω'.

We already know that $\Omega \cup \Omega'$ is the whole set of (w, u) matches the constrain. So IRM will pick arbitrary element in Ω . Notably, by Assumption 2 and Definition 1, we do not impose any invariant constrain in Ω. Then we prove our first argument that IRM reduces to ERM in Ω. □

A.2. Proof of Corollary 1

Proof. By definition,

$$\sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(w, u) + \lambda \|\nabla_w \mathcal{R}^e(w, u)\|^2 \geq 0.$$

Thus, if $\bar{w}, \bar{u}$ satisfies

$$\sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(\bar{w}, \bar{u}) + \lambda \|\nabla_w \mathcal{R}^e(\bar{w}, \bar{u})\|^2 = 0,$$

we have

$$(\bar{w}, \bar{u}) \in \arg\min_{w,u} \sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(w, u) + \lambda \|\nabla_w \mathcal{R}^e(w, u)\|^2.$$

By Assumptions 1,2, we have

$$\sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(\bar{w}, \bar{u}) = 0.$$

By Assumptions 3, $\nabla_w R^e(\bar{w}, \bar{u})$ exists.
If $\nabla_w \mathcal{R}^e(\bar{w}, \bar{u}) = v \neq 0$, we have

$$\lim_{\epsilon \to 0} \frac{\mathcal{R}^e(\bar{w} + \epsilon v, \bar{u}) - \mathcal{R}^e(\bar{w}, \bar{u})}{\epsilon} = \|v\|^2$$

Then for $\|v\|^2/2$, we have there exists $\delta > 0$ such that for all $t \in (-\delta, \delta)$,

$$\frac{\mathcal{R}^e(\bar{w} + tv, \bar{u}) - \mathcal{R}^e(\bar{w}, \bar{u})}{t} > \frac{\|v\|^2}{2}$$

By choosing $t = -\frac{\delta}{2}$,

$$\mathcal{R}^e(\bar{w} - \delta v/2, \bar{u}) < -\frac{\delta \|v\|^2}{4} + \mathcal{R}^e(\bar{w}, \bar{u}) = -\frac{\delta \|v\|^2}{4} < 0,$$

which contradicts the definition of R^e.

Thus, $\|\nabla_w \mathcal{R}^e(\bar{w}, \bar{u})\| = 0$,

$$(\bar{w}, \bar{u}) \in \arg\min_{w,u} \sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(w, u) + \lambda \|\nabla_w \mathcal{R}^e(w, u)\|^2.$$

□

A.3. Proof of Proposition 2

Proof. By definition, when h_u extracts invariant feature, the data distribution in each environment is the same.

Since model is well-specified, assume the data generating distribution satisfies,

$$p(y|\mathbf{x}) = p(y|h_u(\mathbf{x}), w_*).$$

Then by Bernstein–von Mises theorem, when $n_e \to \infty$,

$$w^e \to \mathcal{N}(w_*, F_e^{-1}/n_e),$$

where w_* is optimal solution, F_e is the Fisher information matrix.

Note that when $p(x^e)$ is the invariant across environments, the Fisher information matrix $F_e = F$ is a constant matrix.

Thus, for any $e_1, e_2 \in |\mathcal{E}_{tr}|$, we have

$$w^{e_1} \xrightarrow{\mathcal{D}} w^{e_2}.$$

Since w uses subset of all $\mathcal{D}$ (only n_e samples), we have

$$w \to \mathcal{N}(w_*, F^{-1}/n_e),$$

$$w \xrightarrow{\mathcal{D}} w^e.$$

□

A.4. Proof of Proposition 3

Proof. Similar to Proposition 2, by Bernstein–von Mises theorem, when $n_e \to \infty$,

$$w \to \mathcal{N}(w_*, F^{-1}/n_e),$$

where w_* is optimal solution, F is the Fisher information matrix.

Thus,

$$w = O_p(1/n_e)$$

$$\ln p(y|\mathbf{x}, w, u) = O_p(1/n_e)$$

$$\ln p(\mathcal{D}^e|w, u) = \sum_{i=1}^{n_e} \ln p(y|\mathbf{x}, w, u) = O_p(1)$$

Thus,

$$J_K = O_p(1/K)$$

□

A.5. Proof of Proposition 4

Proof. By Proposition 2,

$$w_e \xrightarrow{\mathcal{D}} w.$$

By our parameterization,

$$w_e \xrightarrow{a.s.} w.$$

For any x, y,

$$\ln p(y|\mathbf{x}, w, u) - \ln p(y|\mathbf{x}, w_e, u) = o_p(1/n_e)$$

Thus,

$$\ln p(\mathcal{D}^e|w, u) - \ln p(\mathcal{D}^e|w_e, u) = o_p(1)$$

$$J_K = o_p(1/K) = 0$$

□

A.6. More Theoretical Results

A.6.1 Failure of other IRM variants

Corollary 2 (Failure of REx)**.** *Under Assumptions 1,2,3, $\forall(\bar{w}, \bar{u}) \in \Omega$, the following equality holds:*

$$(\bar{w}, \bar{u}) \in \arg\min_{w,u} \sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(w, u) + \lambda \mathbb{V}_e[R^e(w, u)]$$

Proof. Since $\mathcal{R}^e(\bar{w}, \bar{u}) = 0$ for all e, we also have $\lambda \mathbb{V}_e[R^e(w, u)] = 0$.

On the other hand $\mathrm{Var}_e[\mathcal{R}^e(w, u)] \geq 0$.

We obtain that

$$(\bar{w}, \bar{u}) \in \arg\min_{w,u} \sum_{e \in \mathcal{E}_{tr}} \mathcal{R}^e(w, u) + \lambda \mathbb{V}_e[\mathcal{R}^e(w, u)]$$

□

Corollary 3 (Failure of InvRat)**.** *Under Assumptions 1,2,3, $\forall(\bar{w}, \bar{u}) \in \Omega$, the following equality holds:*

$$(\bar{w}, \bar{u}) \in \arg\min_{w,u} \max_{w^e} \sum_{e \in \mathcal{E}_{tr}} \big[\mathcal{R}^e(w, u) + \lambda(\mathcal{R}^e(w, u) - \mathcal{R}^e(w^e, u))\big]$$

Proof. By definition,

$$\mathcal{R}^e(w, u) - \mathcal{R}^e(w^e, u) \geq 0, \quad \mathcal{R}^e(w, u) \geq 0, \quad \mathcal{R}^e(w^e, u) \geq 0.$$

Since for $\bar{w}, \bar{u}$, $\mathcal{R}^e(\bar{w}, \bar{u}) = 0$, we also have

$$\mathcal{R}^e(\bar{w}, \bar{u}) - \mathcal{R}^e(w^e, \bar{u}) = 0.$$

Thus,

$$(\bar{w}, \bar{u}) \in \arg\min_{w,u} \max_{w^e} \sum_{e \in \mathcal{E}_{tr}} \big[\mathcal{R}^e(w, u) + \lambda(\mathcal{R}^e(w, u) - \mathcal{R}^e(w^e, u))\big]$$

□

A.6.2 Relaxed Version of Corollary 2

Here, we provide a counterpart of Corollary 2 based on a relaxed version of Assumption 2.

Assumption 4 (Relaxed Sufficient Capacity)**.** *The parameters w and u have sufficient capacity to fit the training data: there exist $\bar{w}$ and $\bar{u}$, such that $\forall e \in \mathcal{E}_{tr}, \mathcal{R}^e(\bar{w}, \bar{u}) \leq \epsilon$.*

Then we have the following results:

Corollary 4 (Failure of REx)**.** *Under Assumptions 1 and 4 $\forall(\bar{w}, \bar{u}) \in \Omega$, then the penalty of relax is upper bounded as following:*

$$\mathbb{V}_e[R^e(w, u)] \leq \epsilon^2$$

Proof.

$$\mathbb{V}_e[R^e(w, u)] \leq \frac{1}{|\mathcal{E}_{tr}|} \sum_e [R^e(w, u)]^2 \leq \epsilon^2.$$

□

B. Implementation Details

B.1. Definition of Non-invariant Indicator

Consider the testing dataset $\mathcal{D}^{test} := \{\mathbf{x}_i, y_i\}_{i=1}^{n_{test}}$. For simplicity, suppose each $\mathbf{x}$ is the concatenation of an invariant and spurious feature, $\mathbf{x}_i = [x_i^{inv}, x_i^s]$. In CMNIST (described in Section 5), x^{inv} corresponds to the digit shape, "0" or "1"; x^s refers to the color, green or red. Given a function f that predicts the class of $\mathbf{x}$, we can measure how much $f(\mathbf{x})$ relies on x^s by replacing x^s with the other color $\bar{x}^s$ while fixing x^{inv}, i.e., x^s is green and $\bar{x}^s$ is red (vice versa). The *non-invariant indicator* is defined as following:

$$\frac{1}{n^{test}} \sum_{i=1}^{n^{test}} \mathbf{1}(f([x_i^{inv}, x_i^s]) \neq f([x_i^{inv}, \bar{x}_i^s])),$$

where $\mathbf{1}(\cdot)$ is the indication function. Non-invariant indicator is the percentage of samples that changes its prediction while spurious feature is changed.

B.2. Algorithm

The full algorithm is summarized as following:

Algorithm 1 BIRM: Bayesian Invariant Risk Minimization

```
Input: Feature extractor $h_u$, classifier $g_w$, prior $p_0$, collection of data from multiple environments $\{\mathcal{D}^e\}_e^{\mathcal{E}_{tr}}$.
Output: The learned $h_u$, classifier $g_w$.
    Initialisation :
while TRUE do
    for $e$ in $1, \dots, |\mathcal{E}_{tr}|$ do
        Sample a batch of data $(\mathbf{x}^e, y^e)$ from $\mathcal{D}^e$
        Obtain the feature representation $h_u(\mathbf{x}^e)$
    end for
    Update $q_u(w)$ by Eq. (6).
    for $e$ in $1, \dots, |\mathcal{E}_{tr}|$ do
        Update $q_u^e(w)$ by Eq. (10).
    end for
    Sample from $q_u(w)$ and $q_u^e(w)$ to optimize Eq. (4) with variance reduction in Eq. (9).
    Update $u$ to minimize Eq. (4).
end while
return $u$, $q_u(w)$
```

B.3. Efficient Implementation

As discussed in [4], a scalar classifier w with fixed value $w = 1.0$ is enough to monitor the invariance. In our algorithm, we can even achieve simpler implementation by fixing the mean of the w $w = 1.0$ as while tuning the variance of the $q_u(w)$, Σ, as a hyper-parameter. Together with the fast adaptation method in Section 4.3, it makes the implementation of BIRM rather efficient and convenient.

B.4. Detailed Experimental Settings

Datasets. The images in CMNIST and ColoredObject are of size $3 \times 32 \times 32$; the images in CifarMnist are of size $3 \times 64 \times 32$. In CMNIST, ColoredObject, and CifarMnist, the label noise is injected by randomly changing the label to the other values (a uniformly generated random label different from the ground-truth). Following the conventional practice [4,32], the spurious feature is generated after injecting label noise.

Experimental Details. For all experiments in this paper, we report the accuracy of the last step [24,55]. All reported statistics are average value over 3 random seeds. We fix the regularization penalty weight as 10^4 for all the experiments and datasets as [4, 32], because we find the performance is not sensitive to the regularization penalty weight larger than 10^3. In CMNIST, we use gradient descent with Adam [28] by a learning rate 4×10^{-4} [4]. The total training steps are 1500. The penalty is added on the step 200. 10^{-3} weight decay is imposed on the MLP. These settings are all consistent with [4, 32]. In ColoredObject and CifarMnist, we adopt stochastic gradient by SGD (0.9 momentum) with learning rate 0.01. The batch size is 512. The total training procedure lasts for 1000 steps and the learning rate is reduced by $1/10$ at the middle of the training. ResNet-18 [25] is adopted for ColoredObject and CifarMnist. Similarly, 10^{-3} weight decay is imposed on the neural networks.

C. Extra Experimental Results

C.1. Empirical Evidence for Proposition 1

Here we present additional experimental results on IRMv1. The IRMv1 penalty vanishes at the end as the model begins to memorize the data, however the non-invariant indicator also remains 70% - 80%. In other words, the model still heavily relies on spurious features while IRM penalty can not detect it. These results are consistent with those reported in Section 3.3.

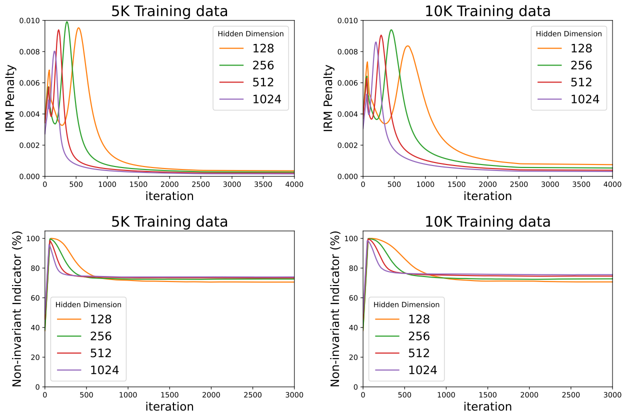


Figure 5. Illustration of training ERM on CMNIST [4] with 3-layer MLP of different hidden dimensions. The penalty of IRM (IRMv1) is measured but not applied to the objective. As the training of ERM proceeds, the IRM penalty decays to zero while the non-invariant indicator shows the existence of large amount of spurious feature in the model. The IRM penalty vanishes faster with larger model and less training data.

C.2. Visualisation of Different IRM methods

Figure 6 illustrates the comparison of different methods on several typical samples from ColorMnist, by explaining their predictions via GradCAM [43]. GradCAM visualizes the component of image which contributes most to the model's prediction, i.e. which area in the image the model is paying attention to. We can see that the ERM method mistakenly relies on the background. IRMv1 and REx can alleviate this issue, however, still fail at some cases (e.g., the 9th sample). BIRM improves the performance of IRMv1 and REx further (e.g., bring the attention of the 9th sample to the upper right corner where the object lives).

C.3. Ablation study

We conduct ablation study to investigate the effect of each ingredient of BIRM. The major improvement of BIRM over the existing baselines is to obtain the posterior distributions w^e and w rather than their point estimates. The *Probabilistic* column of Table 7 shows that the distributional nature of BIRM contributes a lot to the performance improvement. The second ingredient of BIRM is the variance reduced reparameterization trick introduced in Section 4.2. The *VR* column of Table 7 shows that BIRM can drops from 67.2% to 63.86% when *VR* is removed. The third ingredient, fast adaptation (FA), aims to reduce the computational complexity of BIRM by estimating the posterior of w^e with first order approximation as discussed in Section 4.3. Table 7 shows that the test performance drops by only 0.32% when fast adaptation is applied on BIRM.

Probabilistic	VR	FA	Test Accuracy(%)
✓	✓	✓	67.02
✓	✓	✗	67.32
✓	✗	✓	63.86
✗	✗	✗	57.26

Table 7. Ablation study of BIRM on MLP with 390 hidden dimension in CMNIST with 20K data. *Probabilistic* stands for whether to estimate the distribution (or the point estimation) of w and w^e in Eq. (4); *FA* stands for Fast Adaptation; *VR* stands for variance reduction reparametrization trick.

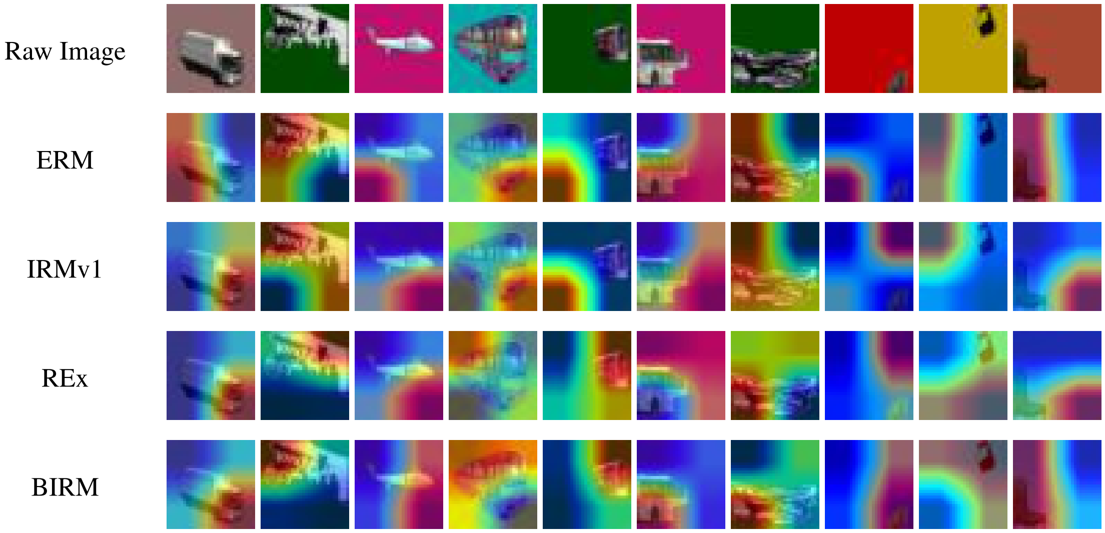


Figure 6. Illustration of attention map of different methods on ColoredObjects test samples. The red attention mask indicates heavy reliance; the blue one refers to ignorance. We can see that the ERM method mistakenly relies on the background. IRMv1 and REx can alleviate this issue, however, still fail at some cases (e.g., the 9th sample). BIRM improves the performance of IRMv1 and REx further (e.g., bring the attention of the 9th sample to the upper right corner where the object lives).