Exploring Disentanglement and Invariance in the Face of Co-variate Shift

Jishnu Ray Chowdhury
jraych2@uic.edu
University of Illinois Chicago

Ishan Bhatnagar
ibhatn2@uic.edu
University of Illinois Chicago

1 ABSTRACT

Despite the strong predictive performance of Machine Learning models, they are still subject to spurious correlations. Working under the I.I.D. assumption they are often hard to generalize to an out of distribution test data. In this project, we tackle the problem of co-variate shift, where the test data is from a different distribution than the training data. Particularly, we approach this problem from a causal framework. We compare IRMv1 [2], CoRe [12], ICP [21], and Entropy Penalty (EP) [3] on different settings. Furthermore, we experiment with disentangled representations, and we try to enhance classification results by gating the features of intermediate hidden state representations based on their influence on the classification probabilities. The magnitude of influence of a feature is computed based on the difference between the classification probabilities obtained from using the features as they are and that obtained from intervening on certain features by counterfactually altering them. Although most of our results are negative, we provide potential reasons for such results and motivate future directions in this area. The code is published in Github: https://github.com/JRC1995/Causal-Inference.

2 INTRODUCTION

Spurious correlations (like that between human birth-rate and the number of storks [18]) are not expected to persist if certain environmental conditions change or if there are shifts in the distributions. Standard machine learning training assumes that both the training and testing data are I.I.D. They are not guaranteed to generalize well to out of distribution data, and they are susceptible to spurious correlations that may exist in the training data owing to some selection bias. Recent examples like one-pixel attack [24] is a stark display of the brittleness of some standard deep learning models in computer vision. Similarly in Natural Language Processing, some of our best models can suffer from the 'Clever Hans Effect'. For example, BERT [10] had been shown to exploit surface-level heuristics [19] (present due to some selection bias) to get high performance in argument comprehension.

If we want our models to be **robust** for sensitive real-world tasks, we must find ways to make our models rely on more stable correlations i.e. correlations which are **invariant** across all environments. Furthermore, it is reasonable to think that the relevant correlations are between the output and the latent causal variables in the 'concept space' rather than the 'pixel space' (or its equivalent in the context of language or other domain). Thus, we also think that our models can benefit from creating **disentangled representations** [4] of input data since disentangled representations can potentially expose the latent causal variables.

There is no strong consensus on the definition of 'disentanglement', but generally 'disentangled representations' are used to denote a form of representation where the different 'aspects' of a data are represented as marginally independent variables in a manner such that one aspect can be changed without influencing the other. Concretely, in a 3D image, the disentangle-able aspects or 'factors' maybe pose, style, lighting, color, position of object, geometrical orientations, shapes etc. In a disentangled representation of the image, we would want each feature to correspond to one factor of variation. For example, one feature may correspond to the geometric orientation of an object, and by changing that associated dimension one may be able to rotate that object without changing other aspects of the image. These aspects can be also thought of as latent data generating factors, and they can be even seen from a causal framework [25] as independent causal mechanisms. Disentanglement may also provide us a way to reduce high-dimensional data into a low dimensional set of features which can be treated as causal variables by some downstream causal estimators that require explicit causal variables to work on.

Thus in this project we are working on the problem of robust supervised learning by exploiting invariant correlations and disentanglement of latent variables.

Our contributions:

(1) We provide empirical comparisons among IRM, CoRe, EP under different settings and on different datasets.
(2) We investigate two novel extensions. In one extension we use disentangled representations with IRM. In the other extension, we consider the influence of hidden state features based on the difference in the output variable when the features are intervened on, to limit the contribution of features that are, on average, less influential. We compare the extensions with the above-mentioned baselines along with various ablations.
(3) We present discussions of the potential flaws and limitations of all the models discussed here and provide future directions.

3 RELATED WORKS

The idea of invariant correlations is closely related to causality. For example, we would think that across different environments, the true causal mechanism tends to be invariant and thus the conditional distribution of the target variable given its direct causes also remains unchanged regardless of any change in the distribution of the target variable or the direct causal variables. Similar ideas had been roughly discussed in multiple works in different manners and under different names ("autonomy", "independence of cause and mechanism", "modularity" etc.) [1, 9, 11, 14, 20, 22]. Peters et al.[21] exploit this connection to find direct causes of the outcome variable. CoRe[12] regularizes on the conditional variance of model prediction and loss given the identity and class of the object in an

image to suppress the potential nuisance factors. Arjovsky et al.[2], also based on similar ideas, seek to find representations which elicit a classifier which is invariantly optimal for all environments, for robust prediction. Outside of causal frameworks, Arpit et al.[3] take an information-theoretic framework for out of distribution generalization. Bengio et al.[5] also work on a similar idea that if the estimated causal graph is close to the true causal graph then the model will adapt faster when the distribution shifts (because the correctly estimated portions of the causal graph should be invariant across distributions). While disentanglement [4] is not often seen from a causal perspective, there are a few works that take a causal perspective on disentanglement [6, 15, 25] (where the disentangled latent variables are related to causal mechanisms) or treat the disentangled latent features as variables for a causal graph [5]. Some of the above-mentioned methods are elaborated below:

3.1 Invariant Causal Prediction (ICP)

Peters et al. [21] propose this method to determine the direct causal parents of a target variable. It begins with a null-hypothesis which is true for a given subset (S) of input variables if there exists a set of causal-coefficients (γ) for the variables in S such that the variables are independent of the noise term for all environments. They perform this hypothesis test for all possible subsets of the set of input variables and treat the intersection of all the subset for which the null-hypothesis couldn't be rejected for a given significance (α) of its p-value as the estimated set of causal variables. $1-\alpha$ can be used as the confidence interval for the co-efficients of the estimated causal variables. One concrete method to use this idea in practice is described below.

(1) For each environment $e \in \mathcal{E}$ and for each possible member S in the powerset of X (the set of input variables), do the following:
(a) Separate observations to set I_e (where e is active) and I_{e^-} (where e isn't active).
(b) Predict Y_e (as $\hat{Y}_e$) from S_e with its regression weights that were estimated by training on I_{e^-} using least-square linear regression.
(c) Use Chow test [8] on Y_e (actual observations) and $\hat{Y}_e$.
(d) Reject the null hypothesis for a particular subset S if the p-value for the test is below $\alpha/|\mathcal{E}|$ for any e.
(2) Find the intersection of all accepted subsets of the set of all input variables in step (1).
(3) The confidence interval for all rejected set should be set to null and for the estimated causal co-efficients (the regression co-efficients) for the variables in each accepted set should be set to $1-\alpha$. Finally, we adjust for residual variances for the non-zero estimated causal (regression) co-efficients of the final accepted variables using pooled data.

3.2 Conditional variance Regularizer (CoRe)

Heinze-Deml et al. [12] take a causal view of the data-generating processes for tasks like image classification. We can think of it in terms of a causal graph where the label Y and the classification object identifiable by some identity factor ID form the direct parents of the latent generative factors. The latent generative factors can be categorized into 'conditionally invariant' (core) features and style features. While the core features are invariant when conditioned on Y and ID, the style-features are also influenced by random environmental noise. As an example, in the task of classifying whether a given person in an image is a celebrity or not, the ID can be any value uniquely picking out that person, the core features can be any stable features of that specific person, and the style feature maybe glasses that the person may sometimes wear and sometimes not. The target is to learn the core features while ignoring the style features. The authors use the variance of the final layer output (and also, the loss) conditioned on the class and ID as the regularization term which can be used as a constraint with Lagrange multipliers.

3.3 Invariant Risk Minimization (IRM)

Arjovsky et al. [2] focus on enforcing the learning of invariant (across environments) correlations for robust machine learning and out of distribution generalization. If a data representation ϕ successfully captures the invariant correlations and filters out environmental noise, then we would expect that:

$$E[Y^e|\phi(X^e) = h] = E[Y^{e'}|\phi(X^{e'}) = h] \quad \forall e, e' \in \mathcal{E}$$

Where $\mathcal{E}$ is the set of all environments. For standard loss functions, this expectation can be modeled by an optimal classifier. Thus for our purpose, we desire a representation ϕ which elicits an invariant predictor $w \cdot \phi$ where w is an optimal classifier simultaneously for all environments. We can formulate this in terms of an optimization function as:

$$min_{\phi:X\to H, w:H\to Y} \sum_{e\in\mathcal{E}_{tr}} R_e(w\cdot\phi) \quad \text{subject to}$$

$$w \in argmin_{w':H\to Y} R_e(w'\cdot\phi)$$

$\mathcal{E}_{tr}$ is the set of training environments and R_e is the empirical risk minimization term (can be a standard loss function). A more practical proxy for the above formulation is:

$$min_{\phi:X\to Y} \sum_{e\in\mathcal{E}_{tr}} R_e(\phi) + \lambda \cdot ||\nabla_{w|w=1.0} R_e(w\cdot\phi)||_2^2$$

This is the IRMv1 formulation. Here, we treat w as a 'dummy classifier' frozen as 1.0. The gradient norm denotes how optimal w is for ϕ in e and it penalizes the model if for any environment $w = 1.0$ is not optimal for ϕ. λ is the penalty weight which balances between the ERM term and the invariance of the predictor $1.\phi$. For the exact mathematical journey that was token to get here, see [2].

3.4 Entropy Penalty (EP)

Entropy Penalty [3] takes an information-theoretic framework to introduce a new regularization term which happens to be very similar to CoRe. Following the Information Bottleneck theory [26] of deep learning, Entropy Penalty attempts to reduce the mutual information between two layers. Intuitively, the deeper layers should be better at modeling more abstract features of the input data at the cost of losing the information of the specific details of less abstract earlier layers. Following the assumption of Gaussian distribution the idea can be simplified to reducing the class-conditional variance of the deeper layers. Empirically, it was found that the penalty worked best on the class-conditional variance of the first hidden state layer representation [3].

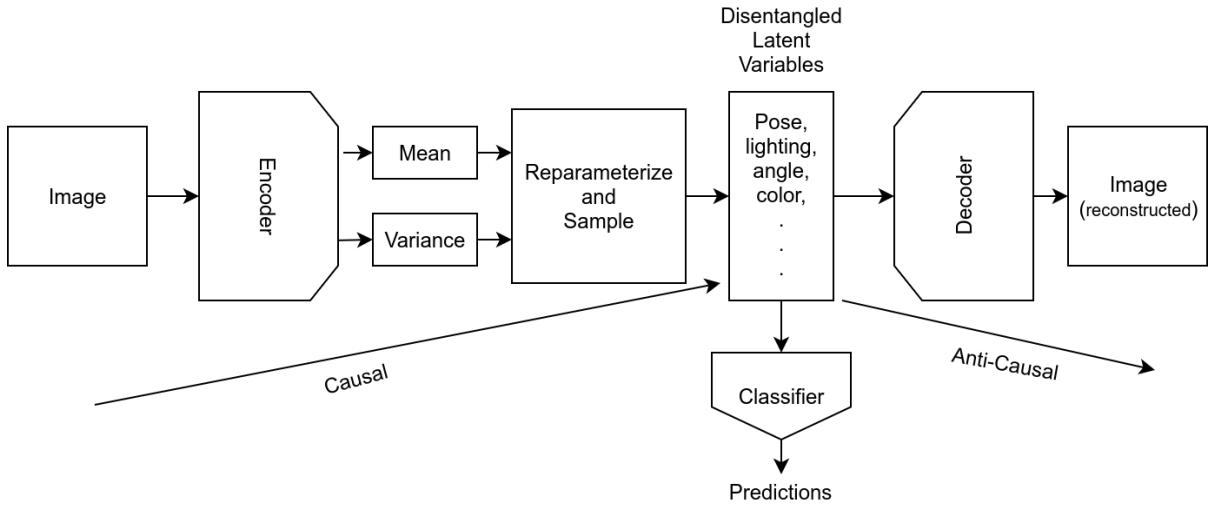


Figure 1: β-VAE with a classifier on top of the latent variables

4 PROBLEM DESCRIPTION

We are interested in exploring the problem of the co-variate shift. Consider the case where we have a training set with pairs X (input) and Y (Outputs) sampled from some distribution $P(X, Y)$. Then, our target is to predict Y' from X' where X' and Y' are sampled from some other distribution $P'(X', Y')$.

5 APPROACH

In this section, we discuss our main approaches.

5.1 β-VAE and IRM

Much of the works discussed thus far is about learning to focus on certain factors of variation and ignore others. For example, CoRe [12] motivates the discovery of 'core' latent features and suppressing the variance caused by nuisance latent factors related to 'style' features. This would require the model to disentangle 'core' features and 'style' features somehow. Similarly, in the colored MNIST setting, it seems plausible that the disentanglement of 'color' and 'shape' would be immensely helpful for the classifiers. Supported by this intuition, we formulate a multi-task problem where we simultaneously train to create a disentangled representation and a classifier on top of it. To elaborate further, we use a β-VAE [13] based self-supervised objective to create disentangled latent variables. A β-VAE modifies the standard ELBO objective of a standard VAE to encourage statistical independence of the latent factors:

$$\mathcal{L}(x; \phi, \theta) = E_{q_\phi(z|x)}[logP_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x)||p(z))$$

Here is x is the original input data and z is the latent variable, and D_{KL} is KL-divergence. β is a hyperparameter which when > 1 helps in disentanglement by further encouraging the posterior distribution $q_\phi(z|x)$ to be similar to the prior $p(z)$ wherein the prior is an isotropic Gaussian distribution. Thus the KL-divergence term multiplied by a $\beta > 1$ ends up encouraging the statistical independence among the dimensions of the latent variable z.

We then use a classifier on top of the latent variable z for the supervised classification objective along with the IRM penalty. The multi-task formulation is a linear combination of the two objectives along with the IRM regularization. We show our setup in Figure 1. From a causal framework, the direction of estimating the latent variables can be thought of as anti-causal where we predict the potential data generating causes from the given data, and the direction of reconstructing the original image from the latent variable can be thought of as a case of causal prediction where we predict the data from the data-generating causes.

While not the same, similar approaches of using the latent variable of a VAE for classification had been taken before in previous works [16, 27].

5.2 Counterfactual Influence

We also consider another approach where we try to suppress the contributions of potential nuisance factors by focusing on features that make the biggest difference in predictions when intervened on. First, we consider an intermediate layer representation of the latent variables estimated by the encoder of a β-VAE. Following [6], the features of the latent variable can be interpreted as exogenous variables and that of the intermediate representations (formed by feeding the latent variable to a neural network layer) can be thought of as endogenous variables which are functional assignments (the functions being neural network connections) of the features of the latent variable. Now let's say we have two images X and X' from different classes, and after encoding them we have two latent variables of n dimensions - $Z_{1:n}$ and $Z'_{1:n}$ respectively. Let's consider a functional assignment ϕ such that:

$$\phi(Z_{1:n}) = h_{1:n}$$

$$\phi(Z'_{1:n}) = h'_{1:n}$$

Now we consider a transformation t such that:

$$t(h_i) = h'_i$$

For every feature i in $h_{1:n}$, we can then compute their influence on the output classes as:

$$I_i = ||f(h_1, h_2, ...t(h_i), ...h_n) - f(h_1, h_2, ...h_i, ..h_n)||_2$$

Where f is a classifier function that transforms the intermediate layer input into class scores (unnormalized). Thus we will have an n-dimensional vector I where each feature i corresponds to the magnitude of the influence of intervening on (applying transformation t) the corresponding feature h_i. Intuitively, the equation can be thought of as analogous to the equation for estimating the average treatment effect. It can also be seen as answering the counterfactual question:

What would be the difference in class scores if the i^{th} feature was from a class different than what it actually is?

Based on I we prepare a gating vector g where each feature g_i in g gates the feature h_i in h based on the value of I_i:

$$g_i = sigmoid(I_i - mean(I))$$

We subtract the mean so that the output of the sigmoid function is guaranteed to range between 0 and 1. We then use another classifier f' for the main prediction as:

$$Y' = f'(g \cdot h)$$

Here h is the n-dimensional vector representing $h_{1:n}$). Intuitively, g performs a gating function over the features $h_{1:n}$. The value of the gate g_i is low for feature h_i if it has low I_i value (low influence) but the value of the gate g_i is high for feature h_i if it has high I_i value (high influence). Thus the features from the intermediate representation h are gated in accordance to their counterfactual influence on the class probabilities. Here we assume that the features with consistently high influences over different environments tend to be the core-features as opposed to being related to nuisance factors.

Furthermore, we also consider the relative differences of the magnitude of influence I_i in I should not vary too much from sample to sample. Here we consider the invariance of the 'core features' throughout environments (and different samples). Thus we also introduce a new regularization term on the norm of the variance of I throughout the current training mini-batch. Overall we train two objective functions:

$$\mathcal{L}_1 = R(Y, f'(h \cdot g)) + \gamma ELBO$$

$$\mathcal{L}_2 = R(Y, f(h)) + \eta ||Var(I)||_2$$

(Here ELBO corresponds to the β-VAE objective function). For the first objective, we do not update the gradients for parameters in function f. For the second objective, we only update the parameters for function f. This ensures that the gates g depends only on the influence based on classifier f. R is any risk minimization function; in our case, we use cross-entropy. In test time, we don't have access to classes of the current samples. So we cannot have counter-factual features h' from samples from different classes. Instead, in test time, we use the mean I estimated from the training samples.

6 DATASET DESCRIPTIONS

6.1 Synthetic Data

The Synthetic Data is generated by an artificial causal model which can be expressed with the following structural equations:

$$Z^e \leftarrow \mathcal{N}(0, e^2)$$

$$X_1^e \leftarrow Z^e \cdot W_{h1} + \mathcal{N}(0, e^2)$$

$$Y_e \leftarrow Z^e \cdot W_{hy} + X_1^e \cdot W_{1y} + \mathcal{N}(0, \sigma_y^2)$$

$$X_2^e \leftarrow Z^e \cdot W_{h2} + Y^e \cdot W_{y2} + \mathcal{N}(0, \sigma_2^2)$$

For the models that we use, Z_e is a latent confounder. The input features has 10 dimensions. The equations for the first 5 features are in the same form as X_1^e (which form the direct causal parents of Y_e), and the last 5 features are of the same form as X_2^e. Y_e and Z_e are also 5 dimensional continuous valued vector. For our comparisons we consider three settings:
(1) FOU: Fully Observed (F) i.e. the latent confounder and its coefficients are inactive; Homoskedastic (O) i.e. σ_y^2 is set as e^2 (turned environmentally dependent); Unscrambled (U) i.e. X is fully observed (no noise added).
(2) PEU: Partially Observed (P) i.e. the latent confounder is active; Heteroskedastic (E) i.e. σ_2^2 is set as e^2; Unscrambled (U).
(3) PES: Partially Observed (P); Heteroskedastic (E); Scrambled (S) i.e. X is 'scrambled' by multiplying with an orthogonal matrix S.

We ran the experiments a total of 10 times on IRM, ICP, and ERM (Empirical Risk Minimization) using linear regression-based methods. Our posted results are the average of 10 runs. In each run, we prepared three environments with .2, 2, and 5 as environmental noise multipliers respectively. We drew 1000 samples from each environment. CoRe and EP need to be conditioned on categorical classes which are not present for this task, so we don't use it in this data. The other methods are either only for classification or rely on the reconstruction loss of image data none of which are quite suited for this task.

The synthetic data generating process is the same one as used by Arjovsky et al. [2].

6.2 Colored MNIST 1.0

Colored-MNIST is a partially-synthetic dataset based upon MNIST (handwritten digit classification dataset). Following [2], we create intermediate synthetic labels y' by setting it as 1 if the digit is 0 to 4, else 0. The final label y is created by flipping y' with probability 0.25. We prepare three environments (first two for training, third for test) where we sample the color id z by flipping the value of y' with probability 0.2,0.1, and 0.9 respectively. The color of the digit is set to red if z=1 else green. Therefore, in each particular environment, the color and the label form a stronger correlation which ERM will no doubt try to exploit, but this correlation is also spurious - it varies with the environments (the correlation is reversed in the test set). Although the digits and y-labels have a weaker correlation it is more stable across different environments. ICP could not be tested in this setting because it requires explicit causal variables.

	FOU		PEU		PES	
Model	CE	NCE	CE	NCE	CE	NCE
ERM	0.096	0.096	7.802	7.818	7.876	0
ICP	2.005	0.005	10	0	5	0
IRM	0.022	0.019	2.123	3.431	2.83	0

Table 1: Results on synthetic data. CE stands for causal errors. NCE stands for non-causal errors.

Model	Training	Test
ERM	87.40%	16.63%
IRM	70.54%	67.28%

Table 2: Reproduction of results on colored MNIST 1.0 (mean of 10 runs)

6.3 Colored MNIST 2.0

We also investigate the models on a different version of colored MNIST which we generate based on Arpit et al.[3]. 10, 000 are randomly sampled images from MNIST is used as test-set, and rest for training and validation. Both the foreground and background colors are varied to generate the dataset. For the training/validation set, for each class one of two randomly chosen colors is randomly assigned for the foreground. Similarly one in between two randomly pre-chosen colors for each class is randomly assigned to the background. The pixels were binarized and a zero-mean Gaussian noise was added with a standard deviation of 0.04. In the test set, however, random background and foreground colors are chosen independently of the class. Thus, while there is some artificially induced correlation between colors and the class in the training or validation set, there is no such thing in the test set. In this case, we can consider that the training and validation set is a mixture of different environments with each environment having samples with a specific correlation between color combinations and the class.

7 EXPERIMENTS

The results of the different models on the synthetic data are given in Table 1. Causal Error is the mean square error between the actual causal co-efficients of the parents of Y in X and the estimated co-efficients. The non-causal error is the average of the norm of the estimated co-efficients (ground truth is 0) of the non-causal variables (those which are not direct causal parents of Y). Qualitatively, on FOU settings we found that ICP behaves better than others for a lot of samples, but for the remaining ones it ends up rejecting all the variables and thereby amassing up a high error. Consistent with Arjovsky et al.[2], we too discover the conservative behavior of ICP where most candidates for causal parents are too easily rejected.

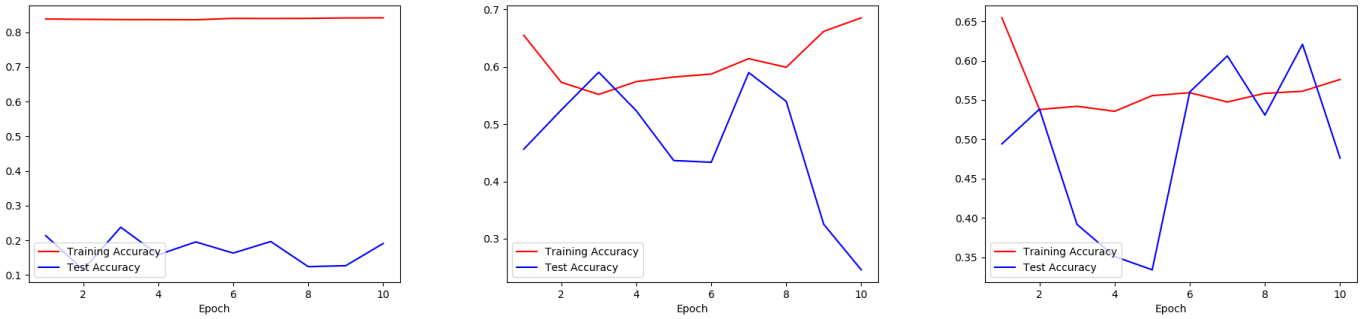
In Table 2, we more or less reproduced the performance of IRM from the paper [2] on Colored MNIST 1.0 by using similar settings as mentioned in the official repository [1]. On this dataset, we used a 2-layered feed-forward neural network. ERM, as expected, overfits on the training set.

Arjovsky et al. [2] use an interesting policy for IRM where it uses a penalty weight (λ) of 1.0 for the first 100-200 steps allowing the ERM part to improve and then increase λ to a very high value (1000-9000). Without this policy, the training is not very stable. Another thing to note here is that all these models in Table 2 were trained with full-batch. How the training would go for IRM in a mini-batch setting with stochastic gradient descent is still not clear. It is important to know how it works in a mini-batch training setup because most large scale deep-learning applications rely on it. Thus, we trained the models in different mini-batch settings on different hyperparameters. Arjovsky et al. [2] provide a formulation for an unbiased estimate of the gradient norm in a mini-batch setting but it requires knowledge of the environment where the batches are coming from which is not always known. We tried training IRM on both cases with mini-batches from particular known environment and unbiased gradient norm calculations, and also in the case of mixed-up batches with no knowledge of which sample is from which environment. In both cases, we find that the mini-batch training dynamics is similarly unstable. If the penalty is too low, the ERM term dominate and the model overfits on training data. If the penalty is high, the model starts to 'overfit' on the test set (sometimes getting around 70% in test set but around 50% in training set), but after some epochs, the ERM term starts to improve and dominate the IRM penalty term. Although sometimes it reaches a sweet point with relatively balanced training set error and test set error, it does not stay there for long (and practically we cannot discover these points without using the test set as validation). We present some graphs showing the mini-batch training dynamics of IRM in figure 2. We did try some systematic strategies to organize mini-batches for better stability but to no avail.

Next, we set up a β-VAE with 5-layered convolutional Encoder and a 5-layer deconvolutional Decoder; both with batch normalizations in between. We then compare all the baselines and our new extensions on colored MNIST 1.0 and colored MNIST 2.0 with comparable settings in Table 3 and Table 4 respectively. For the models without 'β-VAE' in the name, we just used the classifier and relevant extension on the latent variable estimated by the encoder without taking into account the decoder or the overall β-VAE objective. The models with 'β-VAE' in the name were further regularized by the β-VAE objective. We train them all in a mini-batch setting. The exact details can be found in the code that we have shared.

As expected, the ERM model underperforms for both colored MNIST 1.0 and 2.0 by failing to generalize to the adversarially prepared out-of-distribution test data. But, surprisingly none of the models really perform decently. As can be seen, the IRM model while getting the best on the test set in Table 3 underfits the most on the development set (ideally, we would want it to perform well in any distribution including the training and validation distribution not just the test distribution). This happens when we train the IRM model with a different architecture and in a mini-batch settings. Interestingly, IRM works similar to ERM on colored MNIST 2.0 which is not specifically prepared for IRM with simple environments unlike colored MNIST 1.0. CoRe performs similarly to ERM in both cases. For CoRe, we used only the class-conditioned variance of prediction as the penalty because we do not have any extra

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



(a) $\lambda = 1.0$ **(b)** $\lambda = 300$ **(c)** $\lambda = 1000$

Figure 2: Mini-batch Training Dynamics of IRM

Model	Dev	Test
ERM	83.88%	13.03%
IRM	54.96%	60.41%
EP	75.04%	23.31%
CoRe	83.13%	17.49%
β-VAE	79.22%	33.79%
β-VAE + IRM	47.83%	51.15%
Counterfactual	77.40%	40.20%
Counterfactual-β-VAE	79.84%	31.13%

Table 3: Results on colored MNIST 1.0

Model	Dev	Test
ERM	100%	11.81%
IRM	100%	12.30%
EP	49.63%	12.1%
CoRe	100%	12.15%
β-VAE	92.23%	11.22%
β-VAE + IRM	13.54%	9.91%
Counterfactual	100%	12.36%
Counterfactual-β-VAE	73.77%	11%

Table 4: Results on colored MNIST 2.0

'ID' feature to rely on. The result of Entropy Penalty is the most unexpected especially on MNIST 2.0 where it performs pretty well in the original paper [3]. The β-VAE based models do not seem to perform very well either, and when combined with IRM it performs worse. The counterfactual model and the counterfactual β-VAE models incorporate the extensions from subsection 5.2. Counterfactual β-VAE seems to perform similar to β-VAE. The counterfactual model seems to have the best balance of performance but it is still far from ideal.

8 REASONS FOR POOR PERFORMANCE

All the models introduced so far has certain limitations and shortcomings. ICP [21] works with low-scale settings where the causal variables are exposed but otherwise, it is difficult to use. Furthermore, it seems too conservative in its predictions. CoRe [12] was mainly tested in CelebA dataset where the identity of celebrity was used as an ID along with the class to condition the variance on. Outside these specialized setups without any 'ID' feature, we found it difficult to gain much from CoRe as seen in Table 3 and Table 4. We were able to reproduce the performance of IRM from the original paper [2] in Table 1 and Table 2 but we had a difficult time to get similar performance gain in other settings; particularly when using mini-batch and when using different adversarially prepared datasets. Neither the original paper [2] nor the official repository show any experiments conducted on mini-batch settings where we find it to be struggling. Also the equation presented for unbiased gradient norm estimate in mini-batch settings requires us to know from which environment a sample is sampled from which may not be available in a lot of cases. EP [3] is based on the information bottleneck principle but it lacks a theoretical guarantee for learning invariant correlations. It can still be potentially susceptible to spurious correlations. Its result in Table 4 is rather surprising given that it performs much better in the original paper [3]. It may be possible that its performance gain is sensitive to certain hyperparameters and thus more hyperparameter tuning and investigation may be needed. For the β-VAE models it is still questionable if they learn the relevant disentangled representations. More investigation may be needed in this area. Furthermore, there are also potential theoretical issues [17] with learning disentangled representations in a standard fashion. Moreover, even if we have disentangled representations we still need a model to properly utilize them, potentially by discovering the true causal relations between the latent disentangled factors and the output variable. Which is why standard β-VAE-only models are not expected to perform well. However, adding IRM isn't improving the performance either - the different objective functions may be conflicting with each other. The

counterfactual-based extensions may require more thought. It still has theoretical shortcomings - it does not have a strong explicit bias for learning invariant correlations as an example but there are also other potential issues that require some refinements. Moreover, its training objectives are a bit complex which can complicate the loss landscape.

9 FUTURE WORKS

Multiple future directions that can be taken from here. Instead of the multi-task formulation for β-VAE based classification, we can try pre-training the β-VAE first and then apply transfer learning for classification (which would a more standard and natural approach). We can explore better models for disentanglement like β-TCVAE [7]. We can do more exhaustive hyperparameter tuning and experiments on other datasets beyond MNIST. We can try to further develop the extensions presented here. Another interesting related direction to explore is: "Analysis by Synthesis" [23] where a Naive Bayes-like classifier is used but the likelihood in the Bayes' rule is modeled by Variational Autoencoders (with a different VAE for samples from a different class).

REFERENCES

[1] John Aldrich. 1989. Autonomy. *Oxford Economic Papers* 41, 1 (1989), 15–34.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant Risk Minimization. arXiv:stat.ML/1907.02893

[3] Devansh Arpit, Caiming Xiong, and Richard Socher. 2019. Predicting with High Correlation Features. *arXiv preprint arXiv:1910.00164* (2019).

[4] Y. Bengio, A. Courville, and P. Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (Aug 2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[5] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. 2019. A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms. arXiv:cs.LG/1901.10912

[6] Michel Besserve, Rémy Sun, and Bernhard Schölkopf. 2018. Counterfactuals uncover the modular structure of deep generative models. *arXiv preprint arXiv:1812.03253* (2018).

[7] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*. 2610–2620.

[8] Gregory C Chow. 1960. Tests of equality between sets of coefficients in two linear regressions. *Econometrica: Journal of the Econometric Society* (1960), 591–605.

[9] A Philip Dawid and Vanessa Didelez. 2005. *Identifying the consequences of dynamic treatment strategies*. Technical Report. Research Report.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Trygve Haavelmo. 1944. The probability approach in econometrics. *Econometrica: Journal of the Econometric Society* (1944), iii–115.

[12] Christina Heinze-Deml and Nicolai Meinshausen. 2017. Conditional Variance Penalties and Domain Shift Robustness. arXiv:stat.ML/1710.11469

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *ICLR* 2, 5 (2017), 6.

[14] Kevin D Hoover. 1990. The logic of causal inference: Econometrics and the conditional analysis of causation. *Economics & Philosophy* 6, 2 (1990), 207–234.

[15] Niki Kilbertus, Giambattista Parascandolo, and Bernhard Schölkopf. 2018. Generalization in anti-causal learning. *arXiv preprint arXiv:1812.00524* (2018).

[16] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*. 3581–3589.

[17] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2018. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359* (2018).

[18] Robert Matthews. 2000. Storks deliver babies (p= 0.008). *Teaching Statistics* 22, 2 (2000), 36–38.

[19] Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355* (2019).

[20] Judea Pearl. 2000. *Causality: models, reasoning and inference*. Vol. 29. Springer.

[21] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. 2016. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78, 5 (2016), 947–1012. https://doi.org/10.1111/rssb.12167 arXiv:https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssb.12167

[22] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. 2012. On causal and anticausal learning. *In proceedings of ICML* (2012).

[23] Lukas Schott, Jonas Rauber, Matthias Bethge, and Wieland Brendel. 2018. Towards the first adversarially robust neural network model on MNIST. *arXiv preprint arXiv:1805.09190* (2018).

[24] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* (2019).

[25] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. 2019. Robustly Disentangled Causal Mechanisms: Validating Deep Representations for Interventional Robustness. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 6056–6065. http://proceedings.mlr.press/v97/suter19a.html

[26] Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*. IEEE, 1–5.

[27] Chris Varano and Lytton Ave. 2017. Disentangling Variational Autoencoders for Image Classification.