# PREDICTING WITH HIGH CORRELATION FEATURES

**Devansh Arpit, Caiming Xiong, Richard Socher**
Salesforce Research
darpit@salesforce.com

## ABSTRACT

It has been shown that instead of learning actual object features, deep networks tend to exploit non-robust (spurious) discriminative features that are shared between training and test sets. Therefore, while they achieve state of the art performance on such test sets, they achieve poor generalization on out of distribution (OOD) samples where the IID (independent, identical distribution) assumption breaks and the distribution of non-robust features shifts. In this paper, we consider distribution shift as a shift in the distribution of input features during test time that exhibit low correlation with targets in the training set. Under this definition, we evaluate existing robust feature learning methods and regularization methods and compare them against a baseline designed to specifically capture high correlation features in training set. As a controlled test-bed, we design a colored MNIST (C-MNIST) dataset and find that existing methods trained on this set fail to generalize well on an OOD version this dataset, showing that they overfit the low correlation color features. This is avoided by the baseline method trained on the same C-MNIST data, which is designed to learn high correlation features, and is able to generalize on the test sets of vanilla MNIST, MNIST-M and SVHN datasets. Our code is available at https://github.com/salesforce/corr_based_prediction

## 1 INTRODUCTION

It is known that deep networks trained on clean training data (without proper regularization) often learn spurious (non-robust) features which are features that can discriminate between classes but do not align with human perception (Jo & Bengio, 2017; Geirhos et al., 2018a; Tsipras et al., 2018; Ilyas et al., 2019). An example of non-robust feature is the presence of desert in camel images, which may correlate well with this object class. More realistically, models can learn to exploit the abundance of input-target correlations present in datasets, not all of which may be invariant under different environments. Interestingly, such classifiers can achieve good performance on test sets which share the same non-robust features. However, due to this exploitation, these classifiers perform poorly under distribution shift (Geirhos et al., 2018a; Hendrycks & Dietterich, 2019) because it violates the IID assumption which is the foundation of existing generalization theory (Bartlett & Mendelson, 2002; McAllester, 1999b;a).

The research community has approached this problem from different directions. In part of domain adaptation literature (Eg. Ganin & Lempitsky (2014)), the goal is to adapt a model trained on a source domain (often using unlabeled data) so that its performance improves on a target domain that contains the same set of target classes but under a distribution shift. There has also been research on causal discovery (Hoyer et al., 2009; Janzing et al., 2009; Lopez-Paz et al., 2017; Kilbertus et al., 2018) where the problem is formulated as identifying the causal relation between random variables. This framework may potentially then be used to train a model that only depends on the relevant features. However, it is often hard to discover causal structure in realistic settings. Adversarial training (Goodfellow et al., 2014; Madry et al., 2017) on the other hand aims to learn models whose predictions are invariant under small perturbations that are humanly imperceptible. Thus adversarial training can be seen as the worst-case distribution shift in the local proximity of the original training distribution.

We consider the situation in which the distribution of input features that have low correlation with labels in training set undergo a shift in their distribution during test time. The intuition behind picking this definition of distribution shift is that dominant correlations are by definition present more universally across samples and should therefore be relatively more representative of the correct label. Under this situation, we study the behavior of existing regularization techniques designed for robust feature learning and avoiding overfitting, and compare it against a baseline that is designed to find input features that correlate strongly with corresponding labels. Our experimental results show that deep network trained using existing regularization methods on a colored version of MNIST dataset (see appendix A for samples) are unable to generalize well on a distribution-shifted version of the colored MNIST dataset, while the baseline method generalizes well on this test set along with the test sets of vanilla MNIST, MNIST-M, SVHN .

## 2 Baseline Method: Identifying Dominant Correlations

Here we formulate the objective of the baseline regularization method we use in our experiments which is aimed at learning features that correlate strongly with labels. We then study the behavior of this baseline method. Specifically, let $f(\mathbf{x})$ represent the prediction of our model, then the objective of the baseline is to minimize,

$$J(\theta) = \mathbb{E}[(f_\theta(\mathbf{x}) - y)^2] + \frac{\beta}{K}\sum_{k=1}^{K} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}(\mathbf{x}|y=k)}[(f_\theta(\mathbf{x}) - \mu_k)^2] + \lambda\|\theta\|^2 \tag{1}$$

where $\mu_k := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}(\mathbf{x}|y=k)}[f_\theta(\mathbf{x})]$ and $\mathcal{D}$ denotes the data distribution. For the purpose of analysis on synthetic datasets in this section, we use a linear model $f_\theta(\mathbf{x}) := \theta^T\mathbf{x}$. When using a deep network in experiments, we apply the correlation based regularization corresponding to $\beta$ to the first hidden layer of the network. For convolutional networks, a mini-batch has dimensions $(B, C, H, W)$, where we denote B– batch size, C– channels, H– height, W– width. In this case, we reshape this tensor to take the shape $(B \times H \times W, C)$ and treat each row as a hidden vector $\mathbf{h}$.

### 2.1 Theoretical Analysis

We now theoretically study the behavior of Eq. 1 on two synthetic datasets designed to provide insights into the subjective quality of the representation learned.

### 2.1.1 Synthetic Dataset A

The baseline regularization should encourage the neural network to pick features that are dominantly present in class samples and able to discriminate between samples from different classes. To formalize the above intuition, we consider the following synthetic data generating process where the data samples $\mathbf{x} \in \mathbb{R}^d$ and labels $y$ are sampled from the following distribution,

$$y \sim \{-1, 1\} \qquad x_i \sim \begin{cases} \mathcal{N}(y, \sigma^2) & \text{with probability } p_i \\ \mathcal{N}(-y, \sigma^2) & \text{with probability } 1 - p_i \end{cases} \tag{2}$$

where $i \in \{1, 2, \cdots, d\}$, $y$ is drawn with uniform probability, and $\mathbf{x} = [x_1, x_2, \cdots, x_d]$ is a data sample. Also, all $x_i|y$ are independent of each other. Thus depending on the value of $p_i$, a feature $x_i$ has a small or large amount of information about the label $y$. Specifically, values of $p_i$ close to 0.5 do not tell us anything about the value of $y$ while values close to 0 and 1 can reliably predict its value. Here we make the assumption that features with $p_i$ closer to 0.5 are non-robust features whose distribution may shift during test time, while features with $p_i$ closer to 0 and 1 are robust ones. Thus we would ideally want a trained model to be insensitive to non-robust features. The theorem below shows how the model parameters depend on input dimensions for the optimal parameters when training a linear regression model $f_\theta(\mathbf{x}) := \theta^T\mathbf{x}$ using the baseline objective.

**Theorem 1** *Let $\theta^*$ be the minimizer of $J(\theta)$ in Eq. 1 where we have used synthetic dataset A. Then for a large enough $d$, $\theta^* = \mathbf{M}^{-1}|2\mathbf{p} - \mathbf{1}|$, where $\mathbf{M} := \mathbf{\Sigma} + \lambda\mathbf{I} + \beta(\sigma^2\mathbf{I} + 4diag(\mathbf{p} \odot (\mathbf{1} - \mathbf{p})))$, such that $\mathbf{\Sigma}$ is a positive definite matrix if*[1] *$p_i \notin \{0, 0.5, 1\}$ for all $i$.*

[1] This assumption is needed due to technicality.

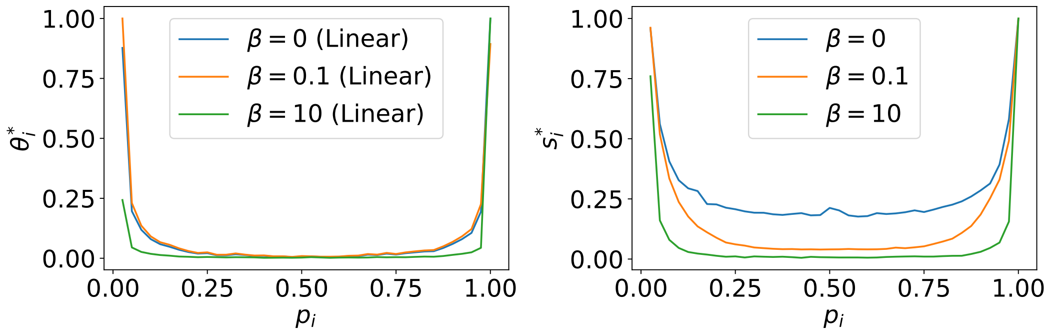

Figure 1: Sensitivity $s_i^*$ of output $f_{\theta^*}(\mathbf{x})$ with respect to input dimensions $x_i$ vs. the probability $p_i$ (controlling correlation between input dimension $i$ and target) for synthetic dataset A (Eq. 2). Left plot shows $\theta_i^*$ (same as sensitivity) computed for a trained linear model. Right plot shows sensitivity computed for a trained MLP. The baseline regularization acts as a filter, suppressing the sensitivity of both these models to weak correlation features ($p_i$ close to 0.5).

As an implication of the above statement, since $\mathbf{M}^{-1}$ is a full rank matrix, aside from the effects due to $\mathbf{\Sigma}$ (which is data dependent and beyond our control), $\theta_i^*$ can in general be non-zero for all input and output correlations. This is especially the case when $\beta = 0$ (no regularization). When using a sufficiently large $\beta$, we find that $\theta_i^*$ gets reduced for larger values of $p_i(1 - p_i)$, i.e., when $p_i$ is closer to 0.5. Thus the baseline regularization helps suppress dependence of the learned model on non-robust (low correlation) features.

We also verify that this behavior also holds for deep networks. In this case the regularization is applied to the representation of the first hidden layer of the network. We conduct experiments with both linear and deep models on samples drawn randomly from synthetic dataset A. The input data is in 500 dimensions and we set the value of $p_i$ for each $i$ to be uniformly from $[0, 1]$ and fix it henceforth. We then randomly sample 15000 input-target pairs from this dataset with $\sigma^2 = 0.0001$. We train a linear regression model and a 3 hidden layer perceptron (MLP) of width 200 with ReLU activation for 1000 iterations using Adam optimizer with learning rate 0.001 and weight decay 0.00001.

In figure 1 (left), we plot the parameters $\theta_i^*$ vs. $p_i$ for the linear regression model. Since the same analysis cannot be done for deep networks, we use the perspective that the output-input sensitivity $\mathbf{s}^*$, where $s_i^* := \mathbb{E}_{\mathbf{x}}\left[\left|\frac{\partial f_{\theta^*}(\mathbf{x})}{\partial x_i}\right|\right]$, is equal to $\theta_i^*$ for linear regression. So for deep networks, we plot $s_i^*$ vs. $p_i$ instead as shown in figure 1 (right). In both models, we normalize the sensitivity values so that the maximum value is 1 for the ease of comparison across different $\beta$ values. Both for linear and deep models, we find that the sensitivity profile goes to 0 away from $p_i = 0$ and 1 when applying the baseline regularization with larger values of coefficients $\beta$; this effect being more dramatic for deep networks. Thus the baseline regularization helps suppress the dependence of model on non-robust (low correlation) features.

### 2.1.2 Synthetic Dataset B

We now consider the following binary classification problem where the data samples $\mathbf{x} \in \mathbb{R}^d$ and labels $y$ are sampled from the following distribution,

$$y \sim \{-1, 1\} \qquad x_i \sim \begin{cases} \mathcal{N}(y, \sigma^2) & \text{with probability } p_i \\ \mathcal{N}(y, k\sigma^2) & \text{with probability } 1 - p_i \end{cases} \tag{3}$$

where $i \in \{1, 2, \cdots, d\}$, $y$ is drawn with uniform probability, and $\mathbf{x} = [x_1, x_2, \cdots, x_d]$ is a data sample. Once again, all $x_i|y$ are independent of each other. Thus depending on the value of $p_i$ and $k$, a feature $x_i$ has a small or large variance. We would ideally like the model to avoid dependence on dimensions with high variance because they are non-robust and therefore have a lower correlation with the label. The theorem below shows how the model parameters depend on input dimensions

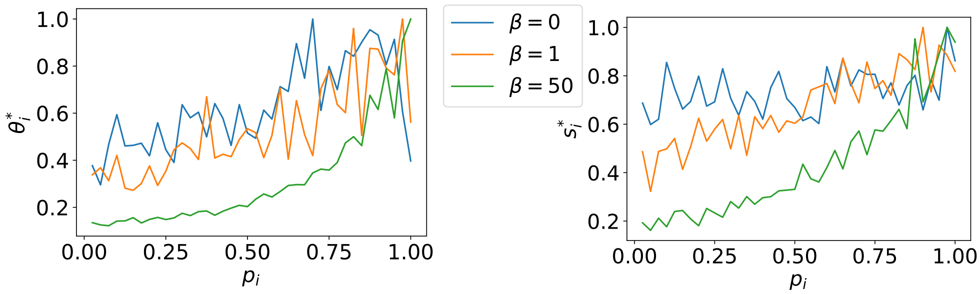

Figure 2: Sensitivity $s_i^*$ of output $f_{\theta^*}(\mathbf{x})$ with respect to input dimensions $x_i$ vs. the probability $p_i$ (deciding the choice between feature with variance $\sigma^2$ vs. $10\sigma^2$) for synthetic dataset B (Eq. 3). Left plot shows $\theta_i^*$ (same as sensitivity) computed for a trained linear model. Right plot shows sensitivity computed for a trained MLP. Baseline regularization suppresses the sensitivity of both these models to large variance features ($p_i$ close to 0).

for the optimal parameters when training a linear regression model $f_\theta(\mathbf{x}) := \theta^T \mathbf{x}$ using the baseline objective.

**Theorem 2** *Let $\theta^*$ be the minimizer of $J(\theta)$ in Eq. 1 where we have used synthetic dataset B. Then for a large enough d, $\theta^* = \mathbf{M}^{-1}\mathbf{1}$, where, $\mathbf{M} := \mathbf{\Sigma} + \lambda \mathbf{I} + \beta\sigma^2 diag(\mathbf{p} + k(\mathbf{1} - \mathbf{p}))$, such that $\mathbf{\Sigma}$ is a positive definite matrix.*

Once again, we find that $\theta_i^*$ is non-zero for all dimensions of the input. Assume without loss of generality that $k > 1$. Then using a sufficiently large $\beta$ would make the value of $\theta_i^*$ approach 0 if $p_i$ is close to 0. In other words, the baseline regularization forces the model to be less sensitive to features with high variance since they are less correlated to label. Thus, such a model's prediction will not be affected significantly under a shift of the distribution of high variance features during test time.

To study the extent of similarity of this behavior between linear regression and deep networks, we once again conduct experiments with both these models on a finite number of samples drawn randomly from synthetic dataset B with $k = 10$ and $\sigma^2 = 0.001$. The rest of the details regarding dataset generation and models and optimization are identical to what was used in section 2.1.1.

The sensitivity $s_i^*$ vs. $p_i$ plots are shown in figure 2 (left) for linear regression and figure 2 (right) for MLP. In the case of linear regression $s_i^* = \theta_i^*$. For both linear regression and MLP, the model's sensitivity to all features are high irrespective of $p_i$ when trained without the baseline regularization ($\beta = 0$) and this is especially more so for the MLP. On the other hand, when training with the baseline regularization, we find that a larger $\beta$ forces the models to be less sensitive to input feature dimensions with higher variance (which correspond to $p_i = 0$).

## 3 Experiments with Data Distribution Shift

The experiments below are aimed at investigating the ability of existing regularization methods to generalize when the distribution of low correlation features shift compared to the baseline method. Details not mentioned in the main text can be found in appendix B.

**Datasets**: We use a colored version of the MNIST dataset (see appendix A for dataset samples and details) for experiment 1, and MNIST-M (Ganin et al., 2016), SVHN (Netzer et al., 2011), MNIST (LeCun & Cortes, 2010) in addition to C-MNIST for experiment 2. All image pixels lie in 0-1 range and are not normalized. The reason for this is that since we are interested in out of distribution (OOD) classification, the normalization constants of training distribution and OOD may be different, in which case data normalized with different statistics cannot be handled by the same network easily.

**Other Details**: We use ResNet-56 (He et al., 2016b) in all our experiments. We use Adam optimizer (Kingma & Ba, 2014) with batch size 128 and weight decay 0.0001 for all experiments unless specified otherwise. We do not use batch normalization in any experiment except for the adaptive

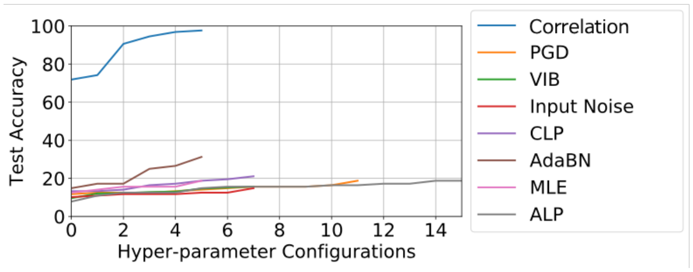

Figure 3: Performance on the distribution shifted test set of C-MNIST for various methods trained on C-MNIST training set. See figure 5 in appendix for samples from C-MNIST dataset.

| Dataset | Accuracy |
|---|---|
| C-MNIST | 96.88 |
| MNIST | 93.75 |
| MNIST-M | 85.94 |
| SVHN | 60.94 |

Table 1: Out of distribution performance on test sets using a model trained with baseline method on C-MNIST dataset.

batch normalization method. Discussion and experiments around batch normalization can be found in appendix C. We do not use any bias parameter in the network because we found it led to less overfitting overall. For all configurations specified for baseline method and existing methods below, the hyper-parameter learning rate was chosen from $\{0.0001, 0.001\}$ unless specified otherwise. For baseline method, the regularization coefficient is chosen from $\{0.1, 1, 10\}$.

**Existing methods**:

1. Vanilla maximum likelihood (MLE) training: Since there are no regularization coefficients in this case, we search over batch sizes from $\{32, 64, 128\}$ for each learning rate value.

2. Variational bottleneck method (VIB, Alemi et al. (2016)) is a method that minimizes the information bottleneck objective and thus acts as a regularization. The regularization coefficient for VIB is chosen from the set $\{0.01, 0.1, 1, 5\}$.

3. Clean logit pairing (CLP): Proposed in Kannan et al. (2018), this method minimizes the $\ell^2$ norm of the difference between the logits of different samples. As shown in proposition 1 (in appendix), minimizing this $\ell^2$ norm is equivalent to minimizing variance of representation in logit space under the assumption that this distribution is Gaussian. In contrast we apply this regularization in the first hidden layer. Due to this similarity, we consider CLP in our experiments. The regularization coefficient for CLP is chosen from $\{0.1, 0.5, 1, 10\}$.

4. Projected gradient descent (PGD) based adversarial training (Madry et al., 2017) has been shown to yield human interpretable features. This makes it a good candidate for investigation. For PGD, $\ell_{\text{inf}}$ perturbation is used with a maximum perturbation $\epsilon$ from the set $\{8, 12, 16, 20\}$ and step size of 2, where all these numbers are divided by 255 since the input is normalized to lie in $[0, 1]$ . The number of PGD steps is chosen from the set $\{20, 50\}$. We randomly choose 12 different configurations out of these combinations.

5. Adversarial logit pairing (ALP, Kannan et al. (2018)) is another approach for adversarial robustness and an alternative to PGD. Since it has the most number of hyper-parameters, we tried a larger number of configurations for this method. Specifically, we use $\ell_{\text{inf}}$ norm with a maximum perturbation $\epsilon$ from the set $\{8, 16, 20\}$ and step size of 2, where all these numbers are divided by 255 since the input is normalized to lie in $[0, 1]$ . The number of PGD steps is chosen from the set $\{20, 50\}$. The regularization coefficient is chosen from $\{0.1, 1, 10\}$. We randomly choose 15 different configurations out of these combinations.

6. Gaussian Input Noise has been shown to have a similar effect as that from adversarial training (Ford et al., 2019) with even better performance in certain cases. We choose Gaussian input noise with standard deviation from the set $\{0.05, 0.1, 0.2, 0.3\}$.

7. Adaptive batch normalization (AdaBN, Li et al. (2016)) has been proposed as a simple way to achieve domain adaptation in which the running statistics of batch normalization are updated with the statistics of the target domain data. Since there are no regularization coefficients in this case, we search over batch sizes from $\{32, 64, 128\}$ for each learning rate value.

**Experiment 1**: In this experiment, we train ResNet-56 on the colored MNIST dataset using the existing methods and baseline regularization, and test the performance of the trained models on the

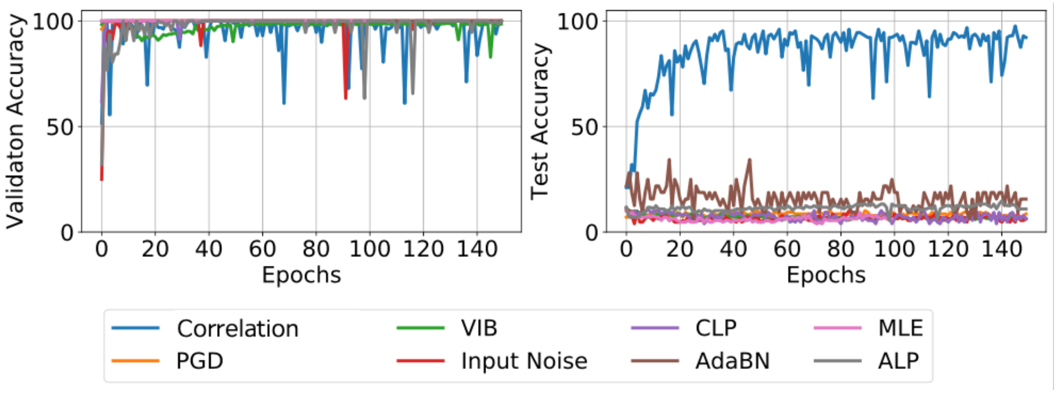

Figure 4: Existing methods severely overfit color features in the C-MNIST training set leading to near $100\%$ accuracy on C-MNIST validation set but close to chance performance on the distribution shifted C-MNIST test set.

distribution shifted test set of colored MNIST dataset in each case. For each method, we record the best test performance for each hyper-parameter configuration used, and after sorting these numbers across all configurations, we plot them in figure 3. We find that all the existing methods tend to severely overfit the non-robust color features in C-MNIST leading to poor performance on the distribution shifted test set of C-MNIST except the baseline method (named correlation).

To further confirm this claim, we plot the validation and test accuracy vs. epoch for all methods for one of the hyper-parameter configurations. The validation set of C-MNIST is a held out set that follows the same distribution as the training set but the two are mutually exclusive. On the other hand, the test set of C-MNIST has different foreground and background colors in all images that are sampled independently of the training set as explained in section A and shown in figure 5. The hyper-parameter configuration chosen (from among the ones used in experiment 1 in section 3) for each method is based on the condition that training accuracy converged to $100\%$ at the end of the training process. This ensures a fair comparison of validation and test accuracy between different methods. The plots can be seen in figure 4. Clearly, existing methods achieve near $100\%$ accuracy on C-MNIST validation set but close to chance performance on the distribution shifted C-MNIST test set, showing that these methods have overfitted the color features. Correlation based learning is able to avoid this dependence.

**Experiments 2**: In this experiment, we hand-pick the model trained with the baseline regularization on C-MNIST in experiment 1 above, such that it simultaneously performs well on SVHN, MNIST-M and MNIST datasets. We used the C-MNIST test set for early stopping. These performances are shown in table 1. These experiments show that the dominant correlation based features learned by the baseline model on C-MNIST dataset capture shape features which allow the model to generalize reasonably well on the other digit datasets with drastic distribution shifts.

## 4 Related Work

**Invariant Risk Minimization**: The goal of IRM (Arjovsky et al., 2019) is to achieve out of distribution generalization under the formalism that certain features have a stable correlation with target across all possible environments. In other words, if there are multiple features that correlate with label, then IRM aims to learn the feature which has the same degree of correlation with label irrespective of the environment, while ignoring other features. IRM achieves this goal by learning representations such that there exists a predictor (Eg. a linear classifier) that is simultaneously optimal for representations across all environments. We instead explore the idea of extracting features that have high correlation with labels.

**Adversarial Training**: There is an abundance of literature around robust optimization (Wald, 1945; Ben-Tal et al., 2009) and adversarial training (Goodfellow et al., 2014; Madry et al., 2017) which study robustness of models to small perturbations around input samples and are often studied using first order methods. Such perturbations can be seen as the worst case distribution shift in the local proximity of the original training distribution. Further, Tsipras et al. (2018) discusses that the representations learned by adversarially trained deep network are more human interpretable. These factors make it a good candidate for investigating its behavior under distribution shift.

**Domain Adaptation**: Domain adaptation (Wang & Deng, 2018; Patel et al., 2014) addresses the problem of distribution shift between source and target domain, and has attracted considerable attention in computer vision, NLP and speech communities (Kulis et al., 2011; Blitzer et al., 2007; Hosseini-Asl et al., 2018). Some of these methods address this issue by aligning the two distributions (Jiang & Zhai, 2007; Bruzzone & Marconcini, 2009), while others by making use of adversarial training (Ganin & Lempitsky, 2014; Ganin et al., 2016) and auxilliary losses (Ghifary et al., 2015; He et al., 2016a). A common characteristic of all these methods is that they require labeled/unlabeled target domain data during the training process.

## 5 Conclusion

We explored the idea of using input feature with high correlation with labels for making prediction such that the distribution of low correlation features shifted during test time. We found that existing methods that are aimed at learning robust representation in the adversarial sense or in the general sense of reducing overfitting are unable to handle such distribution shifts. On the other hand, our regularization specifically designed for this task performed well under this distribution shift during test time.

## Acknowledgments

We would like to thank Bernhard Geiger, Stanisław Jastrzębski, Ehsan Hosseini-Asl, Lav Varshney and Tong Niu for discussions and feedback.

## References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

A. Ben-Tal, L. El Ghaoui, and A.S. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447, 2007.

Lorenzo Bruzzone and Mattia Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):770–787, 2009.

Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018a.

Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 7538–7550, 2018b.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pp. 820–828, 2016a.

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*, pp. 770–778, 2016b.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Ehsan Hosseini-Asl, Yingbo Zhou, Caiming Xiong, and Richard Socher. A multi-discriminator cyclegan for unsupervised non-parallel speech domain adaptation. *arXiv preprint arXiv:1804.00522*, 2018.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pp. 689–696, 2009.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Dominik Janzing, Patrik O Hoyer, and Bernhard Schölkopf. Telling cause from effect based on high-dimensional observations. *arXiv preprint arXiv:0909.4386*, 2009.

Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 264–271, 2007.

Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR 2011*, pp. 1785–1792. IEEE, 2011.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6979–6987, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

David A McAllester. Pac-bayesian model averaging. In *COLT*, volume 99, pp. 164–170. Citeseer, 1999a.

David A McAllester. Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363, 1999b.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: An overview of recent advances. *IEEE Signal Processing Magazine*, 2014.

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

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Abraham Wald. Statistical decision functions which minimize the maximum risk. *Annals of Mathematics*, pp. 265–280, 1945.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

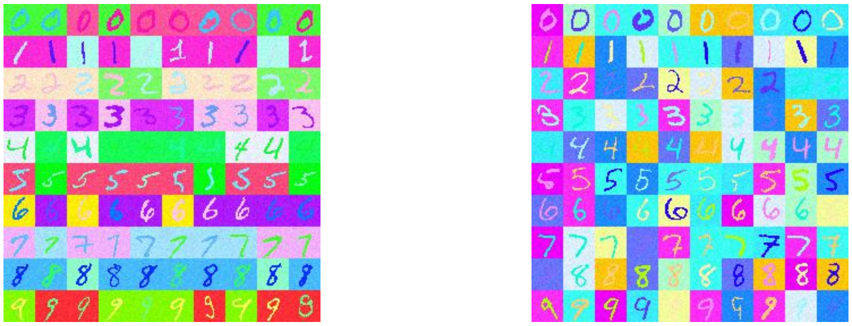

Figure 5: Color MNIST training set (left) and out of distribution test set (right). Each class in training set has two background colors and two foreground colors that are unique only to that class. Each test image has a foreground and background color that is randomly picked out of 10 colors that are chosen independently of the training set.

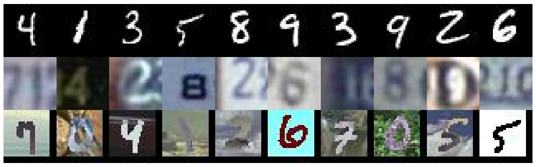

Figure 6: Random samples from MNIST (top), SVHN (middle) and MNIST-M (bottom) datasets are shown to get a visual sense of the hardness of the out of distribution task.

# Appendix

## A Datasets

**Colored MNIST Dataset**: Randomly drawn training and test samples from the C-MNIST dataset generated as described above are shown in figure 5. The colored MNIST dataset (C-MNIST), which is used in experiments 1 and 2, uses all the 60,000 training image in the MNIST dataset to generate the C-MNIST dataset, where we randomly do a 0.9-0.1 split to get the training set and validation set respectively. Similarly, we use all the 10,000 test images in MNIST to generate the C-MNIST test set. We vary both the foreground and background colors to generate our C-MNIST dataset. The reason why we vary colors of both foreground and background is that we want the trained model to avoid overfitting any color bias. A single color in foreground or background would constitute a low variance feature, which as we study in section 2.1.2, leads models to prioritize learning it for both training with vanilla MLE as well as MLE with baseline method.

The C-MNIST training/validation set is generated from its MNIST counter-part as follows:

1. For each class, randomly assign two colors (RGB value) for foreground and two colors (RGB value) for background.

2. Binarize each image (pixels in 0-255 range) around threshold 150 so that pixel values are either 0 or 1 and replicate the channel in each image to have a three channel image.

3. For each image in a class, randomly pick one of the two foreground colors assigned to that class and replace all foreground pixel with that color. Similarly replace background pixels for all images.

4. Add zero mean Gaussian noise with a small standard deviation (0.04 used in our experiments) to all images.

To generate the test set, in step one, we randomly assign a foreground and background color to each image irrespective of the class, and the colors for validation set are chosen independently of the training set.

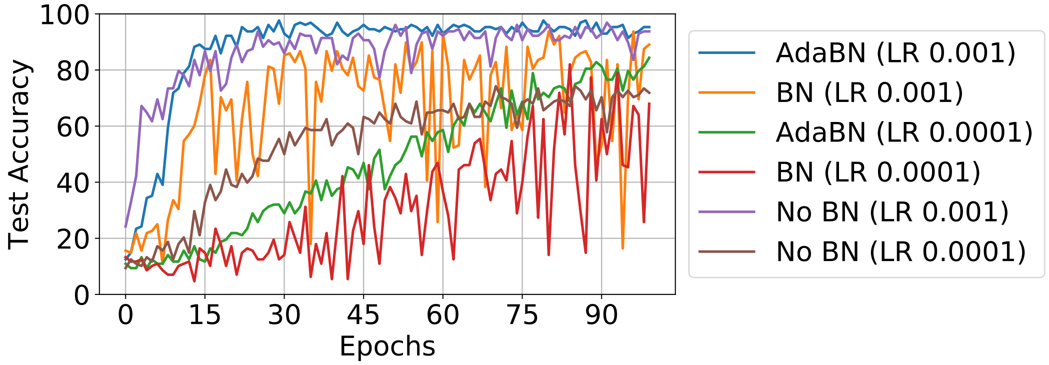

Figure 7: Batch Normalization is unstable without AdaBN when using proposed regularization (experiment on C-MNIST dataset).

**Other Datasets**: During experiments with MNIST, the single channel in each image was replicated to form a three channel image. For SVHN, we resized the image to have 28 × 28 hieght and width.

## B Experimental Details

**Variational bottleneck method** (VIB): As an implementation detail of VIB, we flatten the output of the last convolution layer of ResNet-56 and separately pass it through two linear layer of width 256, one that outputs a vector that we regard as mean $\mu$, and the other that we regard as log-variance $\nu$ (similar to how it is done in variational auto-encoders (Kingma & Welling, 2013)). We then combine their outputs as $\mathbf{o} = \mu + exp(0.5\nu) \odot \epsilon$ where $\epsilon$ is sampled from a standard Gaussian distribution of the same dimension as $\nu$. This output is then passed through a linear layer that transforms it into a vector of dimension same as the number of classes. These implementation details are similar to those in the original VIB paper.

Further, we gradually ramp up the regularization coefficient $\beta$ from 0.0001 to its final value by doubling the current value at the end of every epoch until the final value is reached. This is a popular prctice when minimizing the KL divergence term between posterior and prior which helps optimization.

## C Effects of Batch Normalization on Baseline Method

In this section we study the effect of batch normalization (BN, Ioffe & Szegedy (2015)) when using it in conjunction with baseline method. We train a ResNet-56 on C-MNIST dataset identical to the settings in experiment 1 in main text, with the exception that we use BN. While doing so, we record the accuracy of the model on the C-MNIST test set at every epoch. In addition, we also run experiments where under the same training settings, during evaluation at each epoch, we use AdaBN (Li et al., 2016). These values are plotted in figure 7. We have also plotted runs using baseline method without any batch normalization for reference. In all cases, we use a learning rates (LR) of 0.001 and 0.0001. The plots show that without AdaBN, test accuracy is very unstable on the distribution shifted test set. This seems to be a side-effect of using BN which can be fixed by adapting the running statistics of BN (used during evaluation) with the test set statistics. However, this requires domain knowledge (i.e., samples) of the test set, which is not preferable for our goal.

## D Proofs

**Lemma 1**

$$\mathbb{E}[x_i|y=1] = -\mathbb{E}[x_i|y=-1] = 2p_i - 1 \tag{4}$$

$$\mathbb{E}[x_i^2|y=1] = \mathbb{E}[x_i^2|y=-1] = 1 + \sigma^2 \tag{5}$$

***Proof**: Given the distribution of* $\mathbf{x}$*, we can write each element* $x_i$ *as,*

$$x_i = b_i n_i + (1 - b_i)\bar{n}_i \tag{6}$$

*where* $b_i$ *is sampled from the Bernoilli distribution with probability* $p_i$*,* $n_i \sim \mathcal{N}(y, \sigma^2)$*, and* $\bar{n}_i \sim \mathcal{N}(-y, \sigma^2)$*. Thus,*

$$\mathbb{E}[x_i|y = 1] = p_i - (1 - p_i) = 2p_i - 1 \tag{7}$$

$$\mathbb{E}[x_i|y = -1] = -p_i + (1 - p_i) = 1 - 2p_i \tag{8}$$

*Next,*

$$\mathbb{E}[x_i^2|y = 1] = \mathbb{E}[b_i^2 n_i^2 + (1 - b_i)^2 \bar{n}_i^2 + 2b_i(1 - b_i)n_i\bar{n}_i|y = 1] \tag{9}$$

*We know from the properties of Bernoulli and Gaussian distribution that,*

$$\mathbb{E}[b_i^2] = var(b_i) + \mathbb{E}[b_i]^2 = p_i(1 - p_i) + p_i^2 = p_i \tag{10}$$

$$\mathbb{E}[n_i^2|y = 1] = var(n_i|y = 1) + \mathbb{E}[n_i|y = 1]^2 = \sigma^2 + 1 \tag{11}$$

*We similarly get,*

$$\mathbb{E}[n_i^2|y = -1] = \sigma^2 + 1 \tag{12}$$

$$\mathbb{E}[\bar{n}_i^2|y = 1] = \sigma^2 + 1 \tag{13}$$

$$\mathbb{E}[\bar{n}_i^2|y = -1] = \sigma^2 + 1 \tag{14}$$

*Therefore, using the independence property between random variables,*

$$\begin{aligned}\mathbb{E}[x_i^2|y = 1] &= \mathbb{E}[b_i^2]\mathbb{E}[n_i^2|y = 1] + (1 + \mathbb{E}[b_i^2] - 2\mathbb{E}[b_i])\mathbb{E}[\bar{n}_i^2|y = 1] \\ &+ 2(\mathbb{E}[b_i] - \mathbb{E}[b_i]^2)\mathbb{E}[\bar{n}_i|y = 1]\mathbb{E}[n_i|y = 1] \qquad (15) \\ &= 1 + \sigma^2 \qquad (16)\end{aligned}$$

*We similarly have,*

$$\mathbb{E}[x_i^2|y = -1] = 1 + \sigma^2 \tag{17}$$

**Theorem 3** *Let* $\theta^*$ *be the minimizer of* $J(\theta)$ *in Eq. 1 where we have used synthetic dataset A. Then for a large enough* $d$*,* $\theta^*$ *is given by,*

$$\theta^* = \mathbf{M}^{-1}|2\mathbf{p} - \mathbf{1}| \tag{18}$$

*where,*

$$\mathbf{M} := \mathbf{\Sigma} + \lambda\mathbf{I} + \beta(\sigma^2\mathbf{I} + 4diag(\mathbf{p} \odot (\mathbf{1} - \mathbf{p}))) \tag{19}$$

*such that* $\mathbf{\Sigma}$ *is a positive definite matrix if*[2] $p_i \notin \{0, 0.5, 1\}$ *for all* $i$*.*

***Proof**: We note that,*

$$\begin{aligned}f_\theta(\mathbf{x})|(y = 1) &= \sum_{i=1}^{d} \theta_i x_i|y = 1 \qquad (20) \\ &= \sum_{i=1}^{d} \theta_i(b_i(n_i|y = 1) + (1 - b_i)(\bar{n}_i|y = 1)) \qquad (21)\end{aligned}$$

*For a large enough dimensionality* $d$ *of* $\mathbf{x}$*, central limit theorem (CLT) applies to* $f_\theta(\mathbf{x})|(y = 1)$ *and it converges to a Gaussian distribution. Let* $s_1^2$ *denote the variance of* $f_\theta(\mathbf{x})|y = 1$*. A similar argument applies to* $f_\theta(\mathbf{x})|y = -1$*, in which case we define* $s_{-1}^2$ *to be its variance. Thus,*

$$\begin{aligned}s_1^2 &= var(\sum_{i=1}^{d} \theta_i x_i|y = 1) \qquad (22) \\ &= \sum_{i=1}^{d} var(\theta_i x_i|y = 1) \qquad (23) \\ &= \sum_{i=1}^{d} \theta_i^2 var(x_i|y = 1) \qquad (24)\end{aligned}$$

[2] This assumption is needed due to technicality.

*where the second equality holds because $x_i$'s are independent of one another. Thus using lemma 1,*

$$s_1^2 = \sum_i \theta_i^2(1+\sigma^2-(2p_i-1)^2) \tag{25}$$

$$= \sum_i \theta_i^2(\sigma^2+4p_i(1-p_i)) \tag{26}$$

*We similarly get,*

$$s_{-1}^2 = \sum_i \theta_i^2(\sigma^2+4p_i(1-p_i)) \tag{27}$$

*Since $s_1^2$ and $s_{-1}^2$ are equal, we denote $s^2 = s_1^2 = s_{-1}^2$. Therefore, our objective becomes,*

$$\arg\min_\theta \mathbb{E}[(f_\theta(\mathbf{x})-y)^2] + \lambda\|\theta\|^2 + \beta s^2 \tag{28}$$

$$= \arg\min_\theta \theta^T\mathbb{E}[\mathbf{x}\mathbf{x}^T]\theta - 2\theta^T\mathbb{E}[\mathbf{x}y] + \lambda\|\theta\|^2 + \beta\sum_{i=1}^d \theta_i^2(\sigma^2+4p_i(1-p_i)) \tag{29}$$

*Define $\mathbf{M}$ as,*

$$\mathbf{M} := \mathbf{\Sigma} + \lambda\mathbf{I} + \beta(\sigma^2\mathbf{I} + 4diag(\mathbf{p}\odot(\mathbf{1}-\mathbf{p}))) \tag{30}$$

*where $\mathbf{\Sigma} := \mathbb{E}[\mathbf{x}\mathbf{x}^T]$, we can re-write our objective as,*

$$\arg\min_\theta \theta^T\mathbf{M}\theta - 2\theta^T\mathbb{E}[\mathbf{x}y] \tag{31}$$

*whose solution is given by,*

$$\theta^* = \mathbf{M}^{-1}\mathbb{E}[\mathbf{x}y] \tag{32}$$

*Using lemma 1, we get,*

$$\mathbb{E}[x_iy] = \mathbb{E}[x_i|y=1]\Pr(y=1) - \mathbb{E}[x_i|y=-1]\Pr(y=-1) \tag{33}$$

$$= 0.5(\mathbb{E}[x_i|y=1] - \mathbb{E}[x_i|y=-1]) \tag{34}$$

$$= 0.5(4p_i-2) = 2p_i - 1 \tag{35}$$

*Plugging this value in Eq. 32 yields $\theta^*$.*

*Now we prove that $\mathbf{\Sigma}$ is full rank. Note that it is positive semi-definite since it is a scatter matrix. Next, due to conditional independence, for i $\neq$ j,*

$$\mathbb{E}[x_ix_j] = \mathbb{E}[x_ix_j|y=1]\Pr(y=1) + \mathbb{E}[x_ix_j|y=-1]\Pr(y=-1) \tag{36}$$

$$= \mathbb{E}[x_i|y=1]\mathbb{E}[x_j|y=1]\Pr(y=1) + \mathbb{E}[x_i|y=-1]\mathbb{E}[x_j|y=-1]\Pr(y=-1) \tag{37}$$

*and for i = j,*

$$\mathbb{E}[x_i^2] = \mathbb{E}[x_i^2|y=1]\Pr(y=1) + \mathbb{E}[x_i^2|y=-1]\Pr(y=-1) \tag{38}$$

*Using lemma 1, we get,*

$$\Sigma_{ij} = \begin{cases} 1+\sigma^2 & \textit{if } i=j \\ (1-2p_i)(1-2p_j) & \textit{otherwise} \end{cases} \tag{39}$$

*To prove that $\mathbf{\Sigma}$ is positive definite (and hence full rank), we need to prove that no two columns are parallel. To show this, consider any two indices $i$ and $j$ such that $i \neq j$. We show that there exists no $\alpha \neq 0$ such that the columns $\Sigma_i = \alpha\Sigma_j$. We prove this by contradiction. Suppose $\Sigma_{ii} = \alpha\Sigma_{ji}$, then $\alpha\Sigma_{jj} = \frac{\Sigma_{ii}\Sigma_{jj}}{\Sigma_{ji}}$. Substituting values from Eq. 39,*

$$\alpha\Sigma_{jj} = \frac{(1+\sigma^2)^2}{(1-2p_i)(1-2p_j)} \tag{40}$$

*Thus $\alpha\Sigma_{jj} > 1$. However, $\Sigma_{ij} = (1-2p_i)(1-2p_j) < 1$. Thus there is no non-zero $\alpha$ for which $\Sigma_i = \alpha\Sigma_j$. Hence $\mathbf{\Sigma}$ must be full rank and hence positive definite. Thus we have proved the claim.* □

**Lemma 2**

$$\mathbb{E}[x_i|y=1] = -\mathbb{E}[x_i|y=-1] = 1 \tag{41}$$

$$\mathbb{E}[x_i^2|y=1] = \mathbb{E}[x_i^2|y=-1] = 1 + \sigma^2(p_i + k_i(1-p_i)) \tag{42}$$

***Proof**: Given the distribution of* $\mathbf{x}$*, we can write each element* $x_i$ *as,*

$$x_i = b_i n_i + (1-b_i)\bar{n}_i \tag{43}$$

*where* $b_i$ *is sampled from the Bernoilli distribution with probability* $p_i$*,* $n_i \sim \mathcal{N}(y, \sigma^2)$*, and* $\bar{n}_i \sim \mathcal{N}(y, k_i\sigma^2)$*. Thus,*

$$\mathbb{E}[x_i|y=1] = p_i + (1-p_i) = 1 \tag{44}$$

$$\mathbb{E}[x_i|y=-1] = -p_i - (1-p_i) = -1 \tag{45}$$

*Next,*

$$\mathbb{E}[x_i^2|y=1] = \mathbb{E}[b_i^2 n_i^2 + (1-b_i)^2\bar{n}_i^2 + 2b_i(1-b_i)n_i\bar{n}_i|y=1] \tag{46}$$

*We know from the properties of Bernoulli and Gaussian distribution that,*

$$\mathbb{E}[b_i^2] = var(b_i) + \mathbb{E}[b_i]^2 = p_i(1-p_i) + p_i^2 = p_i \tag{47}$$

$$\mathbb{E}[n_i^2|y=1] = var(n_i|y=1) + \mathbb{E}[n_i|y=1]^2 = \sigma^2 + 1 \tag{48}$$

*We similarly get,*

$$\mathbb{E}[n_i^2|y=-1] = \sigma^2 + 1 \tag{49}$$

$$\mathbb{E}[\bar{n}_i^2|y=1] = k\sigma^2 + 1 \tag{50}$$

$$\mathbb{E}[\bar{n}_i^2|y=-1] = k\sigma^2 + 1 \tag{51}$$

*Therefore, using the independence property between random variables,*

$$\begin{aligned}\mathbb{E}[x_i^2|y=1] &= \mathbb{E}[b_i^2]\mathbb{E}[n_i^2|y=1] + (1 + \mathbb{E}[b_i^2] - 2\mathbb{E}[b_i])\mathbb{E}[\bar{n}_i^2|y=1] \\ &\quad + 2(\mathbb{E}[b_i] - \mathbb{E}[b_i]^2)\mathbb{E}[\bar{n}_i|y=1]\mathbb{E}[n_i|y=1] && (52) \\ &= p_i(1+\sigma^2) + (1-p_i)(1+k\sigma^2) && (53)\end{aligned}$$

*We similarly have,*

$$\mathbb{E}[x_i^2|y=-1] = p_i(1+\sigma^2) + (1-p_i)(1+k\sigma^2) \tag{54}$$

*Rearranging these terms yields the claim.* □

**Theorem 4** *Let* $\theta^*$ *be the minimizer of* $J(\theta)$ *in Eq. 1 where we have used synthetic dataset B. Then for a large enough* $d$*,* $\theta^*$ *is given by,*

$$\theta^* = \mathbf{M}^{-1}\mathbf{1} \tag{55}$$

*where,*

$$\mathbf{M} := \mathbf{\Sigma} + \lambda\mathbf{I} + \beta\sigma^2 diag(\mathbf{p} + k(\mathbf{1} - \mathbf{p})) \tag{56}$$

*such that* $\mathbf{\Sigma}$ *is a positive definite matrix.*

***Proof**: Similar to theorem 1 we have that,*

$$f_\theta(\mathbf{x})|(y=1) = \sum_{i=1}^{d} \theta_i b_i(n_i|y=1) + (1-b_i)(\bar{n}_i|y=1) \tag{57}$$

*For a large enough dimensionality* $d$ *of* $\mathbf{x}$*, central limit theorem (CLT) applies to* $f_\theta(\mathbf{x})|(y=1)$ *and it converges to a Gaussian distribution. Let* $s_1^2$ *denote the variance of* $f_\theta(\mathbf{x})|y=1$*. A similar argument applies to* $f_\theta(\mathbf{x})|y=-1$*, in which case we define* $s_{-1}^2$ *to be its variance. Thus,*

$$s_1^2 = \sum_{i=1}^{d} \theta_i^2 var(x_i|y=1) \tag{58}$$

*Thus using lemma 2,*

$$s_1^2 = \sum_i \theta_i^2(\sigma^2(p_i + k_i(1-p_i))) \tag{59}$$

*We similarly get,*

$$s_{-1}^2 = \sum_i \theta_i^2(\sigma^2(p_i + k_i(1-p_i))) \tag{60}$$

*Since $s_1^2$ and $s_{-1}^2$ are equal, we denote $s^2 = s_1^2 = s_{-1}^2$. Therefore, our objective becomes,*

$$\arg\min_\theta \mathbb{E}[(f_\theta(\mathbf{x}) - y)^2] + \lambda\|\theta\|^2 + \beta s^2 \tag{61}$$

$$= \arg\min_\theta \theta^T \mathbb{E}[\mathbf{x}\mathbf{x}^T]\theta - 2\theta^T\mathbb{E}[\mathbf{x}y] + \lambda\|\theta\|^2 + \beta\sigma^2 \sum_i \theta_i^2(p_i + k_i(1-p_i)) \tag{62}$$

*Define $\mathbf{M}$ as,*

$$\mathbf{M} := \mathbf{\Sigma} + \lambda\mathbf{I} + \beta\sigma^2 diag(\mathbf{p} + k(\mathbf{1} - \mathbf{p})) \tag{63}$$

*where $\mathbf{\Sigma} := \mathbb{E}[\mathbf{x}\mathbf{x}^T]$, we can re-write our objective as,*

$$\arg\min_\theta \theta^T\mathbf{M}\theta - 2\theta^T\mathbb{E}[\mathbf{x}y] \tag{64}$$

*whose solution is given by,*

$$\theta^* = \mathbf{M}^{-1}\mathbb{E}[\mathbf{x}y] \tag{65}$$

*Using lemma 2, we get,*

$$\mathbb{E}[x_i y] = \mathbb{E}[x_i|y=1]\Pr(y=1) - \mathbb{E}[x_i|y=-1]\Pr(y=-1) \tag{66}$$

$$= 0.5(\mathbb{E}[x_i|y=1] - \mathbb{E}[x_i|y=-1]) \tag{67}$$

$$= 1 \tag{68}$$

*Plugging this value in Eq. 65 yields $\theta^*$.*

*Now we prove that $\mathbf{\Sigma}$ is full rank. Note that it is positive semi-definite since it is a scatter matrix. Next, due to conditional independence, for $i \neq j$,*

$$\mathbb{E}[x_i x_j] = \mathbb{E}[x_i x_j|y=1]\Pr(y=1) + \mathbb{E}[x_i x_j|y=-1]\Pr(y=-1) \tag{69}$$

$$= \mathbb{E}[x_i|y=1]\mathbb{E}[x_j|y=1]\Pr(y=1) + \mathbb{E}[x_i|y=-1]\mathbb{E}[x_j|y=-1]\Pr(y=-1) \tag{70}$$

*and for $i = j$,*

$$\mathbb{E}[x_i^2] = \mathbb{E}[x_i^2|y=1]\Pr(y=1) + \mathbb{E}[x_i^2|y=-1]\Pr(y=-1) \tag{71}$$

*Using lemma 2, we get,*

$$\Sigma_{ij} = \begin{cases} 1 + \sigma^2(p_i + k_i(1-p_i)) & \textit{if } i = j \\ 1 & \textit{otherwise} \end{cases} \tag{72}$$

*To prove that $\mathbf{\Sigma}$ is positive definite (and hence full rank), we need to prove that no two columns are parallel. To show this, consider any two indices $i$ and $j$ such that $i \neq j$. We show that there exists no $\alpha \neq 0$ such that the columns $\Sigma_i = \alpha\Sigma_j$. We prove this by contradiction. Suppose $\Sigma_{ii} = \alpha\Sigma_{ji}$, then $\alpha\Sigma_{jj} = \frac{\Sigma_{ii}\Sigma_{jj}}{\Sigma_{ji}}$. Substituting values from Eq. 72,*

$$\alpha\Sigma_{jj} = (1 + \sigma^2(p_i + k(1-p_i)))(1 + \sigma^2(p_j + k(1-p_j))) \tag{73}$$

*Thus $\alpha\Sigma_{jj} > 1$. However, $\Sigma_{ij} = 1$. Thus there is no non-zero $\alpha$ for which $\Sigma_i = \alpha\Sigma_j$. Hence $\mathbf{\Sigma}$ must be full rank and hence positive definite. Thus we have proved the claim.* □

**Proposition 1** *If $\Pr(f_\theta(\mathbf{X}))$ follows a Gaussian distribution, then,*

$$var(f_\theta(\mathbf{X})) = 0.5\mathbb{E}_{\mathbf{X}_1,\mathbf{X}_2\sim\mathcal{D}(\mathbf{X})}[(f_\theta(\mathbf{X}_1) - f_\theta(\mathbf{X}_2))^2] \tag{74}$$

*where* $\mathbf{X}_1$ *and* $\mathbf{X}_2$ *are IID samples from the data distribution* $\mathcal{D}(\mathbf{X})$*.*

***Proof****: Denote* $\mu$ *and* $\sigma^2$ *as the mean and variance of the Gaussian distribution* $\Pr(f_\theta(\mathbf{X}))$ *respectively. Since* $\mathbf{X}_1$ *and* $\mathbf{X}_2$ *are IID samples, we have that,*

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X}_1,\mathbf{X}_2\sim\mathcal{D}(\mathbf{X})}[(f_\theta(\mathbf{X}_1) - f_\theta(\mathbf{X}_2))^2] && (75)\\
&= \mathbb{E}_{\mathbf{X}_1,\mathbf{X}_2\sim\mathcal{D}(\mathbf{X})}[((f_\theta(\mathbf{X}_1) - \mu) - (f_\theta(\mathbf{X}_2) - \mu))^2] && (76)\\
&= \mathbb{E}_{\mathbf{X}_1\sim\mathcal{D}(\mathbf{X})}[(f_\theta(\mathbf{X}_1) - \mu)^2] + \mathbb{E}_{\mathbf{X}_2\sim\mathcal{D}(\mathbf{X})}[(f_\theta(\mathbf{X}_2) - \mu)^2]\\
&- 2\mathbb{E}_{\mathbf{X}_1,\mathbf{X}_2\sim\mathcal{D}(\mathbf{X})}[(f_\theta(\mathbf{X}_1) - \mu)(f_\theta(\mathbf{X}_2) - \mu)] && (77)\\
&= 2\mathbb{E}_{\mathbf{X}\sim\mathcal{D}(\mathbf{X})}[(f_\theta(\mathbf{X}) - \mu)^2] && (78)\\
&= 2\sigma^2 && (79)
\end{aligned}$$

*which proves yields the claim.* □