Beyond Debiasing: Actively Steering Feature Selection via Loss Regularization

Jan Blunk[1] [0009−0001−7037−3545], Niklas Penzel[1] [0000−0001−8002−4130], Paul Bodesheim[1] [0000−0002−3564−6528], and Joachim Denzler[1] [0000−0002−3193−3300]

Computer Vision Group, Friedrich Schiller University Jena, 07743 Jena, Germany
{jan.blunk,niklas.penzel,paul.bodesheim,joachim.denzler}@uni-jena.de
https://inf-cv.uni-jena.de

Abstract. It is common for domain experts like physicians in medical studies to examine features for their reliability with respect to a specific domain task. When introducing machine learning, a common expectation is that machine learning models use the same features as these human experts to solve a task, but that is not always the case. Moreover, datasets often contain features that are known from domain knowledge to generalize badly to the real world, referred to as biases. Current debiasing methods only remove such influences. To additionally integrate the domain knowledge about well-established features into the training of a model, their relevance should be increased. We present a method that allows the manipulation of the relevance of features by actively steering the model's feature selection during the training process. That is, it allows both the discouragement of biases and encouragement of well-established features to incorporate domain knowledge about the feature reliability. We model our objectives for actively steering the feature selection process as a constrained optimization problem, which we implement via a loss regularization that is based on batch-wise feature attributions. We evaluate our approach on a novel synthetic regression dataset and a dataset from the computer vision domain. We observe that it successfully steers the features a model selects during the training process. This is a strong indicator that our method can be used to integrate domain knowledge about well-established features into a model.

Keywords: Feature Steering · Domain Knowledge Integration · Feature Relevance · Trustworthy AI.

1 Introduction

Being able to explicitly manipulate how models utilize features to derive their predictions enables diverse opportunities. In particular, it can be used to improve generalization and interpretability [21] of model predictions.

One motivation for the active interference in a model's feature selection process is a training distribution that contains biases. Biases are a common problem in computer vision [46]. The term describes features that are only spuriously correlated with the label in the training distribution. If a model bases its predictions on such a bias, it fails to generalize to the real world. If it is known from domain knowledge which features constitute a bias, it is desirable to reduce their influence on the model's prediction process.

In addition to discouraging biases, it can also be desirable to encourage the influence of features on the model's prediction process if they are known from domain knowledge to be particularly well-established. This is not only expected to improve generalization but, similarly to debiasing, it could also increase trust in the model's decisions. An example of such domain knowledge are results from medical studies like the ABCD rule introduced by Nachbar and Stolz [31] for the identification of malignant melanoma. Because State-of-the-art models only base their predictions on some but not all of the features proposed by this rule [39], it provides a suitable illustration of the need for active steering of feature selection.

We show that it is not only possible to discourage but also to encourage the influence of features on a model's prediction process. We present a method that actively steers the influence of features via loss regularization during the training process. Our feature steering approach models the desire for correct predictions and intervention in the model's feature selection process as a multi-objective optimization problem and solves it via the weighted sum method [24]. We evaluate our method on a small regression problem to which we add redundancy and the more complex Colored MNIST dataset [2].

2 Related Work

Most prior work that relates to feature steering has been designed for debiasing [42,22,41,38]. Debiasing describes a special case of feature steering that is limited to discouraging a model from basing its predictions on features that are known to generalize badly to the real world. This situation occurs if a feature is spuriously correlated to the label in the training distribution indicating a causal link between the feature and label, even though this causal link does not exist in the test distribution [18]. We refer to these features as biases. Debiasing covers a diverse set of methods [26,33]. Not all approaches to debiasing are related to our feature steering since their design can be very problem-specific.

Both debiasing and our feature steering method belong to explanation-guided learning (EGL). Gao et al. [13] presented an extensive survey including a theoretical definition for this concept. While explainable AI (XAI) [1,4,17] attempts to generate explanations for model predictions, it does not examine how to improve a model's behavior based on these explanations. EGL attempts to integrate the acquired knowledge by simultaneously optimizing for both generalization and the desired properties of the explanations. In our case, the explanations are batch-wise feature attributions, which we attempt to align with the feature steering objectives.

Conceptually, our method shares similarities with an approach proposed by Erion et al. [11]. The authors also proposed a general framework to align feature attributions with domain knowledge. Contrary to our method, they considered the relative importance between features. Moreover, the authors focussed on the integration of domain knowledge about the higher-level properties of the relationships between features to increase performance. We, however, are interested in priors on the selection of features motivated by interpretability and the improvement of generalization.

Implementation-wise, we perform loss regularization to align the feature attributions with domain knowledge. Ross et al. [42] introduced a loss-based regularization approach to EGL. They added a penalty term to the original loss function that integrates domain knowledge via feature attributions generated with input gradients [3]. Rieger et al. proposed contextual decomposition explanation penalization (CDEP) [41], which similarly utilizes penalty terms to perform debiasing. The authors generated their feature attributions with contextual decomposition [30,44], which enabled them to integrate priors about the interaction of features. Reimers et al. also performed debiasing via loss-based regularization [38]. Contrary to the other two approaches, their model-agnostic feature attribution method allowed them to apply their method to features that cannot explicitly be modeled as part of the model inputs. Debiasing via loss-based regularization is also applicable to natural language processing as has been shown by Liu and Avci [22]. We differ from the aforementioned methods with respect to the regularization objective: We are not only interested in decreasing the influence of biases but also in increasing the influence of well-established features.

3 Method

Goal. Our goal is to perform feature steering, that is, to manipulate the influence of specific features on the prediction process of a model during the training process. We want to (1) discourage the usage of undesired features like biases and (2) encourage the usage of desired features identified from domain knowledge. Toward these goals, we implement this via a penalty term that is added to the original loss function.

3.1 Feature Steering

We implement feature steering building upon existing regularization concepts [15, p. 117]. Out of the many different regularization techniques [29], we focus on the modification of the loss function. Specifically, we use penalty terms to incorporate constrained optimization, which allows us to explicitly alter the targeted optimum.

Feature steering can be defined as a constrained optimization problem: On the one hand, we want the model to generate correct predictions via a maximum-likelihood estimation of its parameters and on the other hand we are interested in decreasing or increasing the relevance of certain features. Related debiasing works applied Lagrange multipliers [7] to combine the two objectives of this constraint optimization problem into a single loss function [42,41,38,22]. We generalize this from debiasing to the discouragement and encouragement of arbitrary features, where D refers to the set of features that should be discouraged and E to the set of features that should be encouraged. To model the resulting multi-objective optimization problem, we apply the method of weighted sums [24]. With c_i being a measure of the influence of feature i on the model's prediction process, $\lambda \in \mathbb{R}_{\geq 0}$ as a weight factor and $\mathcal{L}$ as the standard maximum-likelihood loss for network parameters θ, the loss function for general feature steering is defined as:

$$\mathcal{L}'(\theta) = \mathcal{L}(\theta) + \lambda \left(\sum_{i \in D} ||c_i|| - \sum_{i \in E} ||c_i|| \right). \tag{1}$$

For $|| \cdot ||$, we consider the L1 and L2 norms.

3.2 Feature Attribution

To calculate our loss function in practice, the influence c_i of the features whose influence should be steered needs to be determined in every step of the training. The process of determining the influence of specific features is referred to as feature attribution [22].

Contextual decomposition (CD) [30,44] determines the influence of a specific feature by decomposing the output of the model into a linear combination of the influence of the feature and the influence of all other features. This decomposition is iteratively computed from a decomposition of the input within a single forward pass. Therefore, contextual decomposition is designed to determine the influence of features that can be represented as a subset of the inputs of the model.

Reimers et al. [38] model the process of supervised learning via a structural causal model (SCM) [34]. Using this SCM and Reichenbach's Common Cause Principle [37], they derive that the binary question of whether or not a feature influences the prediction of a model boils down to a simple conditional independence test [40]. A feature X_i influences the prediction of a model if it is statistically dependent on the corresponding prediction $\hat{Y}_i$ of the model given the ground truth Y_i:

$$X_i \not\!\perp\!\!\!\perp \hat{Y}_i \mid Y_i. \tag{2}$$

The authors extend this to a quantitative measure of feature attribution by considering the test statistic of independence tests. We follow their approach and use the conditional mutual information (CMI) [25] and an extended version of the Hilbert Schmidt independence criterion (conditional HSIC) [38,16] to determine the influence of features.

Implementation Details for Reimers et al. The feature attribution obtained with CMI as $I(X_i; \hat{Y}_i|Y_i)$ can take infinite values, which would lead to infinitely large weight changes. This problem occurs if the model's prediction can be fully described by X_i and Y_i since the CMI describes how much additional knowledge of $x \in X_i$ reduces the uncertainty about $\hat{Y}_i$ when $y \in Y_i$ is already known [25, Section 8.1]. Therefore, we transform the CMI into a finite interval with a transformation t that is based on a similar transformation proposed by Linfoot [20] for mutual information. For the CMI $I_i = I(X_i; \hat{Y}_i|Y)$ with respect to feature i we define t as:

$$t(I_i) = \sqrt{1 - e^{-2 \cdot I_i}}. \tag{3}$$

We estimate the CMI with an estimator proposed by Zan et al. [48]. Even though the CMI can generally be considered strictly positive, these estimates may be negative [36]. Due to the definition of the square root, t cannot be applied to these negative estimates. To avoid saturation, we do not set t to a fixed value for negative estimates but instead proceed similarly to straight-through estimators [47] and apply an identity transformation:

$$c_i = \begin{cases} t(I_i) & \text{for } I_i > 0 \\ I_i & \text{for } I_i \leq 0. \end{cases} \tag{4}$$

Consequently, we also apply the identity transformation instead of the L1 or L2 norm to the resulting negative feature attributions. Otherwise, a negative feature attribution would result in a larger loss than positive feature attributions with a smaller magnitude, even though features with negative feature attributions have less influence on the model's prediction process than features with positive feature attributions.

3.3 Theoretical Considerations

Our loss function consists of two components modeling the two separate objectives of feature steering: The maximum-likelihood loss seeks correct predictions for the original distribution while the feature steering part implements the manipulation of the feature influence.

For both very small and very large values of the weight factor λ, one of the two components dominates the loss. In the case of very small values of λ the maximum-likelihood loss dominates and we expect no feature steering to be performed. For very large values of λ, the feature steering component dominates and we expect the model to disregard the desire for correct predictions potentially leading to pathological solutions. Since we are interested in both objectives, we have to select λ as a tradeoff between these two extremes.

4 Datasets

We test our feature steering approach on two datasets of different complexity. Our evaluation starts with a small regression-based example and is then extended to feature steering in an image classification setting.

4.1 Redundant Regression Dataset

We first examine the fundamental behavior of our method on a small regression dataset, to which we add redundant information. That is, we create a low-dimensional linear regression problem and perform a dimensionality expansion of the input variables to add redundant information.

Low-Dimensional Regression Dataset. Our regression-based dataset is generated from a linear regression problem with standard-normal distributed random variables $X_0, ..., X_5$ and standard-uniformly distributed regression coefficients $\beta_0, ..., \beta_5$. That is, the target variable Y is calculated as:

$$Y = \sum_{i=0,...,5} \beta_i X_i. \tag{5}$$

Redundancy. We want to evaluate our method on a dataset that has redundant features because we motivate feature steering as encouraging a model to select particularly favorable features out of multiple alternatives for prediction. To generate redundancy between features in our regression problem, we borrow from latent factor analysis (FA) [5,6]. Concretely, we consider the input variables of our regression problem as unobserved "latent variables" from which the observed redundant "manifest variables" are generated [5, Chaper 1]. Generating the manifest variables is the inverse problem to the common procedure in FA of identifying the latent variables from the manifest variables.

We select the principal component analysis (PCA) [35,19] as our method for FA because contrary to general FA [8, p. 585] it has an explicit inverse. PCA models a special case of FA analysis that assumes that the observations are generated as linear combinations without an additive noise term [5, p. 52].

Because it can be regarded as a method for dimensionality reduction, our approach of performing an inverse PCA can be seen as a dimensionality expansion of a small number of random variables, which introduces redundancy. Afterward, we verify that each considered subset of the created high-dimensional manifest variables still contains all information of the low-dimensional latent variables.

PCA performs a transformation that maximizes the variance of the projected data [8]. For this, the data is made zero-mean and then transformed with the real orthogonal matrix $\mathbf{U}^T$ where the columns of $\mathbf{U} = (u_1, ..., u_n)$ are the eigenvectors corresponding to the sorted eigenvalues of the empirical covariance matrix. When performing a dimensionality reduction from n manifest variables to m latent variables with $n > m$, the data is transformed with $\mathbf{U'}^T$, where $\mathbf{U'} = (u_1, ..., u_m)$ consists of the first m columns of $\mathbf{U}$.

Since we are interested in an inverse PCA with dimensionality expansion, we generate our observations x_n of the manifest variables from the generated instances x_m of the latent variables as:

$$x_n = \mathbf{U'} x_m. \tag{6}$$

That is, to generate the observed manifest variables, we uniformly sample $\mathbb{U} \in \mathbb{R}^{9\times 9}$ as a random orthogonal matrix with standard-normal distributed coefficients based on the Haar measure [28,10]. We obtain $\mathbf{U}' \in \mathbb{R}^{9\times 6}$ by selecting six columns from U to add redundancy of three features.

To guarantee the desired redundancy, we can ensure that no information from the latent variables is lost when only considering 6 out of the 9 manifest variables by proving that the latent variables can be reconstructed from the subset of manifest variables. We achieve this by checking the matrix constructed from the rows of $\mathbf{U}'$ generating the considered manifest variables for left invertibility [45].

4.2 Colored MNIST

Colored MNIST [2] was created by Arjovsky et al. as an adapted version of the MNIST dataset [9] for hand-written digit recognition that is designed for the evaluation of debiasing methods.

The authors introduce colored digits and use the additional color information to propose the following binary task: First, they split the images of MNIST into digits < 5 and ≥ 5. Each group represents one of the binary classification labels. However, the authors flip this label with a probability of 0.25. Then, they color the digits based on the label. Similarly to the label, the color is flipped with a certain probability as well. Following Arjovsky et al., we assign the colors so that red digits are generally associated with label 1 and a digit < 5 and green digits are generally associated with label 0 and a digit ≥ 5.

Arjovsky et al. set the color flip probabilities such that in the training dataset the label is more closely associated with the color than with the digit, but not in the test dataset. That is, a model is driven towards learning the color information as a bias, which hurts generalization to the test distribution. This allows the authors to evaluate their debiasing methods. We follow Arjovsky et al. and create our training environment with a color flip probability of 0.2. To be able to perform hyperparameter tuning, the validation environment is created equally. Then, the test environment is created with a color flip probability of 0.5 so that there is no spurious association between color and label in this environment.

Dataset Statistics. We show that for the training distribution, the maximum performance of a model cannot be improved with additional knowledge of the digits compared to only knowing their color. To demonstrate this, we show the optimal decision strategy does not change with additional knowledge of the digits. As a consequence, the minimal error under the optimal decision strategy is the same under both circumstances. This indicates that only with successful debiasing a model is incentivized to learn digit recognition when trained on the training distribution.

The optimal decision strategy when only knowing a digit's color is predicting the label that has the higher probability given the color. The probabilities can directly be inferred from the color flip probabilities. For red digits label 1 and for green digits label 0 is predicted:

$$\mathbb{P}(\text{Label} = 1|\text{Color} = \text{red}) = 0.8, \tag{7}$$

$$\mathbb{P}(\text{Label} = 0|\text{Color} = \text{green}) = 0.8. \tag{8}$$

With additional knowledge of the digit, the optimal decision strategy consists of predicting the label that has the highest probability given the color and digit. The required conditional probabilities can be calculated via joint probabilities following the definition of the conditional probability of random variables [27, p. 151] (for more details see Appendix A.1):

$$\mathbb{P}(\text{Label} = 1|\text{Color} = \text{red}, \text{Digit} < 5) > 0.5, \tag{9}$$

$$\mathbb{P}(\text{Label} = 1|\text{Color} = \text{red}, \text{Digit} \geq 5) > 0.5, \tag{10}$$

$$\mathbb{P}(\text{Label} = 0|\text{Color} = \text{green}, \text{Digit} \geq 5) > 0.5, \tag{11}$$

$$\mathbb{P}(\text{Label} = 0|\text{Color} = \text{green}, \text{Digit} < 5) > 0.5. \tag{12}$$

The optimal decision strategy still consists of predicting 1 for red digits and 0 for green digits.

Since the optimal decision strategy has not changed, the minimum error E achievable under the optimal decision strategy when only knowing the color and also with additional knowledge of the digit is the same under both circumstances and can be calculated as:

$$\begin{aligned} E &= \sum_{c \in \{\text{red},\text{green}\}} \mathbb{P}(\text{False Prediction}|\text{Color} = c) \cdot \mathbb{P}(\text{Color} = c) \\ &= 0.2 \cdot 0.5 + 0.2 \cdot 0.5 = 0.2. \end{aligned} \tag{13}$$

This shows that a model following the optimal decision strategy can achieve an accuracy of 0.8. In other words, a model reaching a higher accuracy on the training dataset of Colored MNIST must have memorized training samples.

5 Experiments

We test our feature steering method on the two datasets presented in the previous section and demonstrate that it makes both discouragement and encouragement of features possible. Additionally, we examine the model behavior depending on the choice of the weight factor λ.

5.1 Evaluation Metrics

First, we describe how the success of our feature steering method is evaluated. Designing an evaluation metric for feature steering is non-trivial because it has to encompass both objectives of feature steering: Correct predictions and manipulation of the influence of individual features.

Debiasing. In debiasing, it is common to have a training and a test dataset that only differ with respect to the spurious association between bias and label, which is only present in the training distribution. Under the assumption that the model is incentivized to learn the bias when trained on the biased training distribution, debiasing can be evaluated via the performance on the unbiased test dataset. As long as we are only interested in the absolute discouragement of features, we can follow this evaluation approach for feature steering.

General Feature Steering. The evaluation approach for debiasing cannot be applied to general feature steering because it implicitly fixes the desired strength of the feature steering to a total discouragement. Since feature steering also includes partial discouragement or encouragement of features, we choose to evaluate both feature steering objectives separately.

To ensure correct predictions on the original distribution, we consider the generalization error on the test dataset. To evaluate the manipulation of the influence of the specific features on the model's prediction process, similarly to debiasing, we evaluate the performance of the models on a dataset with a distribution shift where there is no spurious association between the feature of interest and the label. For our regression dataset, this can easily be created by replacing the input that corresponds to the feature of interest with random normal-distributed noise with the same mean and variance as the original feature.

To measure the influence of a feature i we consider the difference in performance on the original dataset and the dataset with a distribution shift for i. As both datasets have the same labels, a difference in the maximum-likelihood loss on the datasets can only be attributed to the feature i or its interactions. We additionally normalize this difference to compare the feature steering effect on different datasets.

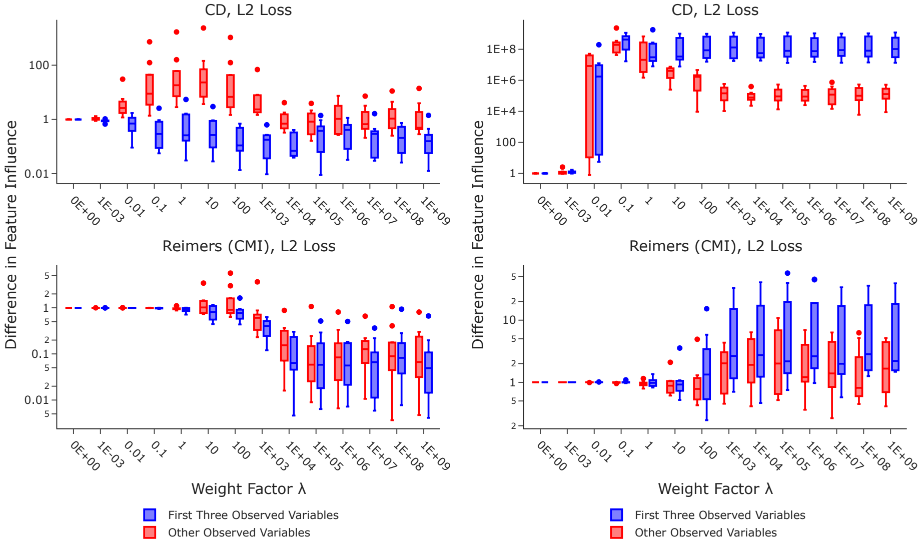
For a model trained with weight factor λ that achieves a maximum-likelihood loss of $\mathcal{E}(\lambda)$ on the original dataset and $\mathcal{E}_i(\lambda)$ on the version manipulated for feature i, the influence $influence(\lambda, i)$ of this feature on the model's prediction process is calculated as:

$$influence(\lambda, i) = \left| \frac{\mathcal{E}_i(\lambda) - \mathcal{E}(\lambda)}{\mathcal{E}_i(0) - \mathcal{E}(0)} \right|. \tag{14}$$

5.2 Results on Redundant Regression Dataset

We generate 9 instances of the redundant regression dataset presented in Section 4.1 with different regression coefficients $\beta_0, ..., \beta_5$ and transformation matrices U'. For each dataset, we generate 1400 training samples and 300 datasets for evaluation. Since we introduce a redundancy of three variables we perform discouragement and encouragement with respect to the first three observed manifest variables.

Our evaluations are conducted with a network that consists of an initial linear layer with rectified linear units (ReLUs) [12,43,32] as activation functions and



(a) Discouragement (b) Encouragement (Ours)

Fig. 1: Evaluation of the feature selection. We perform encouragement and discouragement of the first three observed variables for each of the 9 instances of the redundant regression dataset with our feature steering method using L2 loss. The feature steering objective is evaluated based on the feature influences (as described in Section 5.1).

the same size as the network input followed by a linear output layer (for more details on the training process see Appendix A.2). The weights are initialized with Xavier initialization [14] and biases to zero. Optimization is performed with PyTorch's default AdamW implementation [23] and learning rate $\delta = 0.01$ for 90 epochs. We follow standard practice for a linear regression problem and train with the mean-squared error as our loss function.

Ablation Study. We examine the model behavior when discouraging and encouraging the first three observed manifest variables with our feature steering method depending on the weight factor. In our experiments, we perform discouragement and encouragement separately. That is, we limit Equation 1 to either D or E being the empty set. For this, we consider weight factors $\lambda = 10^{-3}, 10^{-2}, ..., 10^{9}$. The feature attributions for our feature steering method are obtained with contextual decomposition (CD) [30,44] and following Reimers et al. [38] with CMI.

The evolution of the feature influence determined as described in the previous section is shown in Figure 1. The prediction performance is measured via the mean-squared error on the validation dataset, which can be found in Table 1. In

Table 1: Evaluation of the prediction correctness. We consider the maximum-likelihood loss on the test distribution averaged over the instances of the dataset to evaluate the correctness of the predictions generated for encouragement and discouragement of the first three observed variables with L2 loss. In conjunction with Figure 1, λ can be selected as a tradeoff between feature steering and correct predictions.

λ	Discouragement		**Encouragement (Ours)**	
	CD	Reimers (CMI)	CD	Reimers (CMI)
$\lambda = 0$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
$\lambda = 0.01$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$45.500_{\pm 61.339}$	$0.000_{\pm 0.000}$
$\lambda = 1$	$0.002_{\pm 0.002}$	$0.000_{\pm 0.000}$	$9.373 \cdot 10^6{}_{\pm 6.873 \cdot 10^6}$	$0.005_{\pm 0.001}$
$\lambda = 100$	$0.029_{\pm 0.031}$	$0.032_{\pm 0.009}$	$5.218 \cdot 10^7{}_{\pm 5.903 \cdot 10^6}$	$0.413_{\pm 0.034}$
$\lambda = 10^4$	$0.657_{\pm 0.107}$	$0.911_{\pm 0.117}$	$5.461 \cdot 10^7{}_{\pm 7.410 \cdot 10^6}$	$1.753_{\pm 0.301}$
$\lambda = 10^6$	$0.980_{\pm 0.232}$	$0.957_{\pm 0.102}$	$5.764 \cdot 10^7{}_{\pm 1.087 \cdot 10^7}$	$1.873_{\pm 0.345}$

the following, we only consider the observations for the L2 loss but the results for the L1 loss are very similar (see Appendix A.2).

We find that feature steering generally appears to be successful. However, recall from Section 3.3 that extreme values of the weight factor λ are expected to lead to suppression of one of the feature steering objectives. We can particularly observe this pathological behavior for encouragement with CD (see Appendix A.2). Additionally, we observe that the feature steering is very sensitive to λ.

5.3 Results on Colored MNIST

For Colored MNIST, we discourage the color as a spurious bias feature. The experiments are performed with the baseline architecture presented by Arjovsky et al. in their introduction of Colored MNIST [2].

We perform feature attribution for our feature steering method with the conditional-independence-based feature attribution method by Reimers et al. [38]. Because this method expects batched learning, we adapt the training process for batched learning with a batch size of 100, similar to the experiments on the redundant regression dataset. We train the network for 50 epochs (see Appendix A.3).

Ablation Study. We perform discouragement of the color for weight factors $\lambda = 10^{-3}, 10^{-2}, ..., 10^9$ with feature attributions generated with the conditional-independence-based method proposed by Reimers et al. [38]. The feature steering results evaluated as binary accuracies for L2 loss can be found in Figure 2 (for L1 loss, see Appendix A.3).

Due to the construction of Colored MNIST, a successful feature steering should be indicated by an increase in test accuracy. When applying our method

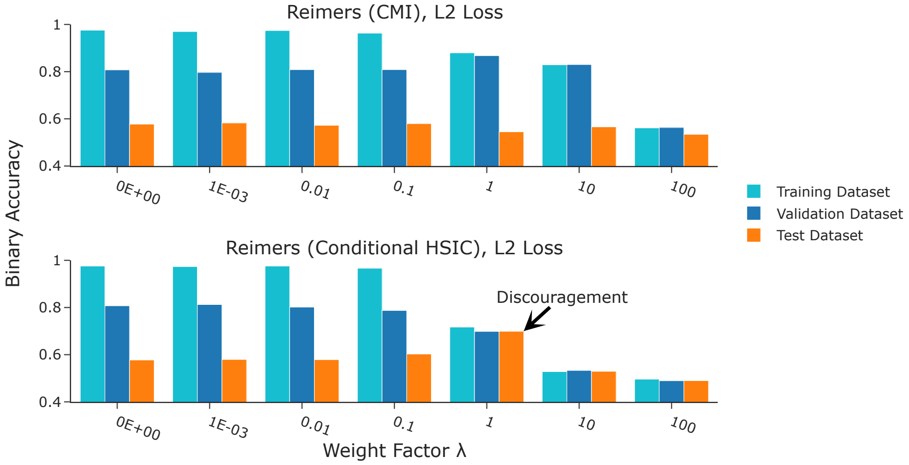


Fig. 2: Evaluation of discouragement on Colored MNIST. For Colored MNIST, we attempt to discourage the model from using the digits' color for prediction. For this, we follow Reimers et al. [38] to generate feature attributions with CMI and conditional HSIC. Because the bias is only present in the training and validation dataset, the success of discouragement can be observed as an increase in test accuracy. An accuracy above 0.8 on the training dataset indicates memorization.

with feature attributions generated with the conditional-independence-based method by Reimers et al. and conditional HSIC, we can observe an increase in accuracy from 0.58 to 0.70. With feature attributions based on CMI, we do observe such a clear increase.

Since we have shown in Section 4.2 that the maximum accuracy on the training distribution achievable under the optimal decision strategy without memorization is 0.8, we can conclude from the observations that the model overfits to the training data. However, this does not seem to impact the generalization to the validation dataset. Our feature steering method appears to stop this memorization even when performed with CMI without impacting the performance on the validation dataset.

6 Conclusions

In this work, we address the alignment of the feature selection process with domain knowledge. In contrast to prior works from the area of debiasing, we present a method that allows for both the discouragement and encouragement of arbitrary features. Our evaluation indicates that it can be used to integrate domain knowledge about well-established features during the model training, aiming at the improvement of the generalization capabilities of and trust in machine learning models.

We observe that our method is very sensitive to the weight factor λ. Additionally, pathological solutions like extreme model outputs for extreme discouragement or encouragement have to be avoided.

We only consider loss-based feature steering. In the future, we plan to investigate other how feature steering can be achieved by other regularization methods like a manipulation of the sampling process. It would also be beneficial to further investigate the evaluation of feature steering. This includes the development of an evaluation metric that can fairly incorporate both the correctness of predictions and the success of feature steering.

References

1. Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant Risk Minimization (2019)
3. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.R.: How to Explain Individual Classification Decisions. J. Mach. Learn. Res. **11**, 1803–1831 (2010)
4. Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020)
5. Bartholomew, D.J.: Latent variable models and factor analysis, Griffin's statistical monographs and courses, vol. 40. Oxford Univ. Press and Griffin, New York and London (1987)
6. Basilevsky, A.: Statistical Factor Analysis and Related Methods: Theory and Applications. Wiley series in probability and mathematical statistics. Probability and mathematical statistics, Wiley InterScience, New York, NY, USA and Chichester and Brisbane and Toronto and Singapore (1994)
7. Bertsekas, D.P.: Constrained optimization and Lagrange multiplier methods, Optimization and neural computation series, vol. 4. Athena Scientific, Belmont, Mass. (1996)
8. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
9. Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Jackel, L.D., LeCun, Y., Muller, U.A., Säckinger, E., Simard, P., Vapnik, V.: Comparison of classifier methods: a case study in handwritten digit recognition. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition (Cat. No.94CH3440-5). pp. 77–82. IEEE Comput. Soc. Press (1994)
10. Diestel, J., Spalsbury, A.: Joys of Haar Measure, Graduate Studies in Mathematics, vol. v.150. American Mathematical Society, Providence (2014)
11. Erion, G., Janizek, J.D., Sturmfels, P., Lundberg, S.M., Lee, S.I.: Improving performance of deep learning models with axiomatic attribution priors and expected gradients. Nature Machine Intelligence **3**(7), 620–631 (2021)
12. Fukushima, K.: Cognitron: a self-organizing multilayered neural network. Biological cybernetics **20**(3-4), 121–136 (1975)
13. Gao, Y., Gu, S., Jiang, J., Hong, S.R., Yu, D., Zhao, L.: Going Beyond XAI: A Systematic Survey for Explanation-Guided Learning (2022)

14. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (2010)
15. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)
16. Gretton, A., Fukumizu, K., Teo, C.H., Le Song, Schölkopf, B., Smola, A.J.: A Kernel Statistical Test of Independence. In: Proceedings of the 20th International Conference on Neural Information Processing Systems. pp. 585–592. NIPS'07, Curran Associates Inc, Red Hook, NY, USA (2007)
17. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys **51**(5), 1–42 (2019)
18. Hinnefeld, J.H., Cooman, P., Mammo, N., Deese, R.: Evaluating Fairness Metrics in the Presence of Dataset Bias (2018)
19. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of educational psychology **24**(6), 417–441 (1933)
20. Linfoot, E.H.: An informational measure of correlation. Information and Control **1**(1), 85–89 (1957)
21. Lipton, Z.C.: The Mythos of Model Interpretability. Queue **16**(3), 31–57 (2018)
22. Liu, F., Avci, B.: Incorporating Priors with Feature Attribution on Text Classification. In: Korhonen, A., Traum, D., Màrquez, L. (eds.) Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 6274–6283. Association for Computational Linguistics, Stroudsburg, PA (2019)
23. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019 (2019)
24. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization **26**(6), 369–395 (2004)
25. McKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 4 edn. (2005)
26. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A Survey on Bias and Fairness in Machine Learning. ACM Computing Surveys **54**(6), 1–35 (2022)
27. Mendenhall, W., Beaver, R.J., Beaver, B.M.: Introduction to probability and statistics. Brooks/Cole, Belmont, Calif., 13. ed. edn. (2009)
28. Mezzadri, F.: How to generate random matrices from the classical compact groups. NOTICES of the AMS **54**(5) (2007)
29. Moradi, R., Berangi, R., Minaei, B.: A survey of regularization strategies for deep models. Artificial Intelligence Review **53**(6), 3947–3986 (2020)
30. Murdoch, W.J., Liu, P.J., Yu, B.: Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In: International Conference on Learning Representations (2018)
31. Nachbar, F., Stolz, W., Merkle, T., Cognetta, A.B., Vogt, T., Landthaler, M., Bilek, P., Braun-Falco, O., Plewig, G.: The ABCD rule of dermatoscopy. High prospective value in the diagnosis of doubtful melanocytic skin lesions. Journal of the American Academy of Dermatology **30**(4), 551–559 (1994)
32. Nair, V., Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. pp. 807–814. ICML'10, Omnipress, Madison, WI (2010)

33. Parraga, O., More, M.D., Oliveira, C.M., Gavenski, N.S., Kupssinskü, L.S., Medronha, A., Moura, L.V., Simões, G.S., Barros, R.C.: Debiasing Methods for Fairer Neural Models in Vision and Language Research: A Survey (2022)
34. Pearl, J.: Causality: Models, reasoning, and inference. Cambridge Univ. Press, Cambridge, 1. publ edn. (2000)
35. Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572 (1901)
36. Polyanskiy, Y., Wu, Y.: Information Theory: From Coding to Learning. Cambridge, MA (2022+)
37. Reichenbach, H.: THE DIRECTION OF TIME. University of California Press, Berkeley, Los Angeles, London (1956)
38. Reimers, C., Bodesheim, P., Runge, J., Denzler, J.: Conditional Adversarial Debiasing: Towards Learning Unbiased Classifiers from Biased Data. In: Bauckhage, C., Gall, J., Schwing, A. (eds.) Pattern Recognition, Image Processing, Computer Vision, Pattern Recognition, and Graphics, vol. 13024, pp. 48–62. Springer International Publishing and Imprint Springer, Cham (2021)
39. Reimers, C., Penzel, N., Bodesheim, P., Runge, J., Denzler, J.: Conditional Dependence Tests Reveal the Usage of ABCD Rule Features and Bias Variables in Automatic Skin Lesion Classification. In: CVPR ISIC Skin Image Analysis Workshop (CVPR-WS). pp. 1810–1819 (2021)
40. Reimers, C., Runge, J., Denzler, J.: Determining the Relevance of Features for Deep Neural Networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020, Springer eBook Collection, vol. 12371, pp. 330–346. Springer International Publishing and Imprint Springer, Cham (2020)
41. Rieger, L., Singh, C., Murdoch, W.J., Yu, B.: Interpretations Are Useful: Penalizing Explanations to Align Neural Networks with Prior Knowledge. In: Proceedings of the 37th International Conference on Machine Learning. ICML'20 (2020)
42. Ross, A.S., Hughes, M.C., Doshi-Velez, F.: Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. In: Bacchus, F., Sierra, C. (eds.) Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. pp. 2662–2670. International Joint Conferences on Artificial Intelligence Organization, California (2017)
43. Rumelhart, D.E., McClelland, J.L.: A General Framework for Parallel Distributed Processing. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations, pp. 45–76 (1987)
44. Singh, C., Murdoch, W.J., Yu, B.: Hierarchical interpretations for neural network predictions. In: International Conference on Learning Representations (2019)
45. Tan, L.: Generalized inverse of matrix and solution of linear system equation. In: Tan, L. (ed.) A Generalized Framework of Linear Multivariable Control, pp. 38–50. Elsevier Science, Oxford (2017)
46. Wang, A., Liu, A., Zhang, R., Kleiman, A., Kim, L., Zhao, D., Shirai, I., Narayanan, A., Russakovsky, O.: REVISE: A Tool for Measuring and Mitigating Bias in Visual Datasets. International Journal of Computer Vision **130**(7), 1790–1810 (2022)
47. Yin, P., Lyu, J., Zhang, S., Osher, S.J., Qi, Y., Xin, J.: Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets. In: International Conference on Learning Representations (2019)
48. Zan, L., Meynaoui, A., Assaad, C.K., Devijver, E., Gaussier, E.: A Conditional Mutual Information Estimator for Mixed Data and an Associated Conditional Independence Test. Entropy (Basel, Switzerland) **24**(9) (2022)

Beyond Debiasing: Actively Steering Feature Selection via Loss Regularization
- Supplementary Material -

Jan Blunk1 [0009−0001−7037−3545], Niklas Penzel1 [0000−0001−8002−4130], Paul Bodesheim1 [0000−0002−3564−6528], and Joachim Denzler1 [0000−0002−3193−3300]

Computer Vision Group, Friedrich Schiller University Jena, 07743 Jena, Germany
{jan.blunk,niklas.penzel,paul.bodesheim,joachim.denzler}@uni-jena.de
https://inf-cv.uni-jena.de

Abstract. This document contains supplementary material for the paper *Beyond Debiasing: Actively Steering Feature Selection via Loss Regularization*. In Section A.1 more details regarding the Colored MNIST dataset [1] used for evaluation are discussed. This is followed by additional details and evaluation results on the redundant regression dataset in Section A.2 and on Colored MNIST in Section A.3.

A Appendix

A.1 Dataset Statistics of Colored MNIST

Conditional Probabilities. To calculate the conditional probabilities needed for the optimal decision strategy, we first determine the required joint probabilities. They can be found in Figure A.1.

Then, the conditional probabilities can be calculated following the definition of the conditional probability of two random variables A, B as $P(A|B) = \frac{\mathbb{P}(A \cap B)}{(\mathbb{P}(B))}$ [3, p. 151]:

$$\mathbb{P}(\text{Label} = 1|\text{Color} = \text{red}, \text{Digit} < 5) = \frac{0.1}{0.1 + 0.075} \approx 0.57, \tag{1}$$

$$\mathbb{P}(\text{Label} = 1|\text{Color} = \text{red}, \text{Digit} \geq 5) = \frac{0.3}{0.3 + 0.025} \approx 0.92, \tag{2}$$

$$\mathbb{P}(\text{Label} = 0|\text{Color} = \text{green}, \text{Digit} \geq 5) = \frac{0.1}{0.1 + 0.075} \approx 0.57, \tag{3}$$

$$\mathbb{P}(\text{Label} = 0|\text{Color} = \text{green}, \text{Digit} < 5) = \frac{0.3}{0.3 + 0.025} \approx 0.92. \tag{4}$$

A.2 Additional Results on Redundant Regression Dataset

Hyperparameter Selection. We decide to perform feature steering with a network that consists of stacked combinations of linear layers and rectified linear units (ReLUs) [2,7,5] concluded by a linear output layer. The hidden layers are

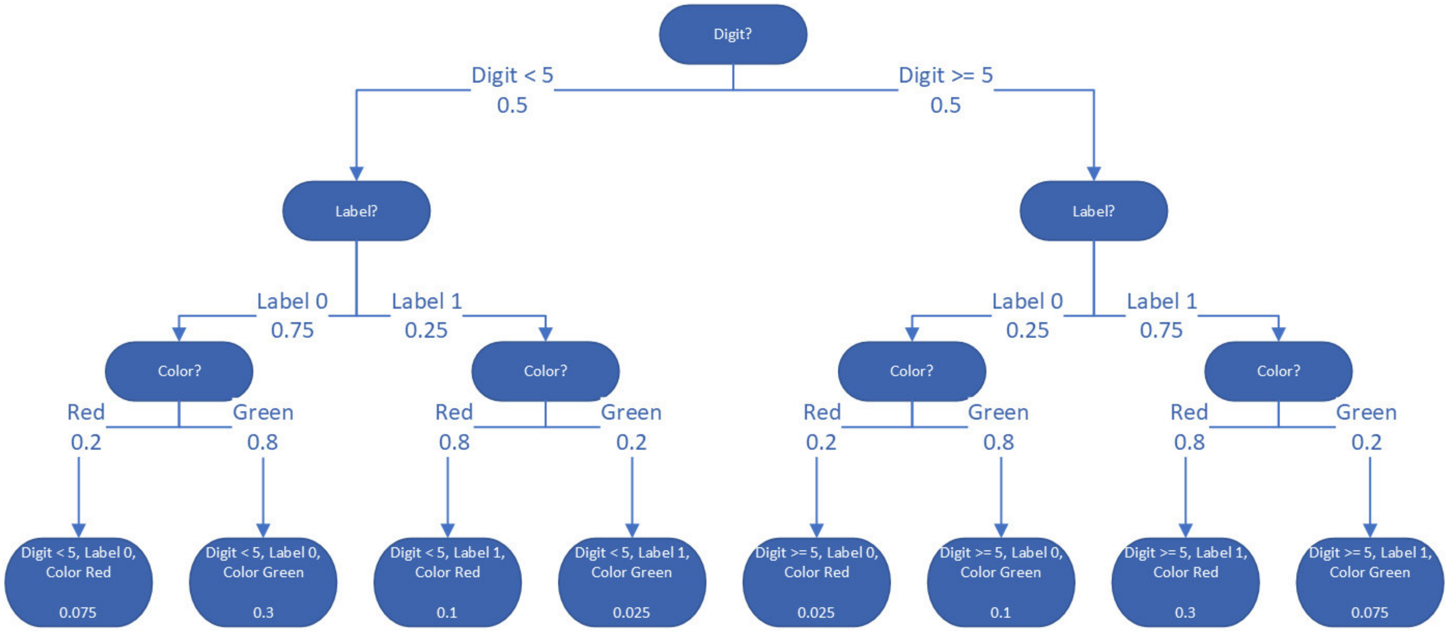

Digit?
Digit < 5
0.5
Digit >= 5
0.5
Label?
Label?
Label 0
0.75
Label 1
0.25
Label 0
0.25
Label 1
0.75
Color?
Color?
Color?
Color?
Red
0.2
Green
0.8
Red
0.8
Green
0.2
Red
0.2
Green
0.8
Red
0.8
Green
0.2
Digit < 5, Label 0, Color Red 0.075
Digit < 5, Label 0, Color Green 0.3
Digit < 5, Label 1, Color Red 0.1
Digit < 5, Label 1, Color Green 0.025
Digit >= 5, Label 0, Color Red 0.025
Digit >= 5, Label 0, Color Green 0.1
Digit >= 5, Label 1, Color Red 0.3
Digit >= 5, Label 1, Color Green 0.075

Fig. A.1: This figure visualizes the calculation of the joint probabilities for combinations of label, digit, and digit color for the training and validation distribution of Colored MNIST.

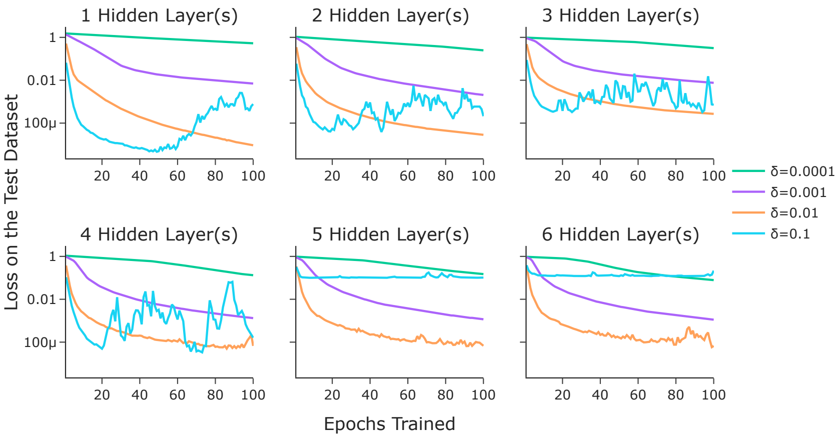
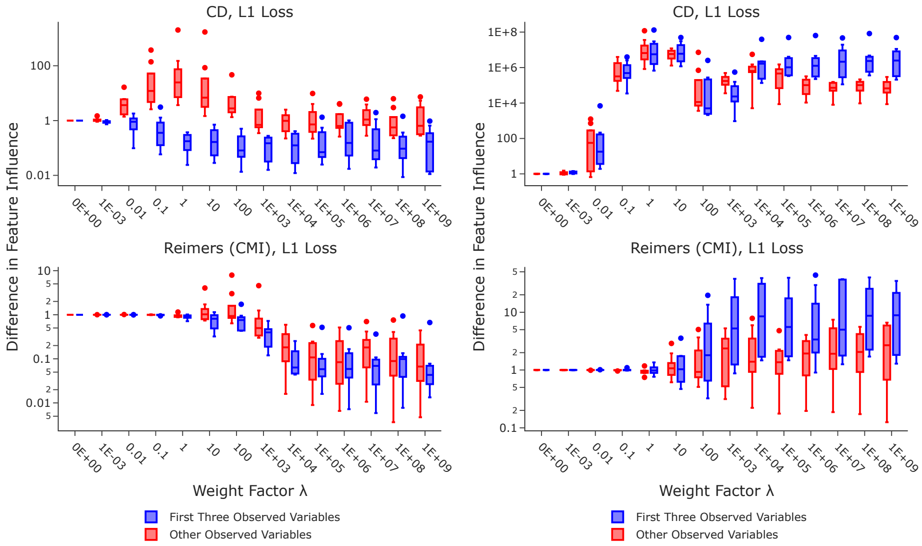


Fig. A.2: Network parameter selection for the redundant regression dataset. The average test loss per epoch on the redundant regression problem is evaluated for different numbers of hidden layers and learning rates δ. As a result of the hyperparameter search, we select a network with one hidden layer and learning rate $\delta = 0.01$, which is trained for 90 epochs.

of similar size to the network input. We tune the number of hidden layers and the learning rate δ for the task. For this, we perform a grid search with respect to the number of hidden layers and learning rates $\delta = 10^{-6}, ..., 10^{0}$. We evaluate the test loss averaged over all instances of the dataset, which is shown in Figure A.2.

Even though the optimal performance is reached with one hidden layer and $\delta = 0.1$ after training for 49 epochs, $\delta = 0.1$ shows a significantly less stable behavior for different numbers of hidden layers compared to the other learning rates. Therefore, we choose to perform our experiments with a network with one hidden layer but a learning rate of $\delta = 0.01$ since this learning rate performs well for all considered numbers of hidden layers. Because the performance does not show any notable further increase after 90 epochs, in our evaluations we restrict the training process to 90 epochs.

Ablation Study for L1 Loss. The feature steering evaluations presented in Section 5.2 have also been performed with L1 loss. The respective evaluation of the feature steering objective via feature influences estimated as described in Section 5.1 is shown in Figure A.3. The mean-squared error on the test dataset is used to evaluate the performance on the original distribution and can be found in Table 1.



(a) Discouragement

(b) Encouragement (Ours)

Fig. A.3: Evaluation of the feature selection for L1 loss. In addition to evaluating the feature steering objective for encouragement and discouragement of the first three observed manifest variables performed with L2 loss as shown in Figure 1, we also perform a similar evaluation for encouragement and discouragement with L1 loss. We again evaluate the feature steering objective based on the feature influences (as described in Section 5.1).

Pathological Solutions. We have described in Section 3.3 that for extreme weight factors λ, one of the feature steering objectives dominates, which can lead to undesired results. Additionally, we have demonstrated in Section 5.2 that these pathological solutions can be observed in practice.

To further examine them, we consider the absolute value of the model output, which is shown in Figure A.4. For large weight factors λ, the feature steering portion of the loss dominates, and when performing discouragement, we observe that the absolute value of the model output approaches a constant near-zero value. That is, the model reaches a trivial solution to discouragement: If the model output is constant the predictions are independent of all features, including the feature to discourage. For encouragement, we can also observe an undesired extreme change in the absolute value of the model outputs when feature attributions are determined with contextual decomposition [4,8].

These observations again highlight the importance of selecting λ as a tradeoff between the objectives of correct predictions and manipulation of the feature selection process.

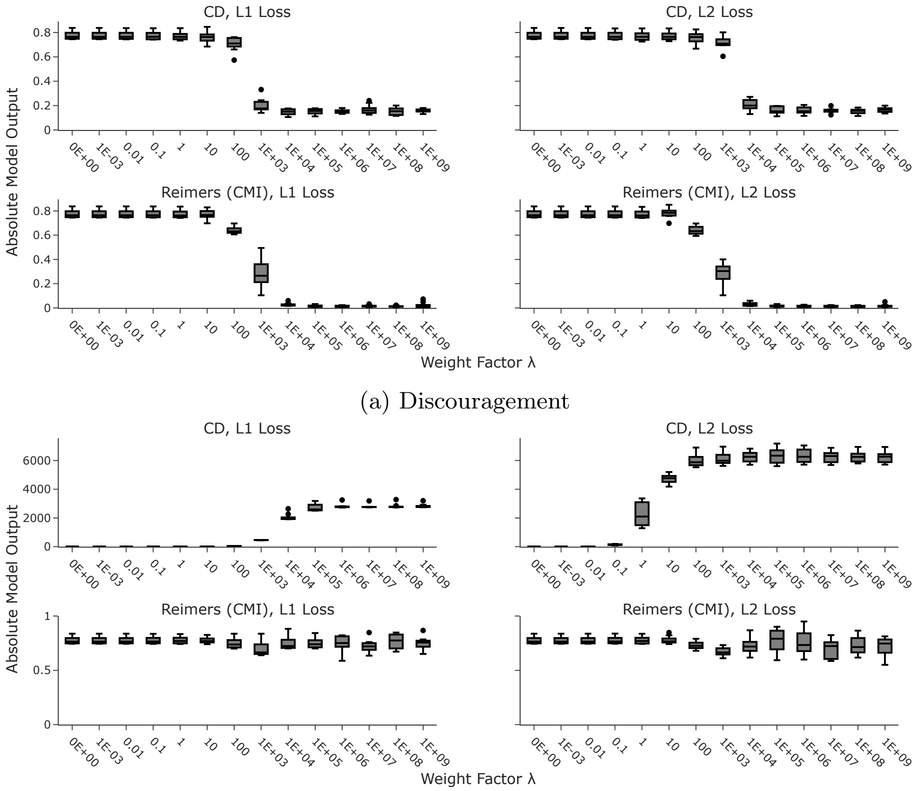
Table 1: Evaluation of the prediction correctness for L1 loss. We consider the maximum-likelihood loss on the test distribution averaged over the instances of the dataset to evaluate the correctness of the predictions generated for encouragement and discouragement of the first three observed manifest variables. In conjunction with Figure A.3b, λ can be selected as a tradeoff between feature steering and correct predictions.

λ	Discouragement		**Encouragement (Ours)**	
	CD	Reimers (CMI)	CD	Reimers (CMI)
$\lambda = 0$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
$\lambda = 0.01$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$	$0.000_{\pm 0.000}$
$\lambda = 1$	$0.010_{\pm 0.016}$	$0.000_{\pm 0.000}$	$1.642_{\pm 2.912}$	$0.005_{\pm 0.001}$
$\lambda = 100$	$0.103_{\pm 0.109}$	$0.033_{\pm 0.018}$	$3036.346_{\pm 382.364}$	$0.486_{\pm 0.070}$
$\lambda = 10^4$	$1.026_{\pm 0.213}$	$0.915_{\pm 0.110}$	$4.326 \cdot 10^6{}_{\pm 1.141 \cdot 10^6}$	$1.716_{\pm 0.253}$
$\lambda = 10^6$	$1.040_{\pm 0.216}$	$0.960_{\pm 0.109}$	$8.117 \cdot 10^6{}_{\pm 1.044 \cdot 10^6}$	$1.806_{\pm 0.282}$

A.3 Additional Results on Colored MNIST

Hyperparameter Selection. The only hyperparameter that needs to be tuned is the number of epochs that are trained. We select it based on the maximum-likelihood loss on the training dataset, which is depicted in Figure A.5. Since there is no notable further decrease in the training loss after 50 epochs, we select 50 as the number of epochs trained for our empirical evaluation of the feature steering.

Ablation Study for L1 Loss. The ablation study presented in Section 5.3 has also been conducted with L1 loss. The obtained evaluation results can be found in Figure A.6. They are very similar to the results for feature steering with L2 loss. We observe that feature steering with the conditional-independence-based feature attribution method by Reimers et al. [6] with conditional HSIC improves the accuracy on the test dataset to 0.64 compared to the initial accuracy of 0.58, which is 0.06 less than the increase observed for L2 loss.



(a) Discouragement

(b) Encouragement (Ours)

Fig. A.4: Further investigation of pathological solutions. We consider the absolute value of the model output when discouraging and encouraging the first three observed variables to explore the characteristics of the pathological solutions observed in Section 5.2 for large weight factors λ. For large values of λ, we observe a trend towards trivial solutions of the feature steering objective.

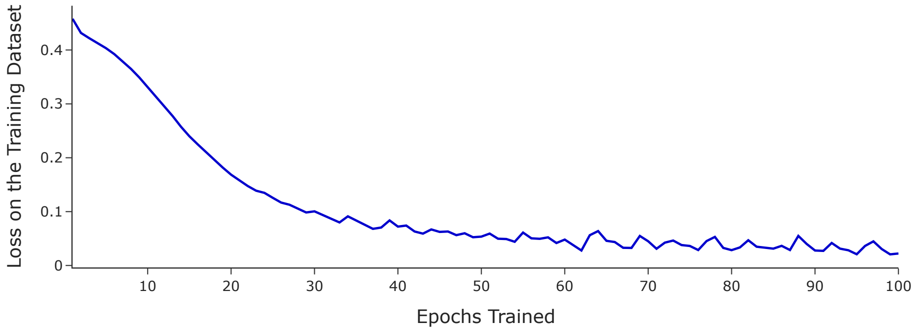


Fig. A.5: Network parameter selection for Colored MNIST. We consider the maximum-likelihood loss on the training dataset of Colored MNIST [1] for a model trained with maximum-likelihood loss to select the number of epochs trained in the empirical evaluations of our feature steering approach. Because the training loss already converges after 50 epochs, we restrict the training to 50 epochs.

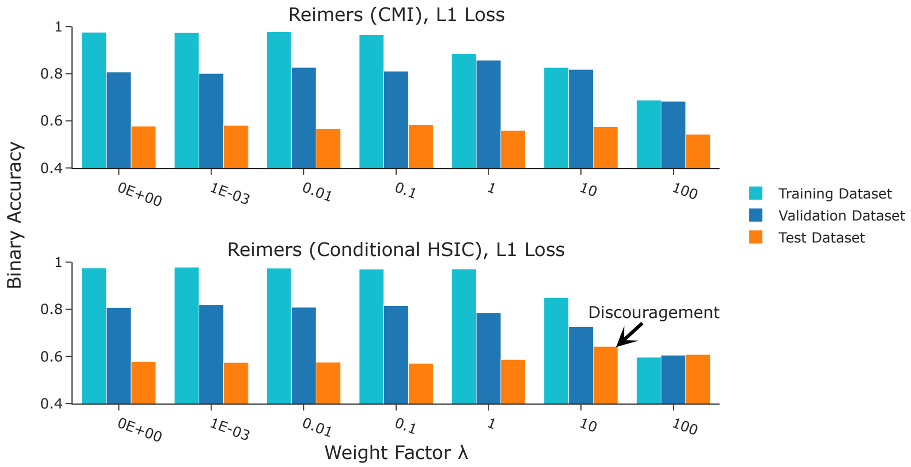


Fig. A.6: Evaluation of discouragement on Colored MNIST for L1 loss. For Colored MNIST, we attempt to discourage the model from using the digits' color for prediction. Here, the required feature attributions are generated with CMI and conditional HSIC following Reimers et al. [6]. The success of discouragement can be observed as an increase in test accuracy. A similar evaluation for L2 loss can be found in Figure 2.

References

1. Arjovsky, M., Bottou, L., Gulrajani, I., Lopez-Paz, D.: Invariant Risk Minimization (2019)
2. Fukushima, K.: Cognitron: a self-organizing multilayered neural network. Biological cybernetics **20**(3-4), 121–136 (1975)
3. Mendenhall, W., Beaver, R.J., Beaver, B.M.: Introduction to probability and statistics. Brooks/Cole, Belmont, Calif., 13. ed. edn. (2009)
4. Murdoch, W.J., Liu, P.J., Yu, B.: Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In: International Conference on Learning Representations (2018)
5. Nair, V., Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. pp. 807–814. ICML'10, Omnipress, Madison, WI (2010)
6. Reimers, C., Bodesheim, P., Runge, J., Denzler, J.: Conditional Adversarial Debiasing: Towards Learning Unbiased Classifiers from Biased Data. In: Bauckhage, C., Gall, J., Schwing, A. (eds.) Pattern Recognition, Image Processing, Computer Vision, Pattern Recognition, and Graphics, vol. 13024, pp. 48–62. Springer International Publishing and Imprint Springer, Cham (2021)
7. Rumelhart, D.E., McClelland, J.L.: A General Framework for Parallel Distributed Processing. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations, pp. 45–76 (1987)
8. Singh, C., Murdoch, W.J., Yu, B.: Hierarchical interpretations for neural network predictions. In: International Conference on Learning Representations (2019)