Removing Spurious Concepts from Neural Network Representations via Joint Subspace Estimation

Floris Holstege [1 2] **Bram Wouters** [1] **Noud van Giersbergen** [1] **Cees Diks** [1 2]

Abstract

An important challenge in the field of interpretable machine learning is to ensure that deep neural networks (DNNs) use the correct or desirable input features in performing their tasks. Concept-removal methods aim to do this by eliminating concepts that are spuriously correlated with the main task from the neural network representation of the data. However, existing methods tend to be overzealous by inadvertently removing part of the correct or desirable features as well, leading to wrong interpretations and hurting model performance. We propose an iterative algorithm that separates spurious from main-task concepts by jointly estimating two low-dimensional orthogonal subspaces of the neural network representation. By evaluating the algorithm on benchmark datasets from computer vision (Waterbirds, CelebA) and natural language processing (MultiNLI), we show it outperforms existing concept-removal methods in terms of identifying the main-task and spurious concepts, while removing only the latter.

1. Introduction

Although deep neural networks (DNNs) have achieved impressive results in computer vision and language modeling, it is notoriously difficult to control which concepts are being used by DNNs in performing their tasks. A *concept* refers to a representation in the data of a human-defined object or phenomenon (Kim et al., 2018). For example, if a model's main task is to distinguish between images of cows and penguins, the training data might contain a spurious correlation between the concepts of background and animal type (cows typically appearing on grassland, and penguins typically in the snow). DNNs frequently rely on such spurious correlations within the data (Gururangan et al., 2018; Srivastava et al., 2020; Wang & Culotta, 2020; Sagawa et al., 2020b; Zhou et al., 2021). For example, they use the background to identify the animal in the picture. This can be problematic in several ways. It makes models less trustworthy. Penguins on grassland could be classified as cows, for example. It also makes models difficult to interpret, as it is unclear to what extent a model relies on which concept. Finally, this could lead to models using undesirable concepts (e.g., race or gender). Therefore, we would like to control whether or not a model uses a specific concept in performing its task.

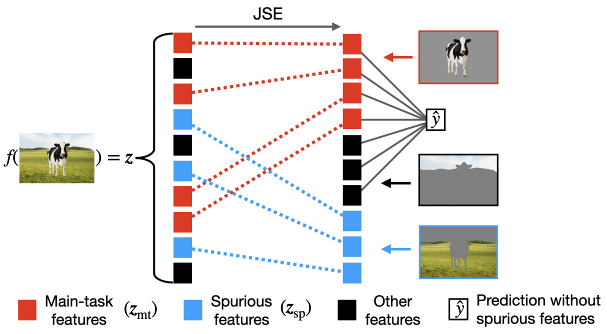
JSE
$f($ $) = z$
$\hat{y}$
Main-task features (z_{mt})
Spurious features (z_{sp})
Other features
Prediction without spurious features

Figure 1. **High-level overview of Joint Subspace Estimation (JSE) for concept removal**: the input $\boldsymbol{x}$ is fed through a neural network $f(\boldsymbol{x})$, from which we can extract the vector representation $\boldsymbol{z}$. Within the vector representation, two orthogonal subspaces are identified: one related to the *spurious concept* (e.g. the background), and one to the *main-task concept* (e.g. animal type).

Removing concepts directly from the training data (e.g., images or text) is non-trivial and costly. An alternative is to focus on the embeddings, which are vector representations of the data generated by the neural network. Post-hoc concept-removal methods aim to eliminate a concept from the embeddings, after the parameters of the neural network are frozen. Typically a concept classifier is trained on the embeddings, from which the concept features are then removed (Ravfogel et al., 2020; 2022a). In the example of distinguishing cows and penguins, we would train a classifier to predict the spurious concept of background type and use this to remove background features from the embeddings. Afterwards, a linear classifier is trained on the transformed

[1] University of Amsterdam, Department of Quantitative Economics [2] Tinbergen Institute. Correspondence to: Floris Holstege <f.g.holstege@uva.nl>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

embeddings to predict the main-task concept (e.g. animal type), preventing the model from using the spurious concept for main-task classification. Post-hoc concept-removal methods in the presence of spurious correlations can be used for out-of-distribution (OOD) generalization (Joshi et al., 2022) and for making models more interpretable (Elazar et al., 2021). They can also be used to remove sensitive information, such as gender (Bolukbasi et al., 2016).

A drawback of post-hoc concept-removal methods is that they tend to eliminate also main-task features from the embeddings (Belinkov, 2022; Kumar et al., 2022). This is because, due to the spurious correlation, a concept classifier might inadvertently use main-task features to predict the spurious concept. For example, a cow's horns could be used to predict a grassland background. As a consequence, using such a concept classifier to remove the spurious concept from the embeddings will also remove information associated with the main task. This potentially hurts main-task model performance and also could lead to wrong interpretations of the model after concept removal. It therefore limits the application of removal methods in the most relevant cases, namely when the spurious correlation is strong and likely to be exploited by the DNN.

Our contribution: this paper proposes a novel post-hoc concept-removal method by jointly identifying two low-dimensional orthogonal subspaces, one associated with the spurious concept (e.g. background) and the other with the main-task concept (e.g. animal type). This crucially differs from existing methods, which only use the spurious concept to determine the subspace of spurious concept features. Furthermore, we make the identification of the subspaces systematic by introducing statistical tests that attribute directions in the embedding space to either the main-task or the spurious concept. The method, which we call Joint Subspace Estimation (JSE), is shown to be able to remove spurious features while retaining vital main-task information, also in the case of strong spurious correlations. Applied to benchmark datasets for image recognition (Waterbirds, CelebA) and natural language processing (MultiNLI), JSE is shown to outperform existing concept-removal methods in its ability to identify the main-task and spurious concepts, and to remove only the latter. A high-level overview of the method is given in Figure 1. Our code with an implementation of JSE is publicly available.[*]

2. Spurious Correlations and Concept Removal

We consider the random variables $\mathcal{D} = (y_{\mathrm{mt}}, y_{\mathrm{sp}}, \boldsymbol{x})$, where $y_{\mathrm{mt}} \in \{0, 1\}$ is the main-task concept label, $y_{\mathrm{sp}} \in \{0, 1\}$ is the spurious concept label and $\boldsymbol{x} \in \mathcal{X}$ represents the input features. Each input $\boldsymbol{x}$ contains subsets $\boldsymbol{x}_{\mathrm{mt}}$ and $\boldsymbol{x}_{\mathrm{sp}}$ of features corresponding to the main-task and spurious concept, respectively. In the example of cows and penguins, $\boldsymbol{x}_{\mathrm{mt}}$ and $\boldsymbol{x}_{\mathrm{sp}}$ correspond to the pixels showing the animal and the background. It is assumed that $\boldsymbol{x}_{\mathrm{mt}}$ and $\boldsymbol{x}_{\mathrm{sp}}$ causally determine the associated labels y_{mt} and y_{sp}, respectively. Their joint joint probability density is then given by:

$$p\left(y_{\mathrm{mt}}|\boldsymbol{x}_{\mathrm{mt}}\right) p\left(y_{\mathrm{sp}}|\boldsymbol{x}_{\mathrm{sp}}\right) p\left(\boldsymbol{x}_{\mathrm{mt}}, \boldsymbol{x}_{\mathrm{sp}}\right). \quad (1)$$

This implies that the main-task label y_{mt} is conditionally independent of the spurious features, $\boldsymbol{x}_{\mathrm{sp}}$, but it does not mean that y_{mt} and $\boldsymbol{x}_{\mathrm{sp}}$ are independent; they can be dependent due to dependence between $\boldsymbol{x}_{\mathrm{mt}}$ and $\boldsymbol{x}_{\mathrm{sp}}$. Since they are not causally related, we say they are spuriously correlated. At the level of trained neural networks, this means that a main-task classifier tends to make use of the spurious features $\boldsymbol{x}_{\mathrm{sp}}$ within $\boldsymbol{x}$. Previous work offers a number of possible reasons, ranging from stochastic gradient descent (SGD) training dynamics (Pezeshki et al., 2021) and overparameterization (Sagawa et al., 2020b; D'Amour et al., 2022) to inductive biases of DNNs (Rahaman et al., 2019).

We now restrict our analysis to DNNs for classification, which typically is done using a complicated function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}^d$ mapping the input features to a vector representation, followed by a linear layer. We assume the embedding vectors $\boldsymbol{z} \in \mathbb{R}^d$ have a similar structure as the input features $\boldsymbol{x}$, in the sense that each $\boldsymbol{z}$ has subsets $\boldsymbol{z}_{\mathrm{mt}} \in \mathcal{Z}_{\mathrm{mt}} \subseteq \mathbb{R}^d$ and $\boldsymbol{z}_{\mathrm{sp}} \in \mathcal{Z}_{\mathrm{sp}} \subseteq \mathbb{R}^d$ that causally determine the labels y_{mt} and y_{sp}, respectively. It should be stressed that this does not necessarily hold in practice. The trained DNN could have mixed the different features, or have one of them removed, because of their predictive ability for the main-task label y_{mt}. However, there is empirical evidence that when trained on data with a spurious correlation, neural networks tend to learn both main-task and spurious features (Kirichenko et al., 2023; Izmailov et al., 2022; Rosenfeld et al., 2022).

In addition, we assume that $\mathcal{Z}_{\mathrm{mt}}$ and $\mathcal{Z}_{\mathrm{sp}}$ are linear subspaces of the embedding space $\mathbb{R}^d$. This sometimes goes under the name of *linear subspace hypothesis* (Bolukbasi et al., 2016; Vargas & Cotterell, 2020). Previous work shows that linear subspaces can encode information about complex concepts (Bau et al., 2017). Moreover, non-linear information about the spurious concept cannot be used by the last layer for binary classification (Ravfogel et al., 2023).

One possible concept-removal approach is to project the embedding vectors $\boldsymbol{z}$ onto a linear subspace, before feeding them to a linear classifier. Suppose $\boldsymbol{v}_{\mathrm{sp},1}, \boldsymbol{v}_{\mathrm{sp},2}, \ldots, \boldsymbol{v}_{\mathrm{sp},d_{\mathrm{sp}}}$ is an orthonormal basis of the spurious embedding subspace $\mathcal{Z}_{\mathrm{sp}} \subseteq \mathbb{R}^d$ and $\boldsymbol{V}_{\mathrm{sp}}$ is the matrix $(d \times d_{\mathrm{sp}})$ whose columns are the basis vectors. Then the transformation from $\boldsymbol{z}$ to $(\boldsymbol{I} - \boldsymbol{V}_{\mathrm{sp}}\boldsymbol{V}_{\mathrm{sp}}^{\top})\boldsymbol{z}$ is the projection onto the orthogonal complement of $\mathcal{Z}_{\mathrm{sp}}$ and thereby removes the spurious features from the

[*] https://github.com/fholstege/JSE

representation. A linear classifier that uses the transformed embeddings to predict the binary main-task label y_{mt} will not use the spurious features.

In practice, however, it is highly non-trivial to estimate (the basis of) the subspace $\mathcal{Z}_{\mathrm{sp}}$. Due to the spurious correlations, classifiers that use the embedding vectors to predict the spurious label y_{sp} also make use of the main-task embeddings $\boldsymbol{z}_{\mathrm{mt}}$. As a consequence, an estimate of $\mathcal{Z}_{\mathrm{sp}}$ will also contain directions that are actually part of $\mathcal{Z}_{\mathrm{mt}}$. Projecting out the estimate of $\mathcal{Z}_{\mathrm{sp}}$ removes main-task information and therefore hurts the performance of the resulting main-task classifier (Ravfogel et al., 2020; Belinkov, 2022; Kumar et al., 2022). Our JSE method addresses this problem by estimating not only $\mathcal{Z}_{\mathrm{sp}}$, but also the main-task embedding space $\mathcal{Z}_{\mathrm{mt}}$.

2.1. Related Work

Concept-removal methods: Concept removal was initially based on adversarial approaches (Goodfellow et al., 2014), typically to mitigate undesirable biases (Edwards & Storkey, 2016; Zhang et al., 2018; Wang et al., 2019; 2021). This is frequently referred to as adversarial removal (ADV). However, the ability of these methods to remove concepts has been called into question (Elazar & Goldberg, 2018). An alternative is to remove a linear subspace from the embeddings (Bolukbasi et al., 2016; Ethayarajh et al., 2019; Dev & Phillips, 2019; Dev et al., 2021). A key method in this category is iterative null-space projection (INLP, Ravfogel et al., 2020), in which a linear classifier predicts the concept labels, and the coefficients of the classifier are orthogonally projected from the embeddings. This is repeated until the concept can no longer be predicted. A follow-up method is relaxed linear adversarial concept erasure (RLACE), in which an orthogonal projection matrix is trained such that the concept cannot be predicted (Ravfogel et al., 2022a). More recently, least-squares concept erasure (LEACE) was proposed, which provably prevents all linear classifiers from predicting concept labels (Belrose et al., 2023).

Interpretability: An interpretable DNN obeys domain-specific constraints, allowing it to be better understood by humans (Rudin et al., 2022). Concept-removal methods contribute to interpretability by allowing for greater control of which features are used by the DNN, as well as a better understanding of which features are used. In terms of control, concept-removal methods can be used to remove protected attributes, or sensitive information (Xu et al., 2017). Regarding understanding, concept-removal methods can be used by observing how a neural network responds after the concept has been removed from its embeddings (Elazar et al., 2021; Ravfogel et al., 2021). However, this approach has been called into question because they encode other information in addition to the concept (Belinkov, 2022; Kumar et al., 2022). JSE addresses this limitation, allowing greater control over spurious concepts, and understanding whether these are used by DNNs.

Spurious correlations: The problem of neural networks relying on spurious correlations has arisen in both computer vision (Geirhos et al., 2019; Xiao et al., 2021; Singla & Feizi, 2022) and NLP (Niven & Kao, 2019; Kaushik & Lipton, 2018; McCoy et al., 2019). There is a wide range of methods addressing spurious correlations in neural networks, including data augmentation (Hermann et al., 2020), invariant learning (Arjovsky et al., 2019; Ahuja et al., 2021), or instance-reweighting (Sagawa et al., 2020a;b; Idrissi et al., 2022). The latter category is most akin to concept-removal methods, as it uses (limited) availability of spurious concept labels, but lacks the advantage of interpretability.

3. Joint Subspace Estimation

We will now introduce Joint Subspace Estimation (JSE) in which the spurious and main-task embedding subspaces $\mathcal{Z}_{\mathrm{sp}}$ and $\mathcal{Z}_{\mathrm{mt}}$ are estimated simultaneously. Section 3.1 explains how to simultaneously estimate individual basis vectors for $\mathcal{Z}_{\mathrm{sp}}$ and $\mathcal{Z}_{\mathrm{mt}}$. Section 3.2 introduces an iterative procedure to find multiple basis vectors for $\mathcal{Z}_{\mathrm{sp}}$ and $\mathcal{Z}_{\mathrm{mt}}$. Section 3.3 presents two statistical tests to stop the iterative procedure and determine the dimensions of $\mathcal{Z}_{\mathrm{sp}}$ and $\mathcal{Z}_{\mathrm{mt}}$.

3.1. Estimating Spurious and Main-task Concept Vectors

As a starting point, consider simultaneously estimating one vector $\boldsymbol{v}_{\mathrm{sp}} \in \mathcal{Z}_{\mathrm{sp}}$ and another vector $\boldsymbol{v}_{\mathrm{mt}} \in \mathcal{Z}_{\mathrm{mt}}$. A usual approach for estimating $\boldsymbol{v}_{\mathrm{sp}}$ is to train a logistic regression on the embeddings, $\hat{y}_{\mathrm{sp}} = \mathrm{Logit}^{-1}(\boldsymbol{z}^\top \boldsymbol{w}_{\mathrm{sp}} + b_{\mathrm{sp}})$, and then use the (normalized) coefficients $\boldsymbol{v}_{\mathrm{sp}} = \boldsymbol{w}_{\mathrm{sp}}/||\boldsymbol{w}_{\mathrm{sp}}||$ as a so-called concept vector $\boldsymbol{v}_{\mathrm{sp}}$ that contains information about the concept (Kim et al., 2018). However, if we perform logistic regression in a sample where the spurious and main-task features are correlated, the estimate of $\boldsymbol{v}_{\mathrm{sp}}$ might have components in the direction of main-task features. To discourage the estimate of $\boldsymbol{v}_{\mathrm{sp}}$ to use main-task features (and vice versa for the estimate of $\boldsymbol{v}_{\mathrm{mt}}$), we make the following assumption about the relation between the two subspaces.

Orthogonality Assumption. *The linear subspaces $\mathcal{Z}_{\mathrm{sp}}$ and $\mathcal{Z}_{\mathrm{mt}}$ are orthogonal, i.e. each vector $\boldsymbol{v}_{\mathrm{sp}} \in \mathcal{Z}_{\mathrm{sp}}$ is perpendicular to each vector $\boldsymbol{v}_{\mathrm{mt}} \in \mathcal{Z}_{\mathrm{mt}}$.*

This is consistent with the assumption that the features determining the labels y_{sp} and y_{mt} are distinct (Park et al., 2023), and the empirical observation that high-level concepts are distinctly represented in the embeddings (Kirichenko et al., 2023). We emphasize that orthogonality does not imply independence between main-task and spurious features, as assumed in earlier work (Chen et al., 2020).

An alternative perspective is that $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ are not necessarily orthogonal, but that we aim to identify subspaces of $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ that are orthogonal to each other and informative about the respective labels. If $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ are high-dimensional (applicable in most realistic settings), there are enough degrees of freedom for these subspaces to cover significant parts of $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ in terms of their ability to predict y_{sp} and y_{mt}. We illustrate the effect of the orthogonality assumption in Figure 2 (panels A and D) for Toy data, and analyse how JSE behaves when the assumption does not hold in Appendix B.

We thus simultaneously perform a logistic regression on the embeddings $\boldsymbol{z}$ for y_{sp} and y_{mt}, subject to the constraint of orthogonality of $\boldsymbol{w}_{\text{sp}}$ and $\boldsymbol{w}_{\text{mt}}$. This means that for a sample $\{y_{\text{mt},i}, y_{\text{sp},i}, \boldsymbol{z}_i\}_{i=1}^{n}$ the estimates $\hat{\boldsymbol{w}}_{\text{sp}}, \hat{\boldsymbol{w}}_{\text{mt}}, \hat{b}_{\text{sp}}, \hat{b}_{\text{mt}}$ are obtained by performing the following optimization:

$$\underset{\substack{\boldsymbol{w}_{\text{sp}},\boldsymbol{w}_{\text{mt}},b_{\text{sp}},b_{\text{mt}} \\ (\boldsymbol{w}_{\text{sp}} \perp \boldsymbol{w}_{\text{mt}})}}{\arg\min} \sum_{i=1}^{n} \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{sp},i}, y_{\text{sp},i}) + \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{mt},i}, y_{\text{mt},i}), \tag{2}$$

where $\mathcal{L}_{\text{BCE}}$ is the binary cross-entropy (BCE). Furthermore, $\hat{y}_{\text{sp},i} = \text{Logit}^{-1}\left(\boldsymbol{z}_i^{\top}\boldsymbol{w}_{\text{sp}} + b_{\text{sp}}\right)$, and similarly for $\hat{y}_{\text{mt},i}$. The estimated spurious and main-task concept vectors are then $\hat{\boldsymbol{v}}_{\text{sp}} = \hat{\boldsymbol{w}}_{\text{sp}}/||\hat{\boldsymbol{w}}_{\text{sp}}||$ and $\hat{\boldsymbol{v}}_{\text{mt}} = \hat{\boldsymbol{w}}_{\text{mt}}/||\hat{\boldsymbol{w}}_{\text{mt}}||$.

3.2. Iteratively Estimating Multiple Concept and Main-task Vectors

The subspaces $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ will generally not be one-dimensional. Thus, the estimated spurious concept vector $\hat{\boldsymbol{v}}_{\text{sp}}$ could still contain main-task components (and vice versa for $\hat{\boldsymbol{v}}_{\text{mt}}$). To address this, we propose an iterative procedure to estimate the orthonormal bases of $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$, which are guaranteed to be orthogonal to each other.

For now, let us focus on estimating a vector $\boldsymbol{v}_{\text{sp}} \in \mathcal{Z}_{\text{sp}}$. By applying the procedure of Equation 2 gives a $\hat{\boldsymbol{v}}_{\text{sp}}$ and $\hat{\boldsymbol{v}}_{\text{mt}}$, where $\hat{\boldsymbol{v}}_{\text{sp}}$ may still have components in $\mathcal{Z}_{\text{mt}}$ and $\hat{\boldsymbol{v}}_{\text{mt}}$ may have components in $\mathcal{Z}_{\text{sp}}$. We propose to project out the direction $\hat{\boldsymbol{v}}_{\text{mt}}$ from the embeddings and to repeat the optimization of Equation 2 for the resulting subspace. Doing this multiple times will eventually remove all main-task information, guaranteeing that the estimated vector $\hat{\boldsymbol{v}}_{\text{sp},1}$ is orthogonal to the (estimated) main-task subspace.

By projecting out $\hat{\boldsymbol{v}}_{\text{sp},1}$ from the embeddings $\boldsymbol{z}$ and repeating the whole procedure d_{sp} times, we estimate an orthonormal basis $\hat{\boldsymbol{v}}_{\text{sp},1}, \hat{\boldsymbol{v}}_{\text{sp},2}, \ldots, \hat{\boldsymbol{v}}_{\text{sp},d_{\text{sp}}}$ of $\mathcal{Z}_{\text{sp}}$ that is orthogonal to the (main-task) subspace. The method described is a nested for-loop (see Algorithm 1), where the inner loop finds and projects out main-task vectors, and the outer loop finds and projects out spurious vectors. After having found a basis of $\mathcal{Z}_{\text{sp}}$, one can repeat the inner loop one last time to find $d_{\text{mt}} = \dim(\mathcal{Z}_{\text{mt}})$ vectors $\hat{\boldsymbol{v}}_{\text{mt},1}, \hat{\boldsymbol{v}}_{\text{mt},2}, \ldots, \hat{\boldsymbol{v}}_{\text{mt},d_{\text{mt}}}$ constituting an estimated basis for $\mathcal{Z}_{\text{mt}}$. The computational cost of the double for-loop is limited, as the number of lightweight optimizations (Equation 2) is at most $d_{\text{mt}} \times d_{\text{sp}}$. For a detailed description of the algorithm, see Appendix C.

Algorithm 1 JSE algorithm to estimate orthonormal bases for $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$. The conditions in the **if**-statements are discussed in Section 3.3.

```
Input: a sample $\{y_{\text{mt},k}, y_{\text{sp},k}, \boldsymbol{z}_k\}_{k=1}^{n}$ consisting of two binary labels and a vector $\boldsymbol{z}_k \in \mathbb{R}^d$.
Initialize embedding matrix $\boldsymbol{Z} = (\boldsymbol{z}_1\, \boldsymbol{z}_2\, \cdots\, \boldsymbol{z}_n)^{\top}$.
Initialize $\boldsymbol{Z}_{\text{sp}}^{\perp} \leftarrow \boldsymbol{Z}$.
for $i = 1, ..., d$ do
  $\boldsymbol{Z}_{\text{remain}} \leftarrow \boldsymbol{Z}_{\text{sp}}^{\perp}$
  for $j = 1, ..., d$ do
    Estimate $\hat{\boldsymbol{w}}_{\text{sp}}$, $\hat{\boldsymbol{w}}_{\text{mt}}$ with Equation 2 (use $\boldsymbol{Z}_{\text{remain}}$).
    $\hat{\boldsymbol{v}}_{\text{sp},i} \leftarrow \hat{\boldsymbol{w}}_{\text{sp}}/||\hat{\boldsymbol{w}}_{\text{sp}}||$ and $\hat{\boldsymbol{v}}_{\text{mt},j} \leftarrow \hat{\boldsymbol{w}}_{\text{mt}}/||\hat{\boldsymbol{w}}_{\text{mt}}||$.
    if $\hat{\boldsymbol{v}}_{\text{mt},j}$ is a proper main-task direction then
      Projection $\boldsymbol{Z}_{\text{remain}} \leftarrow \boldsymbol{Z}_{\text{sp}}^{\perp}(\boldsymbol{I} - \hat{\boldsymbol{V}}_{\text{mt}}\hat{\boldsymbol{V}}_{\text{mt}}^{\top})$, with
      $\hat{\boldsymbol{V}}_{\text{mt}} = (\hat{\boldsymbol{v}}_{\text{mt},1}\, \hat{\boldsymbol{v}}_{\text{mt},2}\, \cdots\, \hat{\boldsymbol{v}}_{\text{mt},j})$.
    else
      break
    end if
  end for
  if $\hat{\boldsymbol{v}}_{\text{sp},i}$ is a proper spurious direction then
    Projection $\boldsymbol{Z}_{\text{sp}}^{\perp} \leftarrow \boldsymbol{Z}(\boldsymbol{I} - \hat{\boldsymbol{V}}_{\text{sp}}\hat{\boldsymbol{V}}_{\text{sp}}^{\top})$, where
    $\hat{\boldsymbol{V}}_{\text{sp}} = (\hat{\boldsymbol{v}}_{\text{sp},1}\, \hat{\boldsymbol{v}}_{\text{sp},2}\, \cdots\, \hat{\boldsymbol{v}}_{\text{sp},i})$.
    $\hat{\boldsymbol{v}}'_{\text{mt},\ell} \leftarrow \hat{\boldsymbol{v}}_{\text{mt},\ell}$, for $\ell = 1, 2, \ldots, \ell_0$, with $\ell_0 = j$.
  else
    break
  end if
end for
return Bases $\{\hat{\boldsymbol{v}}_{\text{sp},m}\}_{m=1}^{i-1}$ and $\{\hat{\boldsymbol{v}}'_{\text{mt},\ell}\}_{\ell=1}^{\ell_0-1}$.
```

So far, we have treated the subspaces $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ equally. This symmetry is broken in Algorithm 1, as the main-task directions are identified in the inner loop and the concept directions in the outer loop. In Appendix C we give empirical evidence that swapping the loops has little effect on the outcome of the JSE method.

One can remove the spurious concept by projecting the embeddings $\boldsymbol{z}$ on the orthogonal complement of the spurious concept subspace, $\mathcal{Z}_{\text{sp}}^{\perp}$. The transformed embeddings can then be used for main-task prediction by a linear classifier. In Appendix C, we show that training the linear classifier on $\mathcal{Z}_{\text{mt}}$ instead of $\mathcal{Z}_{\text{sp}}^{\perp}$ gives similar performance.

3.3. Testing when to Stop Adding Vectors

So far, in the description of the iterative algorithm we have assumed the dimensions d_{sp} and d_{mt} of the respective subspaces $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ to be known. In practice, the dimen-

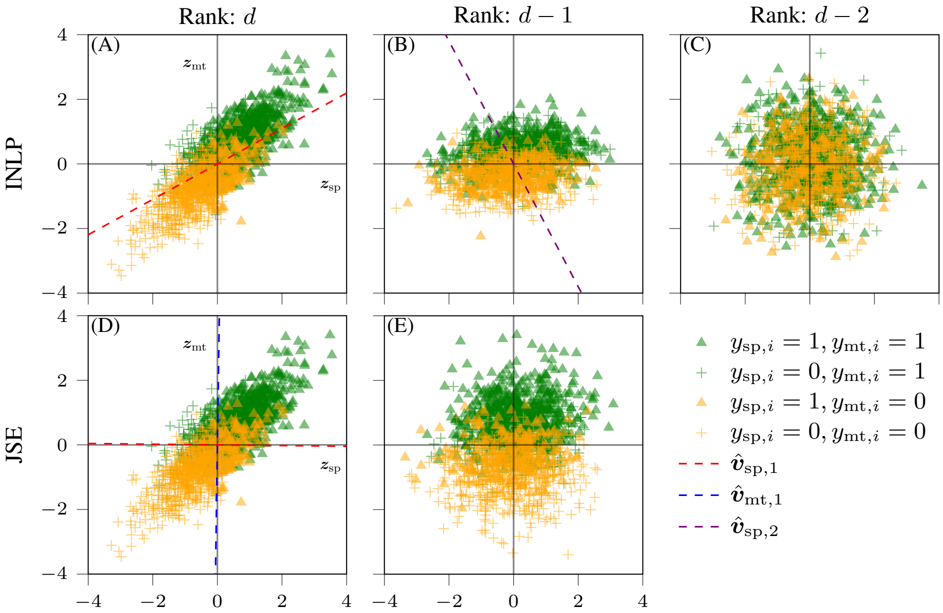

Figure 2. **Illustration of JSE, in comparison to INLP**: based on the $d(=20)$-dimensional Toy dataset (see Section 4.1) with $\rho = 0.8$ and sample size n =2,000. Two-dimensional slices of $\boldsymbol{z}$ are shown. Panels A and D have the spurious feature on the x-axis and the main-task feature on the y-axis. The remaining panels show the axes that best separate the main-task labels. JSE identifies a single spurious vector (panel D) and the remaining class separation is attributed to the main-task concept (panel E). INLP identifies (superpositions of) the main-task and spurious directions as spurious (panels A and B), and the main-task information is removed (panel C).

sions must be estimated via stopping criteria of the (nested) **for** loops in Algorithm 1. Let us focus on the condition in the outer loop. The condition in the inner loop is, *mutatis mutandis*, the same. For a given (normalized) direction $\boldsymbol{v}_{\text{sp}} \in \mathbb{R}^d$ in the embedding space, the statement "$\boldsymbol{v}_{\text{sp}}$ is a proper spurious direction" means that two criteria are met:

1. **The direction $\boldsymbol{v}_{\text{sp}}$ is informative about the spurious label** y_{sp}, meaning that the embeddings projected onto $\boldsymbol{v}_{\text{sp}}$ are able to predict the spurious label. To be concrete, a logistic regression based on the projected embeddings should have a higher accuracy than a classifier that just predicts the majority class, which we refer to as a 'random classifier'.

2. **The direction $\boldsymbol{v}_{\text{sp}}$ should be more predictive of the spurious concept than of the main-task concept.** Due to the spurious correlation, a vector in $\mathcal{Z}_{\text{sp}}$ is likely also predictive for the main-task concept. We nonetheless associate it with the spurious subspace $\mathcal{Z}_{\text{sp}}$, as long as its prediction accuracy for the spurious label is higher than for the main-task label.

Note that the first criterion is already used in Ravfogel et al. (2020), while the second is novel and addresses the problem of inadvertently removing main-task information in existing concept-removal methods. This is illustrated in Figure 2 (panel B), where the INLP-method of Ravfogel et al. (2020) removes a feature that is more predictive of the main-task concept than the spurious concept.

To make these criteria operational, we introduce two statistical tests in terms of differences between BCE's. For the first criterion we compare the BCE of $\hat{y}_{\text{sp}}^{(\boldsymbol{v}_{\text{sp}})} = \text{Logit}^{-1}\left(\gamma_{\text{sp}}\boldsymbol{z}^\top\boldsymbol{v}_{\text{sp}} + b_{\text{sp}}\right)$, which is a predictor for the label y_{sp} based on the embeddings projected onto $\boldsymbol{v}_{\text{sp}}$, and the BCE of a majority-rule 'random classifier'. The model parameters γ_{sp} and b_{sp} will be trained by minimizing the BCE. For the second criterion we compare the BCE of $\hat{y}_{\text{sp}}^{(\boldsymbol{v}_{\text{sp}})}$ with the analogously defined $\hat{y}_{\text{mt}}^{(\boldsymbol{v}_{\text{sp}})}$, which is a predictor of y_{mt}. Both tests are performed using a t-statistic, using a weighted average of the BCE's over the four combinations of y_{sp} and y_{mt}. For a precise definition of the hypotheses, test statistics, and their properties, see Appendix D.

4. Experiments

This section gives experimental evidence that JSE is able to identify the main-task and spurious concepts, and that in these respects JSE outperforms existing concept-removal methods. Since for realistic datasets there exists no ground truth about how main-task and spurious concepts are represented in the last-layer embeddings, we mostly need to rely on proxy experiments. A first experiment is the problem of OOD generalization (see Section 4.2), where models are trained on data with a spurious correlation, and tested on data without. A good performance of a model after concept removal indicates that the spurious concept was removed adequately, while main-task features have remained intact.

We proceed with an experiment on Toy data in Section 4.3, for which we know the true main-task and spurious features. To further substantiate our claims, we present Grad-CAM results in Section 4.4 and conduct experiments similar to Ravfogel et al. (2020) and Kumar et al. (2022) to test the validity of concept-removal methods, in Sections 4.5 and 4.6 respectively. In this results section, we compare JSE with other last-layer concept-removal methods mentioned in Section 2.1: iterative null-space projection (INLP), relaxed linear adversarial concept erasure (RLACE), adversarial removal based on a single linear adversary (ADV) and least-squares concept erasure (LEACE). Details about the datasets, experimental setup and parameter selection are in Appendix I. For numerical details, see Appendix A.

4.1. Datasets

JSE and existing concept-removal methods are applied to a Toy dataset, for which knowledge about the true main-task and spurious features is available, as well as benchmark datasets from computer vision (Waterbirds, CelebA) and natural language processing (MultiNLI). A brief description of the datasets and how we use them is given here. See Appendix I.1 for detailed information.

Toy data: We create d-dimensional embeddings drawn from a multivariate normal distribution with a block correlation matrix, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu} = \boldsymbol{0}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{\text{sp,mt}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{\Sigma}_{\text{sp,mt}} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}.$$

Note that the embeddings here are not neural network representations of underlying input features. We set $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ to be one-dimensional subspaces, with direction vectors $\boldsymbol{w}_{\text{sp}} = (\gamma_{\text{sp}}, 0, 0, \dots, 0)^\top$ and $\boldsymbol{w}_{\text{mt}} = (0, \gamma_{\text{mt}}, 0, \dots, 0)^\top$, respectively. We define binary labels y_{sp} and y_{mt},

$$p(y_{\text{sp}} = 1|\boldsymbol{z}) = \text{Logit}^{-1}\left(\boldsymbol{z}^\top \boldsymbol{w}_{\text{sp}} + b_{\text{sp}}\right),$$

and likewise for $p(y_{\text{mt}} = 1|\boldsymbol{z})$. Throughout the simulations we take $d = 20$, $b_{\text{sp}} = b_{\text{mt}} = 0$ and $\gamma_{\text{sp}} = \gamma_{\text{mt}} = 3$. Parameter ρ is the correlation between the spurious and main-task features, determining the spurious relation $p_{\text{train}}(y_{\text{mt}}|\boldsymbol{z}_{\text{sp}})$ between the main-task label and the spurious feature.

Vision: We use two common datasets containing a spurious correlation. The first is the Waterbirds dataset (Sagawa et al., 2020a), where the main-task concept is bird type (waterbird vs. landbird) and the spurious concept is background (water vs. land). The second is the CelebA dataset, where the main-task concept is hair color (blond vs. non-blond) and the spurious concept is sex (female vs. male). We use a pre-trained ResNet50 architecture (He et al., 2016), which is then finetuned on the respective dataset, after which the concept removal is applied to the last layer.

NLP: We use the MultiNLI dataset (Williams et al., 2018), which contains pairs of sentences. The main-task concept is, whether or not the first sentence contradicts the second. Following an experiment from Joshi et al. (2022), we use as spurious concept the presence or absence of punctuation marks ('!!') at the end of the second sentence. For exemplary pairs of sentences, see Appendix I.1. Each run in our experiments starts with finetuning a BERT model, after which concept removal is applied to the [CLS] embeddings.

For the vision and NLP datasets, we cannot directly control the strength of the spurious relation between the main-task label and the spurious features, $p_{\text{train}}(y_{\text{mt}}|\boldsymbol{z}_{\text{sp}})$. We therefore use $p_{\text{train}}(y_{\text{mt}} = y|y_{\text{sp}} = y)$, with $y \in \{0, 1\}$, as a proxy. To increase the precision of our method and for computational efficiency, we reduce the dimension of the last-layer embeddings to $d = 300$ for Waterbirds and CelebA, and $d = 100$ for multiNLI via Principal Component Analysis (PCA).

4.2. Out-of-distribution (OOD) Generalization

We consider the problem of OOD generalization, where the training and OOD test data have a different dependence between main-task and spurious features (i.e. $p_{\text{train}}(\boldsymbol{x}_{\text{mt}}, \boldsymbol{x}_{\text{sp}}) \neq p_{\text{OOD}}(\boldsymbol{x}_{\text{mt}}, \boldsymbol{x}_{\text{sp}})$). This generally leads to $p_{\text{train}}(y_{\text{mt}}|\boldsymbol{x}_{\text{sp}}) \neq p_{\text{OOD}}(y_{\text{mt}}|\boldsymbol{x}_{\text{sp}})$, with similar discrepancies at the level of the embeddings. A DNN using the spurious features to predict the main-task label will see performance loss when applied to the OOD data. Similarly, if a concept-removal method has inadvertently also removed main-task features, the resulting DNN will also observe a deterioration in performance.

Figure 3 shows the results of applying different concept-removal methods to the problem of OOD generalization. JSE outperforms the other methods for the Toy and vision datasets, especially when the spurious correlation is strong. When applied to the text dataset, it performs similar to the best other concept-removal methods (LEACE and RLACE). We suspect that during finetuning of BERT the main-task and spurious concepts become overlapping in the [CLS] embeddings, in line with previous work by Dalvi et al. (2022).

The performance loss of INLP, RLACE and LEACE is explained by the fact that they also inadvertently remove main-task features. This behaviour is illustrated for the Toy dataset in Figure 2. For ADV the performance deteriorates because the spurious features remain present after the training procedure, in line with previous work (Belinkov, 2022; Ravfogel et al., 2022a). JSE also shows performance loss for the Toy and vision datasets, albeit to

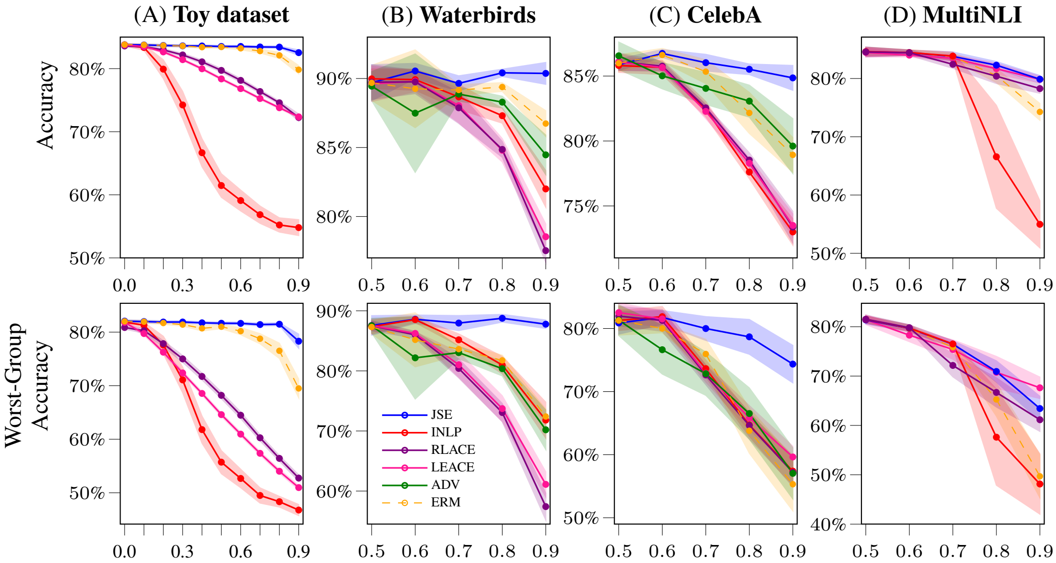

Figure 3. **OOD generalization, compared to other concept-removal methods**: We plot the (worst-group) accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training set (ρ for the Toy dataset, $p_{\text{train}}(y_{\text{mt}} = y|y_{\text{sp}} = y)$ for the other datasets). Averages based on 100, 5, 5 and 5 runs, respectively. The shaded area reflects the 95% confidence interval.

a lesser extent. As the spurious correlation increases, we suspect our method becomes more sensitive to finite-sample estimation noise. We illustrate this further in Appendix B.2.

As an additional benefit, JSE is the first concept-removal method that is (for all datasets) competitive with several instance-reweighting techniques such as group-weighted ERM, group-distributional robust optimization (GDRO), and just train twice (JTT). For an overview of these techniques and a comparison with JSE, see Appendix G. We emphasise that this class of techniques is purely aimed at OOD generalization, while the primary focus of JSE is to improve the interpretability of the model.

4.3. Regression Analysis for Toy Data

Because we have access to the ground truth for the Toy dataset, we can quantify whether spurious features are removed and main-task features are preserved. Let $\tilde{z}$ denote the embedding vectors after being transformed by a concept-removal method. We aim to find linear models that use $\tilde{z}$ to predict z_{mt} or z_{sp}. Estimation is conducted via ordinary least squares (OLS) on a test set without correlation between z_{mt} and z_{sp}, to prevent the regression from predicting one with the other. A good concept-removal method creates embeddings $\tilde{z}$ that can reconstruct z_{mt} (low MSE) and cannot reconstruct z_{sp} (high MSE). Figure 4 shows that JSE preserves the main-task feature, while other methods remove (part of) it. The spurious feature is removed to a similar extent by JSE, INLP and RLACE. Interestingly, while LEACE prevents the prediction of y_{sp} with $\tilde{z}$, it appears that information related to z_{sp} remains present.

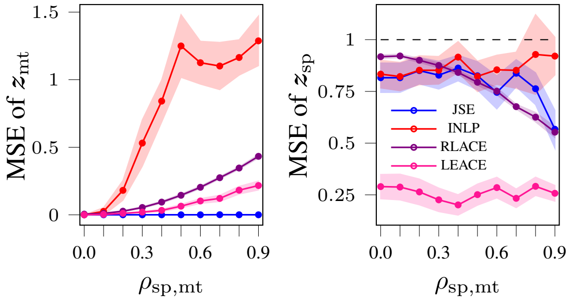

Figure 4. **Ability to reconstruct main-task and spurious concept features after concept removal.** We show the mean-squared error (MSE) of predicting $z_{\text{mt}}, z_{\text{sp}}$ via OLS on the transformed embeddings. The dotted line on the right plot indicates the MSE when there is no information of z_{sp} left. Averages based on 100 runs, and shaded area reflects the 95% confidence interval.

4.4. Grad-CAM for Waterbirds

To obtain insight into the workings of the concept-removal methods on the Waterbirds dataset, we use Grad-CAM (Selvaraju et al., 2017) to highlight parts of the images used by the models (after concept removal). Figure 5 shows the results (see Appendix E for more images and concept-removal methods). JSE produces a model that relies predominantly on the bird features and neglects the background. ERM and INLP use the background for their correct and incorrect predictions. Interestingly, INLP, LEACE and RLACE perform much worse on images that appear more frequently in the training set (e.g. landbirds on land) than on images from minority groups (e.g. landbirds on water). We observe a similar pattern for the other datasets. We posit that this is because features get mixed, as described by Kumar et al. (2022), leading INLP, LEACE and RLACE to associate spurious features with the main-task concept. This is visible in Figure 5, where after INLP the model classifies a waterbird as a landbird by using the water background.

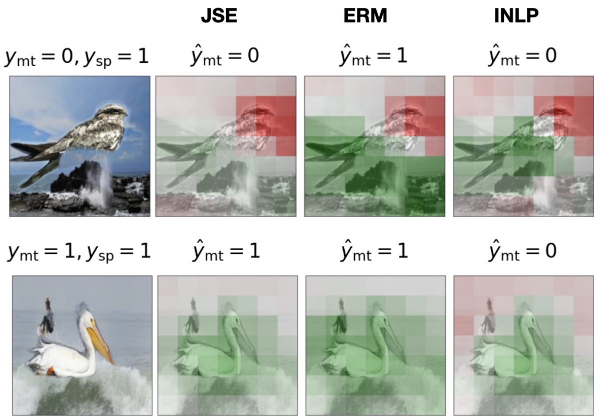
JSE
ERM
INLP
$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 1$
$\hat{y}_{\mathrm{mt}} = 0$
$\hat{y}_{\mathrm{mt}} = 1$
$\hat{y}_{\mathrm{mt}} = 0$
$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 1$
$\hat{y}_{\mathrm{mt}} = 1$
$\hat{y}_{\mathrm{mt}} = 1$
$\hat{y}_{\mathrm{mt}} = 0$

Figure 5. **Grad-CAM for the last layer of Resnet-50 predicting the main-task label**: Red (green) patches indicate a contribution towards a prediction $y_{\mathrm{mt}} = 0$ ($y_{\mathrm{mt}} = 1$).

4.5. Removing Concepts from Original Images of CelebA

We conduct an experiment for the CelebA dataset that is similar to one from Ravfogel et al. (2022a). The goal is to qualitatively show what features are removed by a concept-removal method. Instead of working with the embeddings, we downscale the images to 50 by 50 grey-scale images, flatten them to a 2,500-dimensional vector, and apply concept-removal methods to the raw pixels. The main-task concept is the presence of 'glasses', while the spurious concept is 'smiling'. We set the spurious correlation at $p(y_{\mathrm{mt}} = y | y_{\mathrm{sp}} = y) = 0.8$. The results are shown in Figure 6, with more images and methods in Appendix F. Other concept-removal methods change pixels that relate to both the smile and the glasses. JSE is the only method that primarily focuses on the smile.

4.6. Preserving Main-task Features in Text Data

We use the MultiNLI dataset to perform an experiment similar to one from Kumar et al. (2022). First, we train a BERT model on a subset of the data with only one value of the spurious concept label (we choose $y_{\mathrm{sp}} = 0$, meaning there is no '!!' at the end of the second sentence). We refer to this as a 'clean' BERT model. According to Ravichander et al. (2021), a model trained in this manner should not use spurious features. As a consequence, applying a concept-removal method to a clean BERT model should not affect the main-task classification. We test this by training concept-removal methods on data with both spurious labels present and measure how well they generalize OOD.

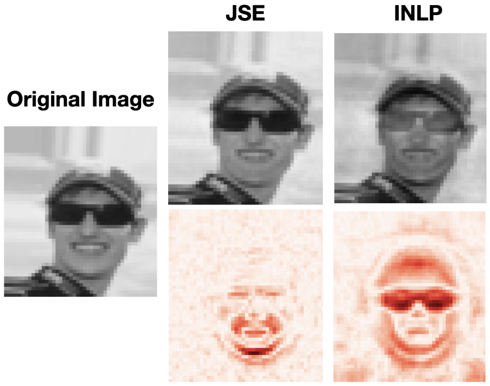

Figure 6. **Application of concept-removal methods to raw pixel data**: The first row shows the image after it is transformed by the concept-removal method. The second row shows the absolute difference between the transformed and original image to indicate which pixels have been changed.

The results of this experiment in Figure 7 show that only JSE is applied without performance loss, with respect to the clean BERT model (ERM). This indicates that it has been able to remove features related to the spurious concept label, without removing main-task features. Similar to Kumar et al. (2022), for other concept-removal methods, the main-task features are removed to a greater extent as the spurious correlation increases, affecting main-task performance.

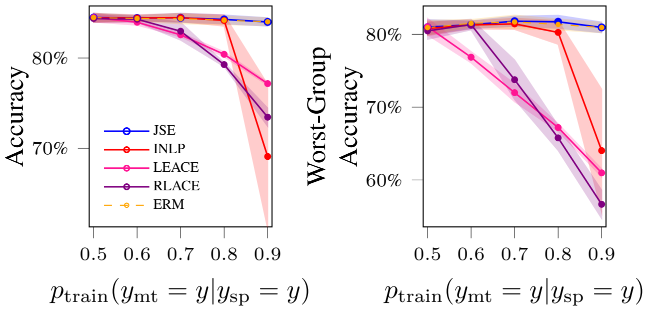

Figure 7. **Result of applying concept removal methods to a 'clean' BERT model**: We plot the (worst-group) accuracy on a test set without spurious correlation, as a function of the spurious correlation in the dataset that was used for the concept removal method. The BERT model is finetuned on a dataset without variation in the spurious concept ($y_{\mathrm{sp}} = 0$). Averages are based on 5 runs, and the shaded area reflects the 95% confidence interval.

5. Conclusion and Discussion

This paper has introduced and empirically tested joint subspace estimation (JSE), a novel post-hoc concept-removal method that improves the interpretability and control of neural network representations in the presence of spurious correlations. JSE outperforms existing concept-removal methods, despite making assumptions (linearity, orthogonality) about the structure of the embedding space.

Future work will be needed to develop tests for these assumptions, or see if they can be relaxed. One example is jointly estimating subspaces based on their non-linear relationship with the spurious and main-task labels, as done by Ravfogel et al. (2022b). Our results highlight the difficulty of separating different concepts in BERT's [CLS] embeddings. This underlines the need to better disentangle concepts in embeddings of large language models, for instance through different training procedures, as done by Zhang et al. (2021).

Impact Statement

Our work has the potential to give individuals greater control over which features are used by DNNs. This potentially has societal consequences: ensuring that a DNN uses the correct or desirable features is crucial before we can apply it in high-stake domains such as medicine or finance (Rudin, 2019). However, as highlighted in the paper, our work is not without limitations. Care should be taken to measure the effectiveness of the approach in removing spurious concepts in the context in which it is to be deployed.

Acknowledgements

We are very grateful to participants at various seminars, including the Tinbergen Institute (TI) Econometrics Seminar, the Center for Economics and Non-linear Dynamics in Economics and Finance (CENDEF) seminar at the Amsterdam School of Economics, the Business Analytics (BA) seminar of the Amsterdam Business school, as well as the Netherlands Econometrics Study Group (NESG) 2024, for their insightful comments and feedback.

References

Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.-C., Bengio, Y., Mitliagkas, I., and Rish, I. Invariance principle meets information bottleneck for out-of-distribution generalization. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 3438–3450, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/1c336b8080f82bcc2cd2499b4c57261d-Paper.pdf.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *Arxiv Computing Research Repository (CoRR)*, abs/1907.02893, 2019. URL https://arxiv.org/abs/1907.02893.

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3319–3327, 2017. doi: 10.1109/CVPR.2017.354.

Belinkov, Y. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Belrose, N., Schneider-Joseph, D., Ravfogel, S., Cotterell, R., Raff, E., and Biderman, S. Leace: Perfect linear concept erasure in closed form. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 66044–66063. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d066d21c619d0a78c5b557fa3291a8f4-Paper-Conference.pdf.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

Chen, Z., Bei, Y., and Rudin, C. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. doi: 10.1038/s42256-020-00265-z. URL https://doi.org/10.1038/s42256-020-00265-z.

Dalvi, F., Khan, A. R., Alam, F., Durrani, N., Xu, J., and Sajjad, H. Discovering latent concepts learned in BERT. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=POTMtpYI1xH.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., McLean, C., Mincu, D., Mitani, A., Montanari, A., Nado, Z., Natarajan, V., Nielson, C., Osborne, T. F., Raman, R., Ramasamy, K., Sayres, R., Schrouff, J., Seneviratne, M., Sequeira, S., Suresh, H., Veitch, V., Vladymyrov, M., Wang, X., Webster, K., Yadlowsky, S., Yun, T., Zhai, X., and Sculley, D. Underspecification presents challenges for credibility in modern machine learning. *J. Mach. Learn. Res.*, 23(1), jan 2022. ISSN 1532-4435.

Dev, S. and Phillips, J. Attenuating bias in word vectors. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 879–887. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/dev19a.html.

Dev, S., Li, T., Phillips, J. M., and Srikumar, V. OSCaR: Orthogonal subspace correction and rectification of biases in word embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5034–5050, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.411. URL https://aclanthology.org/2021.emnlp-main.411.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Edwards, H. and Storkey, A. Censoring representations with an adversary. In *International Conference in Learning Representations (ICLR2016)*, pp. 1–14, May 2016. URL https://iclr.cc/archive/www/doku.php%3Fid=iclr2016:main.html. 4th International Conference on Learning Representations, ICLR 2016.

Elazar, Y. and Goldberg, Y. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 11–21, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1002. URL https://aclanthology.org/D18-1002.

Elazar, Y., Ravfogel, S., Jacovi, A., and Goldberg, Y. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 03 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00359. URL https://doi.org/10.1162/tacl_a_00359.

Ethayarajh, K., Duvenaud, D., and Hirst, G. Understanding undesirable word embedding associations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1696–1705, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1166. URL https://aclanthology.org/P19-1166.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ganin15.html.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Hermann, K., Chen, T., and Kornblith, S. The origins and prevalence of texture bias in convolutional neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19000–19015. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/db5f9f42a7157abe65bb145000b5871a-Paper.pdf.

Idrissi, B. Y., Arjovsky, M., Pezeshki, M., and Lopez-Paz, D. Simple data balancing achieves competitive worst-group-accuracy. In *First Conference on Causal Learning and Reasoning*, 2022. URL https://openreview.net/forum?id=cDxT7WYhaD.

Izmailov, P., Kirichenko, P., Gruver, N., and Wilson, A. G. On feature learning in the presence of spurious correlations. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 38516–38532. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/fb64a552feda3d981dbe43527a80a07e-Paper-Conference.pdf.

Joshi, N., Pan, X., and He, H. Are all spurious features in natural language alike? an analysis through a causal lens. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9804–9817, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.666. URL https://aclanthology.org/2022.emnlp-main.666.

Kaushik, D. and Lipton, Z. C. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 5010–5015, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1546. URL https://aclanthology.org/D18-1546.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kim18d.html.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kirichenko, P., Izmailov, P., and Wilson, A. G. Last layer re-training is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=Zb6c8A-Fghk.

Kumar, A., Tan, C., and Sharma, A. Probing classifiers are unreliable for concept removal and detection. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17994–18008. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/725f5e8036cc08adeba4a7c3bcbc6f2c-Paper-Conference.pdf.

Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. Just train twice: Improving group robustness without training group information. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6781–6792. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

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

Niven, T. and Kao, H.-Y. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4658–4664, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1459. URL https://aclanthology.org/P19-1459.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. *Arxiv Computing Research Repository (CoRR)*, 2023. URL https://arxiv.org/abs/2311.03658.

Pezeshki, M., Kaba, O., Bengio, Y., Courville, A. C., Precup, D., and Lajoie, G. Gradient starvation: A learning proclivity in neural networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1256–1272. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/

0987b8b338d6c90bbedd8631bc499221-Paper.pdf.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M., and Goldberg, Y. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7237–7256, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.647. URL https://aclanthology.org/2020.acl-main.647.

Ravfogel, S., Prasad, G., Linzen, T., and Goldberg, Y. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In Bisazza, A. and Abend, O. (eds.), *Proceedings of the 25th Conference on Computational Natural Language Learning*, pp. 194–209, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.conll-1.15. URL https://aclanthology.org/2021.conll-1.15.

Ravfogel, S., Twiton, M., Goldberg, Y., and Cotterell, R. D. Linear adversarial concept erasure. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18400–18421. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/ravfogel22a.html.

Ravfogel, S., Vargas, F., Goldberg, Y., and Cotterell, R. Adversarial concept erasure in kernel space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6034–6055, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.405.

Ravfogel, S., Goldberg, Y., and Cotterell, R. Log-linear guardedness and its implications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9413–9431, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.523. URL https://aclanthology.org/2023.acl-long.523.

Ravichander, A., Belinkov, Y., and Hovy, E. Probing the probing paradigm: Does probing accuracy entail task relevance? In Merlo, P., Tiedemann, J., and Tsarfaty, R. (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3363–3377, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295.

Rényi, A. On the asymptotic distribution of the sum of a random number of independent random variables. *Acta Math*, 8:193–199, 1957.

Rosenfeld, E., Ravikumar, P. K., and Risteski, A. Domain-adjusted regression or: ERM may already learn features sufficient for out-of-distribution generalization. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. URL https://openreview.net/forum?id=Ypo0AckYW8.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16(none):1 – 85, 2022. doi: 10.1214/21-SS133. URL https://doi.org/10.1214/21-SS133.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020a.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020b.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Singla, S. and Feizi, S. Salient imagenet: How to discover spurious features in deep learning? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=XVPqLyNxSyh.

Srivastava, M., Hashimoto, T., and Liang, P. Robustness to spurious correlations via human annotations. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9109–9119. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/srivastava20a.html.

Vargas, F. and Cotterell, R. Exploring the linear subspace hypothesis in gender bias mitigation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2902–2913, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.232. URL https://aclanthology.org/2020.emnlp-main.232.

Wang, L., Yan, Y., He, K., Wu, Y., and Xu, W. Dynamically disentangling social bias from task-oriented representations with adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3740–3750, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.293. URL https://aclanthology.org/2021.naacl-main.293.

Wang, T., Zhao, J., Yatskar, M., Chang, K.-W., and Ordonez, V. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5309–5318, 2019. doi: 10.1109/ICCV.2019.00541.

Wang, Z. and Culotta, A. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3431–3440, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.308. URL https://aclanthology.org/2020.findings-emnlp.308.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010. URL https://www.vision.caltech.edu/datasets/cub_200_2011/.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Xiao, K. Y., Engstrom, L., Ilyas, A., and Madry, A. Noise or signal: The role of image backgrounds in object recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=gl3D-xY7wLq.

Xu, K., Cao, T., Shah, S., Maung, C., and Schweitzer, H. Cleaning the null space: a privacy mechanism for predictors. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 2789–2795. AAAI Press, 2017.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pp. 335–340, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360128. doi: 10.1145/3278721.3278779. URL https://doi.org/10.1145/3278721.3278779.

Zhang, X., van de Meent, J.-W., and Wallace, B. Disentangling representations of text by masking transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 778–791, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.60. URL https://aclanthology.org/2021.emnlp-main.60.

Zhou, B., Khosla, A., Lapedriza, À., Torralba, A., and Oliva, A. Places: An image database for deep scene understanding. *Arxiv Computing Research Repository (CoRR)*, abs/1610.02055, 2016. URL http://arxiv.org/abs/1610.02055.

Zhou, C., Ma, X., Michel, P., and Neubig, G. Examining and combating spurious features under distribution shift. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12857–12867. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zhou21g.html.

Appendix Outline

In Section A, we provide details on the results in Section 4.2 for the four datasets (Waterbirds, CelebA, MultiNLI and Toy). Section B contains additional results for the Toy dataset. In Section C we present additional details on the implementation of the JSE algorithm. In Section D we lay out the testing procedure for the JSE algorithm. Section E and F provide additional details and images for the experiments conducted in Section 4.4 and 4.5 respectively. In Section G, we compare JSE to other instance-reweighting methods. In Section H we illustrate how JSE can be used to deal with multiple spurious concept labels. Finally, Section I provides a more detailed description of the datasets, as well as implementation details for the experiments.

A. Full Set of Results for Waterbirds, CelebA, MultiNLI and Toy Dataset

Table 1. **Results for the Waterbirds dataset**: Table shows the average, worst-group, and per-group accuracy on a test set where $p_{\text{OOD}}(y_{\text{mt}} = y \mid y_{\text{sp}} = y) = 0.5$, with $y \in \{0, 1\}$, as a function of $p_{\text{train}}(y_{\text{mt}} = y \mid y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. Standard error is reported between brackets.

		$p_{\text{train}}(y_{\text{mt}} = y \mid y_{\text{sp}} = y)$				
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9
JSE	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 0$	90.72 (0.83)	92.43 (0.44)	90.58 (0.27)	91.46 (0.76)	91.85 (0.75)
	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 1$	88.16 (1.08)	89.05 (0.50)	87.96 (0.64)	89.64 (0.50)	89.56 (0.76)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 0$	91.43 (0.24)	90.16 (0.46)	91.25 (0.28)	90.25 (0.41)	89.63 (0.81)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 1$	89.75 (0.57)	89.56 (0.33)	90.69 (0.32)	89.60 (0.53)	88.75 (0.66)
	Worst-group	87.57 (0.83)	88.60 (0.36)	87.96 (0.64)	88.76 (0.33)	87.77 (0.36)
	Average	89.70 (0.66)	90.55 (0.28)	89.64 (0.28)	90.41 (0.13)	90.37 (0.40)
ERM	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 0$	90.11 (0.72)	92.63 (1.70)	94.12 (0.28)	96.91 (0.06)	98.44 (0.16)
	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 1$	88.85 (1.16)	85.34 (2.35)	83.62 (0.51)	83.17 (0.38)	77.47 (1.44)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 0$	91.71 (0.23)	89.31 (0.51)	87.94 (0.42)	81.74 (0.45)	72.93 (0.87)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 1$	89.10 (0.64)	91.09 (0.84)	92.49 (0.21)	92.40 (0.22)	91.96 (0.43)
	Worst-group	87.31 (0.55)	85.17 (2.29)	83.62 (0.51)	81.74 (0.45)	72.40 (0.62)
	Average	89.69 (0.65)	89.25 (1.44)	89.17 (0.20)	89.38 (0.16)	86.73 (0.48)
INLP	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 0$	92.32 (0.49)	90.26 (0.60)	85.65 (0.91)	80.97 (0.73)	71.80 (1.52)
	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 1$	87.36 (0.63)	89.39 (0.62)	90.47 (0.49)	92.82 (0.31)	89.79 (0.40)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 0$	90.69 (0.35)	91.62 (0.32)	93.05 (0.25)	93.71 (0.17)	92.74 (0.46)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 1$	90.22 (0.30)	88.88 (0.35)	88.54 (0.52)	83.80 (0.40)	79.69 (0.84)
	Worst-group	87.36 (0.63)	88.53 (0.26)	85.17 (0.50)	80.97 (0.73)	71.80 (1.52)
	Average	89.98 (0.36)	89.92 (0.39)	88.66 (0.46)	87.30 (0.28)	82.00 (0.68)
LEACE	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 0$	89.32 (0.85)	87.35 (0.69)	81.67 (1.41)	73.72 (1.09)	61.11 (1.07)
	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 1$	89.43 (1.05)	92.12 (0.50)	93.29 (0.55)	94.08 (0.25)	91.14 (0.38)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 0$	92.12 (0.28)	92.96 (0.35)	94.74 (0.23)	94.95 (0.31)	95.42 (0.45)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 1$	88.72 (0.50)	87.07 (0.35)	85.23 (0.78)	80.97 (0.49)	78.57 (1.12)
	Worst-group	87.55 (0.33)	86.24 (0.14)	81.02 (1.06)	73.72 (1.09)	61.11 (1.07)
	Average	89.61 (0.65)	89.80 (0.38)	88.04 (0.63)	84.80 (0.43)	78.53 (0.47)
RLACE	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 0$	89.84 (0.79)	86.97 (0.88)	81.19 (1.36)	73.06 (0.77)	57.41 (1.14)
	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 1$	89.19 (1.07)	92.29 (0.40)	93.46 (0.44)	94.88 (0.22)	92.93 (0.27)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 0$	92.06 (0.25)	93.15 (0.36)	94.61 (0.29)	95.39 (0.23)	96.11 (0.24)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 1$	88.91 (0.58)	87.23 (0.23)	85.02 (0.83)	80.56 (0.54)	75.48 (0.63)
	Worst-group	87.51 (0.42)	86.00 (0.30)	80.41 (0.81)	73.06 (0.77)	57.41 (1.14)
	Average	89.73 (0.64)	89.75 (0.43)	87.88 (0.58)	84.86 (0.29)	77.53 (0.44)
ADV	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 0$	89.69 (0.55)	90.58 (2.55)	93.86 (0.29)	96.42 (0.18)	97.70 (0.30)
	$y_{\text{mt}} = 0$, $y_{\text{sp}} = 1$	88.55 (1.01)	82.17 (3.50)	83.04 (0.52)	80.38 (0.62)	70.89 (2.02)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 0$	91.93 (0.24)	90.53 (0.73)	88.10 (0.26)	82.80 (0.46)	76.76 (1.29)
	$y_{\text{mt}} = 1$, $y_{\text{sp}} = 1$	89.25 (0.49)	92.24 (0.74)	92.55 (0.23)	92.99 (0.32)	93.36 (0.53)
	Worst-group	87.53 (0.62)	82.17 (3.50)	83.04 (0.52)	80.37 (0.62)	70.19 (1.67)
	Average	89.44 (0.54)	87.49 (2.19)	88.87 (0.26)	88.29 (0.22)	84.47 (0.73)

Table 2. **Results for the CelebA dataset**: Table shows the average, worst-group, and per-group accuracy on a test set where $p_{\text{OOD}}(y_{\text{mt}} = y|y_{\text{sp}} = y) = 0.5$, with $y \in \{0, 1\}$, as a function of $p_{\text{train}}(y_{\text{mt}} = y|y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. Standard error is reported between brackets.

		$p_{\text{train}}(y_{\text{mt}} = y\|y_{\text{sp}} = y)$				
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9
JSE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.40 (0.35)	87.56 (0.71)	87.52 (0.36)	88.32 (1.57)	90.36 (0.43)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	82.72 (1.23)	82.24 (0.56)	81.12 (1.18)	82.56 (0.77)	81.84 (1.37)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.04 (0.70)	84.48 (1.23)	82.72 (1.10)	79.12 (1.74)	74.40 (1.55)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	92.32 (0.37)	92.68 (0.30)	92.76 (0.23)	92.08 (0.57)	92.84 (0.37)
	Worst-group	80.88 (0.90)	81.76 (0.56)	80.00 (0.96)	78.68 (1.41)	74.36 (1.51)
	Average	85.87 (0.23)	86.74 (0.15)	86.03 (0.33)	85.52 (0.21)	84.86 (0.49)
ERM	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.36 (0.56)	89.52 (0.90)	91.52 (0.43)	93.96 (0.66)	95.28 (0.35)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	82.80 (1.29)	80.76 (0.41)	78.40 (1.27)	75.00 (1.69)	68.88 (1.38)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.68 (1.10)	82.24 (1.31)	76.76 (1.29)	63.80 (1.82)	55.32 (2.15)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	92.12 (0.38)	93.96 (0.37)	94.68 (0.63)	95.88 (0.45)	96.24 (0.47)
	Worst-group	81.28 (1.18)	80.00 (0.34)	75.96 (1.19)	63.80 (1.82)	55.32 (2.15)
	Average	85.99 (0.33)	86.62 (0.26)	85.34 (0.47)	82.16 (0.81)	78.93 (0.68)
INLP	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	85.92 (0.70)	81.88 (0.82)	73.64 (0.69)	65.64 (1.48)	58.24 (1.40)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.08 (1.16)	85.28 (0.57)	85.88 (0.46)	87.96 (0.57)	87.48 (1.00)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.64 (0.98)	88.36 (1.18)	89.68 (0.58)	88.76 (0.67)	86.48 (0.96)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	91.60 (0.46)	87.16 (0.74)	80.60 (0.56)	68.08 (0.97)	59.80 (1.01)
	Worst-group	81.28 (0.81)	81.88 (0.82)	73.64 (0.69)	64.72 (0.92)	57.40 (1.26)
	Average	85.81 (0.27)	85.67 (0.28)	82.45 (0.30)	77.61 (0.26)	73.00 (0.52)
LEACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	85.48 (0.73)	81.52 (0.63)	72.92 (0.71)	65.84 (0.99)	59.64 (0.86)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.92 (0.92)	85.80 (0.62)	86.12 (0.70)	88.04 (0.29)	85.48 (1.11)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.60 (0.87)	88.80 (1.30)	89.48 (0.34)	87.64 (0.60)	85.68 (0.51)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	91.32 (0.45)	86.52 (0.69)	80.60 (0.85)	71.64 (1.68)	63.20 (0.89)
	Worst-group	82.52 (0.79)	81.52 (0.63)	72.92 (0.71)	65.84 (0.99)	59.64 (0.86)
	Average	86.08 (0.33)	85.66 (0.13)	82.28 (0.29)	78.29 (0.19)	73.50 (0.45)
RLACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	85.52 (0.77)	81.28 (0.87)	72.64 (0.78)	64.64 (1.09)	57.92 (1.29)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.68 (1.14)	85.84 (0.75)	86.36 (0.61)	89.36 (0.60)	88.12 (1.34)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.44 (0.86)	89.32 (1.22)	90.60 (0.57)	90.16 (0.46)	87.88 (1.28)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	91.76 (0.39)	86.60 (0.66)	80.60 (0.67)	69.96 (1.47)	59.56 (1.33)
	Worst-group	81.96 (0.78)	81.28 (0.87)	72.64 (0.78)	64.64 (1.09)	57.20 (1.03)
	Average	86.10 (0.34)	85.76 (0.19)	82.55 (0.11)	78.53 (0.29)	73.37 (0.62)
ADV	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	88.04 (0.54)	87.48 (0.63)	91.44 (0.15)	93.44 (0.64)	95.56 (0.23)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.72 (1.62)	82.16 (0.37)	77.92 (1.21)	76.36 (0.76)	69.44 (1.60)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.80 (2.06)	77.04 (2.29)	72.84 (1.75)	66.52 (2.08)	57.04 (2.06)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	91.68 (0.96)	93.40 (0.47)	94.00 (0.36)	95.96 (0.52)	96.40 (0.61)
	Worst-group	81.32 (1.24)	76.64 (1.97)	72.84 (1.75)	66.52 (2.08)	57.04 (2.06)
	Average	86.56 (0.54)	85.02 (0.51)	84.05 (0.63)	83.07 (0.63)	79.61 (1.08)

Table 3. **Results for the MultiNLI dataset**: Table shows the average, worst-group, and per-group accuracy on a test set where $p_{\text{OOD}}(y_{\text{mt}} = y|y_{\text{sp}} = y) = 0.5$, with $y \in \{0, 1\}$, as a function of $p_{\text{train}}(y_{\text{mt}} = y|y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. Standard error is reported between brackets.

		$p_{\text{train}}(y_{\text{mt}} = y\|y_{\text{sp}} = y)$				
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9
JSE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	87.49 (0.69)	88.83 (0.52)	90.34 (0.36)	92.53 (0.47)	93.94 (0.56)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	87.31 (0.42)	84.30 (0.66)	82.24 (0.98)	78.21 (0.92)	73.97 (1.22)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	81.74 (0.54)	79.66 (0.35)	76.46 (0.35)	70.93 (1.10)	63.42 (1.26)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	81.81 (0.31)	84.86 (0.46)	86.14 (0.86)	87.47 (0.42)	88.18 (0.96)
	Worst-group	81.38 (0.41)	79.66 (0.35)	76.46 (0.35)	70.93 (1.10)	63.42 (1.26)
	Average	84.59 (0.40)	84.42 (0.23)	83.80 (0.38)	82.28 (0.32)	79.88 (0.33)
ERM	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	87.42 (0.69)	88.99 (0.55)	91.06 (0.37)	94.30 (0.43)	97.06 (0.29)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	87.20 (0.49)	83.73 (0.60)	80.48 (1.11)	70.24 (1.91)	51.84 (3.09)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	81.87 (0.48)	79.65 (0.34)	75.84 (0.27)	65.92 (1.84)	52.05 (1.37)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	81.92 (0.31)	85.39 (0.55)	87.57 (0.59)	91.60 (0.66)	96.13 (0.70)
	Worst-group	81.50 (0.35)	79.65 (0.34)	75.84 (0.27)	65.26 (1.72)	49.70 (2.22)
	Average	84.60 (0.39)	84.44 (0.24)	83.74 (0.31)	80.52 (0.62)	74.27 (0.73)
INLP	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	87.44 (0.69)	88.96 (0.57)	90.50 (0.41)	75.09 (4.41)	56.45 (1.77)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	87.20 (0.49)	83.84 (0.63)	81.95 (0.77)	65.95 (4.04)	58.26 (4.10)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	81.86 (0.49)	79.73 (0.33)	76.51 (0.34)	62.10 (4.29)	50.64 (3.92)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	81.94 (0.32)	85.30 (0.52)	86.43 (0.67)	63.12 (7.22)	54.54 (3.12)
	Worst-group	81.47 (0.36)	79.73 (0.33)	76.51 (0.34)	57.62 (4.97)	48.13 (3.12)
	Average	84.61 (0.39)	84.46 (0.22)	83.85 (0.36)	66.56 (4.51)	54.97 (2.06)
LEACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.95 (0.56)	83.25 (0.58)	79.56 (0.78)	75.49 (0.80)	69.39 (0.85)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	87.07 (0.41)	89.36 (0.56)	91.13 (0.40)	91.60 (0.38)	91.77 (0.37)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	81.95 (0.69)	85.11 (0.25)	87.81 (0.51)	88.87 (0.70)	90.15 (0.46)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	81.93 (0.37)	78.29 (0.67)	75.39 (1.00)	70.76 (1.65)	68.05 (1.30)
	Worst-group	81.43 (0.50)	78.29 (0.67)	75.39 (1.00)	70.76 (1.65)	67.63 (1.11)
	Average	84.47 (0.40)	84.00 (0.15)	83.47 (0.32)	81.68 (0.48)	79.84 (0.30)
RLACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	87.16 (0.72)	88.90 (0.47)	81.89 (0.82)	76.61 (1.60)	69.33 (1.95)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	87.24 (0.55)	84.02 (0.72)	91.09 (0.39)	92.02 (0.69)	93.70 (0.38)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	81.64 (0.51)	79.81 (0.31)	84.59 (1.12)	86.27 (0.87)	88.90 (1.04)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	82.16 (0.32)	85.01 (0.43)	72.16 (1.18)	66.69 (1.51)	61.15 (1.20)
	Worst-group	81.52 (0.44)	79.81 (0.31)	72.16 (1.18)	66.69 (1.51)	61.15 (1.20)
	Average	84.55 (0.43)	84.43 (0.25)	82.43 (0.35)	80.40 (0.64)	78.27 (0.41)

Table 4. **Results for the Toy dataset for** $\rho \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$. Table shows the average, worst-group, and per-group accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training data. Each accuracy is obtained by averaging over 100 runs. Standard error is reported between brackets.

		ρ				
Method	**Accuracy**	0.0	0.1	0.2	0.3	0.4
JSE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	83.71 (0.17)	83.62 (0.18)	83.46 (0.18)	83.36 (0.18)	83.40 (0.16)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.67 (0.16)	83.56 (0.17)	83.36 (0.19)	83.31 (0.18)	83.24 (0.19)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.73 (0.19)	83.71 (0.20)	83.79 (0.19)	83.71 (0.18)	83.63 (0.20)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.77 (0.15)	83.83 (0.16)	83.81 (0.16)	83.84 (0.17)	83.82 (0.17)
	Worst-group	81.95 (0.12)	81.86 (0.13)	81.81 (0.13)	81.82 (0.13)	81.68 (0.12)
	Average	83.73 (0.09)	83.69 (0.09)	83.61 (0.09)	83.56 (0.09)	83.53 (0.09)
ERM	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	83.55 (0.16)	83.46 (0.18)	83.42 (0.19)	83.52 (0.21)	83.85 (0.23)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.82 (0.17)	83.73 (0.17)	83.38 (0.19)	83.17 (0.22)	82.44 (0.29)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.89 (0.19)	83.83 (0.20)	83.72 (0.20)	83.50 (0.21)	82.82 (0.35)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.61 (0.16)	83.72 (0.16)	83.83 (0.16)	83.92 (0.20)	84.22 (0.21)
	Worst-group	81.90 (0.13)	81.80 (0.13)	81.63 (0.13)	81.41 (0.15)	80.72 (0.25)
	Average	83.74 (0.08)	83.70 (0.08)	83.60 (0.09)	83.54 (0.09)	83.35 (0.10)
INLP	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	83.76 (0.18)	83.07 (0.29)	78.93 (0.80)	72.41 (1.16)	64.74 (1.17)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.64 (0.17)	83.44 (0.24)	80.08 (0.74)	74.74 (1.10)	67.58 (1.24)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.60 (0.22)	83.66 (0.25)	80.52 (0.73)	75.42 (1.08)	68.16 (1.21)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.79 (0.16)	83.43 (0.23)	79.52 (0.82)	72.99 (1.14)	65.16 (1.16)
	Worst-group	81.72 (0.14)	81.26 (0.24)	77.32 (0.81)	70.58 (1.19)	61.41 (1.24)
	Average	83.70 (0.09)	83.41 (0.18)	79.77 (0.75)	73.90 (1.09)	66.45 (1.11)
LEACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	84.05 (0.17)	80.84 (0.20)	77.17 (0.20)	73.39 (0.21)	69.41 (0.21)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.67 (0.18)	85.95 (0.16)	87.71 (0.14)	89.06 (0.14)	89.93 (0.14)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.64 (0.19)	86.23 (0.18)	88.09 (0.15)	89.38 (0.14)	90.38 (0.13)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.69 (0.18)	80.95 (0.20)	77.60 (0.20)	73.79 (0.21)	70.10 (0.21)
	Worst-group	81.82 (0.13)	79.74 (0.18)	76.26 (0.17)	72.38 (0.17)	68.55 (0.17)
	Average	83.77 (0.09)	83.50 (0.08)	82.65 (0.08)	81.42 (0.09)	79.98 (0.10)
RLACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	83.69 (0.21)	81.44 (0.23)	78.52 (0.26)	75.37 (0.26)	72.31 (0.27)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.49 (0.20)	85.42 (0.18)	86.90 (0.17)	88.25 (0.14)	89.08 (0.13)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.57 (0.25)	85.43 (0.23)	87.13 (0.18)	88.47 (0.16)	89.42 (0.14)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.60 (0.21)	81.67 (0.25)	79.27 (0.26)	75.98 (0.28)	72.99 (0.28)
	Worst-group	81.20 (0.16)	80.30 (0.21)	77.72 (0.24)	74.53 (0.25)	71.38 (0.25)
	Average	83.59 (0.09)	83.49 (0.09)	82.96 (0.10)	82.02 (0.11)	80.95 (0.11)

Table 5. **Results for the Toy dataset for** $\rho \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Table shows the average, worst-group, and per-group accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training data. Each accuracy is obtained by averaging over 100 runs. Standard error is reported between brackets.

				ρ		
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9
JSE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	83.27 (0.16)	83.31 (0.17)	83.25 (0.18)	83.30 (0.18)	83.49 (0.26)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.20 (0.18)	83.06 (0.18)	82.99 (0.20)	83.06 (0.21)	81.87 (0.45)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.57 (0.20)	83.54 (0.21)	83.48 (0.22)	83.31 (0.20)	82.35 (0.47)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.67 (0.18)	83.71 (0.18)	83.76 (0.18)	83.65 (0.19)	84.05 (0.25)
	Worst-group	81.54 (0.12)	81.55 (0.13)	81.42 (0.14)	81.44 (0.14)	80.27 (0.42)
	Average	83.43 (0.09)	83.41 (0.10)	83.38 (0.09)	83.33 (0.10)	82.94 (0.18)
ERM	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	83.61 (0.22)	83.76 (0.27)	84.60 (0.31)	86.00 (0.31)	88.72 (0.28)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	82.76 (0.28)	82.10 (0.40)	80.39 (0.53)	77.80 (0.77)	69.97 (0.88)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.03 (0.30)	82.50 (0.42)	80.62 (0.60)	77.78 (0.79)	70.11 (0.94)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.93 (0.22)	84.16 (0.24)	84.97 (0.28)	85.99 (0.32)	88.86 (0.26)
	Worst-group	80.76 (0.22)	80.06 (0.36)	78.51 (0.53)	76.01 (0.77)	68.41 (0.91)
	Average	83.35 (0.09)	83.15 (0.12)	82.67 (0.17)	81.90 (0.26)	79.43 (0.35)
INLP	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	59.94 (1.03)	58.54 (0.90)	55.66 (0.81)	54.71 (0.68)	53.06 (0.69)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	62.83 (1.19)	61.77 (1.26)	58.31 (1.29)	56.64 (1.27)	56.01 (1.53)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	62.82 (1.19)	60.43 (1.31)	58.98 (1.26)	57.09 (1.26)	58.23 (1.48)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	60.52 (1.01)	57.87 (0.95)	56.60 (0.82)	55.09 (0.65)	55.14 (0.70)
	Worst-group	55.51 (1.06)	53.06 (0.95)	50.06 (0.77)	48.59 (0.55)	47.06 (0.56)
	Average	61.58 (0.94)	59.71 (0.89)	57.44 (0.76)	55.93 (0.68)	55.67 (0.74)
LEACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	65.63 (0.20)	62.01 (0.22)	58.69 (0.22)	55.42 (0.23)	52.64 (0.25)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	90.76 (0.12)	91.21 (0.12)	91.49 (0.12)	91.70 (0.12)	91.63 (0.14)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	91.02 (0.13)	91.61 (0.12)	91.82 (0.13)	92.11 (0.12)	92.05 (0.14)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	65.99 (0.23)	62.41 (0.21)	58.94 (0.25)	55.85 (0.26)	52.96 (0.29)
	Worst-group	64.60 (0.17)	60.95 (0.18)	57.37 (0.18)	54.02 (0.19)	50.97 (0.20)
	Average	78.37 (0.10)	76.83 (0.09)	75.27 (0.10)	73.80 (0.09)	72.35 (0.10)
RLACE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	68.59 (0.27)	64.77 (0.27)	60.84 (0.29)	57.30 (0.28)	53.89 (0.30)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	89.97 (0.14)	90.60 (0.13)	91.02 (0.12)	91.21 (0.13)	90.71 (0.30)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	90.22 (0.13)	90.87 (0.13)	91.30 (0.13)	91.65 (0.14)	90.89 (0.39)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	69.32 (0.33)	65.42 (0.31)	61.70 (0.31)	57.85 (0.32)	54.39 (0.35)
	Worst-group	67.65 (0.27)	63.87 (0.27)	59.74 (0.27)	55.89 (0.25)	52.31 (0.27)
	Average	79.53 (0.13)	77.92 (0.13)	76.22 (0.13)	74.50 (0.13)	72.48 (0.20)

B. Additional Results for Toy dataset

B.1. Removing the Orthogonality Assumption

In this section, we provide an analysis of how JSE performs when the $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ are not orthogonal subspaces. We create an example of the Toy dataset where the orthogonality assumption does not hold, by changing the angle of $\boldsymbol{w}_{\text{sp}}$ and $\boldsymbol{w}_{\text{mt}}$ to 75°. Let $a = \cos(\frac{15\pi}{180}), b = \sin(\frac{15\pi}{180})$, and $\boldsymbol{w}_{\text{sp}} = \left(\gamma, 0, 0, \dots, 0\right)^{\top}$ and $\boldsymbol{w}_{\text{mt}} = \left(\frac{\gamma}{1+\frac{a}{b}}, \frac{\gamma}{1+\frac{b}{a}}, 0 \dots, 0\right)^{\top}$. The main-task labels are now determined by a linear combination of the spurious and main-task directions. In Figure 8, we illustrate the subspaces found by JSE for this scenario. JSE finds a spurious and main-task vector that are slightly different from the basis vectors, yet orthogonal. This illustrates that when the $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$ are not orthogonal subspaces, JSE finds two orthogonal subspaces that best fit given the data.

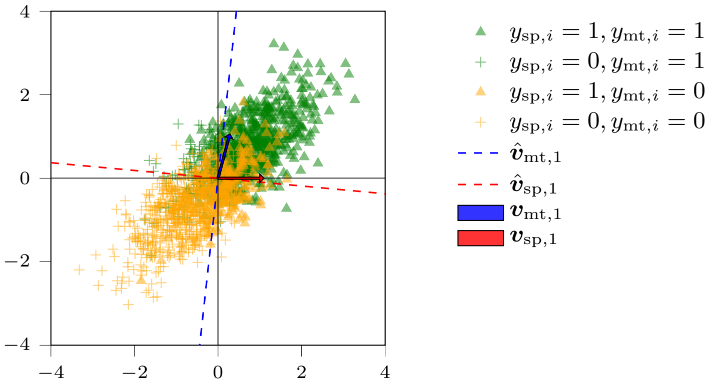

Figure 8. **Illustration of vectors found by JSE with non-orthogonal spurious and main-task subspaces**: The blue and red arrow indicate the basis of the spurious and main-task subspaces, which have an angle of 75°. Data is based on a single simulation from the $d(=20)$-dimensional Toy dataset of Section 4.1 with $\rho = 0.8$ and sample size $n = 2{,}000$.

Figure 9 shows the performance for the Toy dataset when the subspaces are not orthogonal. Compared to the case where the subspaces are orthogonal, the performance of JSE is slightly worse. It removes a small part of the main-task direction, and leaves a small part of the spurious direction. However, it still outperforms other concept-removal methods. These methods perform relatively worse, because they also remove part of the main-task direction - even when there is no correlation between the spurious and main-task direction. When the spurious direction is removed, the part of the main-task direction that is non-orthogonal to it is also removed.

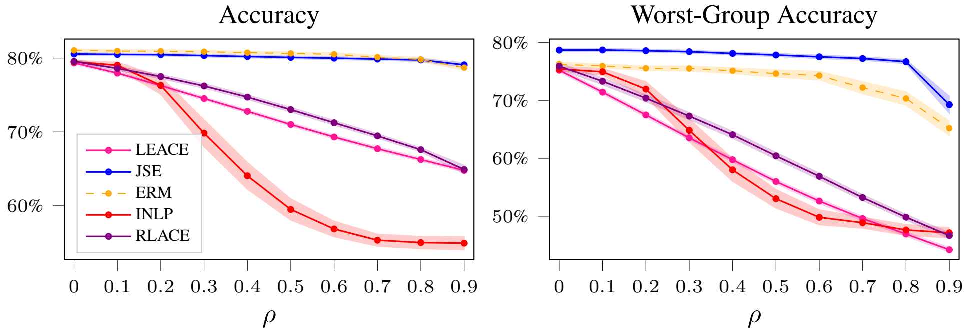

Figure 9. **OOD generalization for Toy dataset, when the angle between the spurious and main-task vector is** 75°: We plot the (worst-group) accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training data. Each accuracy is obtained by averaging over 100 runs. The shaded area reflects the 95% confidence interval.

B.2. Finite-sample Estimation Noise for JSE and INLP

In this section, we briefly illustrate how the performance of JSE and INLP is affected by the interaction of (i) the size of the training set, and (ii) the correlation between the spurious and main-task features. Figure 10 shows the result of applying JSE and INLP to the Toy dataset for different sizes of the dataset. For JSE, we observe that with a limited sample size and a high spurious correlation, it is harder to separate the spurious and main-task features. We attribute this to finite-sample noise: our method finds two orthogonal vectors that fit well in the training data, but they are less likely to align with the data-generating process. Interestingly, INLP becomes worse as the sample size increases. With a greater sample size, it is more likely that INLP assigns the main-task feature as belonging to the spurious subspace, since its predictive ability of the spurious concept label is more likely to be detected.

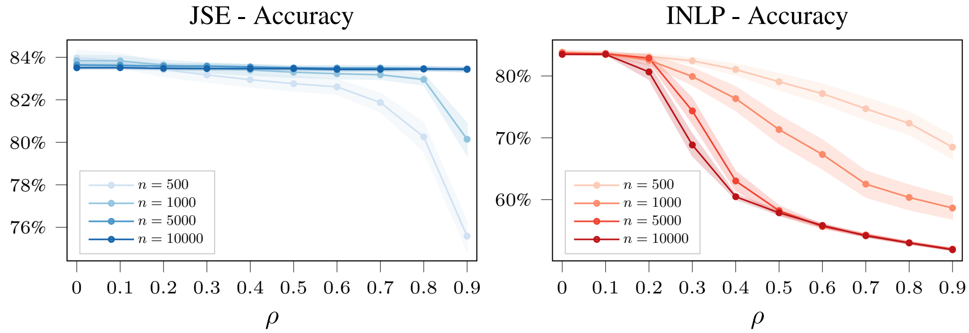

Figure 10. **Effect of training set size and spurious correlation strength for Toy dataset**: We plot the accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training data. Each accuracy is obtained by averaging over 100 runs. The shaded area reflects the 95% confidence interval.

C. The JSE Algorithm

In this section, we first provide a more detailed version of the JSE algorithm, initially described in Section 3.2. We briefly investigate two modifications of the algorithm in Section C.2. Finally, we perform a brief ablation on the Waterbirds for how JSE performs without first applying PCA to the embeddings in Section C.3.

C.1. Detailed Description of the JSE Algorithm

In Equation 2 we note that we optimize over the coefficients for two logistic regressions, while enforcing an orthogonality constraint. Here, we briefly explain how this constraint is enforced. We solve the following unconstrained problem using stochastic gradient descent (SGD):

$$\hat{\boldsymbol{w}}_{\text{sp}}, \hat{\boldsymbol{w}}_{\text{mt}}, \hat{b}_{\text{sp}}, \hat{b}_{\text{mt}} = \underset{\boldsymbol{w}_{\text{sp}}, \boldsymbol{w}_{\text{mt}}, b_{\text{sp}}, b_{\text{mt}}}{\arg\min} \sum_{i=1}^{n} \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{sp},i}, y_{\text{sp},i}) + \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{mt},i}, y_{\text{mt},i}),$$

where we define the predictions as follows:

$$\begin{aligned} \hat{y}_{\text{sp},i} &= \text{Logit}^{-1}\left(\boldsymbol{z}_i^{\top}\boldsymbol{w}_{\text{sp}} + b_{\text{sp}}\right), \\ \hat{y}_{\text{mt},i} &= \text{Logit}^{-1}\left(\boldsymbol{z}_i^{\top}(\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{w}_{\text{sp}}})\boldsymbol{w}_{\text{mt}} + b_{\text{mt}}\right), \\ \boldsymbol{P}_{\boldsymbol{w}_{\text{sp}}} &= \boldsymbol{w}_{\text{sp}}\left(\boldsymbol{w}_{\text{sp}}^{\top}\boldsymbol{w}_{\text{sp}}\right)^{-1}\boldsymbol{w}_{\text{sp}}^{\top}. \end{aligned}$$

By definition, $(\boldsymbol{I} - \boldsymbol{P}_{\boldsymbol{w}_{\text{sp}}})\boldsymbol{w}_{\text{mt}}$ is orthogonal to $\boldsymbol{w}_{\text{sp}}$. Therefore, the predictions for each $y_{\text{mt},i}$ are based upon a set of coefficients that is orthogonal to $\boldsymbol{w}_{\text{sp}}$. By weighing both losses equally, there is no reason for this optimization problem to favor the loss for one set of labels over the other.

In Algorithm 2 we provide the exact same procedure as in Algorithm 1, but in greater detail.

Algorithm 2 JSE algorithm to estimate orthonormal bases for $\mathcal{Z}_{\text{sp}}$ and $\mathcal{Z}_{\text{mt}}$. The calculation of the test statistics is discussed in Section D.3

```
Require: a sample $\{y_{\text{mt},k}, y_{\text{sp},k}, \boldsymbol{z}_k\}_{k=1}^n$ consisting of two binary labels and a vector $\boldsymbol{z}_k \in \mathbb{R}^d$.
  Initialize a $(n \times d)$-dimensional embedding matrix $\boldsymbol{Z} = \left(\boldsymbol{z}_1\, \boldsymbol{z}_2\, \cdots\, \boldsymbol{z}_n\right)^\top$.
  Initialize $\boldsymbol{Z}_{\text{sp}}^{\perp} \leftarrow \boldsymbol{Z}$.
  Choose a significance level $\alpha$ for the test statistics, resulting in critical value $t_{1-\alpha}$
  for $i = 1, ..., d$ do
    $\boldsymbol{Z}_{\text{remain}} \leftarrow \boldsymbol{Z}_{\text{sp}}^{\perp}$
    for $j = 1, ..., d$ do

      Estimate $\hat{\boldsymbol{w}}_{\text{sp}}$ and $\hat{\boldsymbol{w}}_{\text{mt}}$ with Equation 2, using embeddings $\boldsymbol{Z}_{\text{remain}}$.

      Define normalizations $\hat{\boldsymbol{v}}_{\text{sp},i} \leftarrow \hat{\boldsymbol{w}}_{\text{sp}}/||\hat{\boldsymbol{w}}_{\text{sp}}||$ and $\hat{\boldsymbol{v}}_{\text{mt},j} \leftarrow \hat{\boldsymbol{w}}_{\text{mt}}/||\hat{\boldsymbol{w}}_{\text{mt}}||$.

      Estimate $\hat{\gamma}_{\text{sp}}, \hat{b}_{\text{sp}} = \underset{\gamma_{\text{sp}}, b_{\text{sp}}}{\arg\min} \sum_{h=1}^{n} \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{sp},h}^{(\hat{\boldsymbol{v}}_{\text{mt},j})}, y_{\text{sp},h})$, where $\hat{y}_{\text{sp},h}^{(\hat{\boldsymbol{v}}_{\text{mt},j})} = \text{Logit}^{-1}\left(\gamma_{\text{sp}}\, \boldsymbol{z}_{h,\text{remain}}^\top \hat{\boldsymbol{v}}_{\text{mt},j} + b_{\text{sp}}\right)$
      Calculate $t_{\text{mt,rnd}}, t_{\text{mt,sp}}^{(\hat{\boldsymbol{v}}_{\text{mt},j})}$ using $\hat{\gamma}_{\text{sp}}, \hat{b}_{\text{sp}}$ (see Section D.3)

      if $(t_{\text{mt,rnd}} < -t_{1-\alpha}$ ) and $(t_{\text{mt,sp}}^{(\hat{\boldsymbol{v}}_{\text{mt},j})} > t_{1-\alpha})$ then
        Projection $\boldsymbol{Z}_{\text{remain}} \leftarrow \boldsymbol{Z}_{\text{sp}}^{\perp}(\boldsymbol{I} - \hat{\boldsymbol{V}}_{\text{mt}} \hat{\boldsymbol{V}}_{\text{mt}}^\top)$, where $\hat{\boldsymbol{V}}_{\text{mt}} = \left(\hat{\boldsymbol{v}}_{\text{mt},1}\, \hat{\boldsymbol{v}}_{\text{mt},2}\, \cdots\, \hat{\boldsymbol{v}}_{\text{mt},j}\right)$
      else
        break
      end if
    end for

    Estimate $\hat{\gamma}_{\text{mt}}, \hat{b}_{\text{mt}} = \sum_{h=1}^{n} \underset{\gamma_{\text{mt}}, b_{\text{mt}}}{\arg\min} \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{mt},h}^{(\hat{\boldsymbol{v}}_{\text{sp},i})}, y_{\text{mt},h})$, where $\hat{y}_{\text{mt},h}^{(\hat{\boldsymbol{v}}_{\text{sp},i})} = \text{Logit}^{-1}\left(\gamma_{\text{mt}} \boldsymbol{z}_{i,\text{sp}}^{\perp\top} \hat{\boldsymbol{v}}_{\text{sp},i} + b_{\text{sp}}\right)$
    Calculate $t_{\text{sp,rnd}}, t_{\text{mt,sp}}^{(\hat{\boldsymbol{v}}_{\text{sp},i})}$ using $\hat{\gamma}_{\text{mt}}, \hat{b}_{\text{mt}}$ (see Section D.3)

    if $(t_{\text{sp,rnd}} < -t_{1-\alpha}$ ) and $(t_{\text{mt,sp}}^{(\hat{\boldsymbol{v}}_{\text{sp},i})} < -t_{1-\alpha})$ then
      Projection $\boldsymbol{Z}_{\text{sp}}^{\perp} \leftarrow \boldsymbol{Z}(\boldsymbol{I} - \hat{\boldsymbol{V}}_{\text{sp}} \hat{\boldsymbol{V}}_{\text{sp}}^\top)$, where $\hat{\boldsymbol{V}}_{\text{sp}} = \left(\hat{\boldsymbol{v}}_{\text{sp},1}\, \hat{\boldsymbol{v}}_{\text{sp},2}\, \cdots\, \hat{\boldsymbol{v}}_{\text{sp},i}\right)$.
      $\hat{\boldsymbol{v}}'_{\text{mt},\ell} \leftarrow \hat{\boldsymbol{v}}_{\text{mt},\ell}$, for $\ell = 1, 2, \ldots, \ell_0$, with $\ell_0 = j$.
    else
      break
    end if
  end for
  return Bases $\{\hat{\boldsymbol{v}}_{\text{sp},m}\}_{m=1}^{i-1}$ and $\{\hat{\boldsymbol{v}}'_{\text{mt},\ell}\}_{\ell=1}^{\ell_0-1}$.
```

C.2. Modifications of the JSE Algorithm

In this section, we briefly compare the formulation of the JSE algorithm to two alternatives.

- **Swapping the loops:** one might consider interchanging the inner loop and the outer loop of the JSE algorithm. Now, at each step, first the spurious vectors $\hat{\boldsymbol{v}}_{\text{sp},1}, \hat{\boldsymbol{v}}_{\text{sp},2}, \ldots, \hat{\boldsymbol{v}}_{\text{sp},d_{\text{sp}}}$ are projected out before the main-task vector (at that step) is estimated.
- **Projecting onto main-task subspace**: instead of projecting $\boldsymbol{z}$ onto the orthogonal complement of $\mathcal{Z}_{\text{sp}}$, one could be interested in projecting onto $\mathcal{Z}_{\text{mt}}$. In this case, rather than the transformation $(\boldsymbol{I} - \boldsymbol{V}_{\text{sp}} \boldsymbol{V}_{\text{sp}}^\top)\boldsymbol{z}$, one uses the transformed embeddings $(\boldsymbol{V}_{\text{mt}} \boldsymbol{V}_{\text{mt}}^\top)\boldsymbol{z}$.

In Table 6 we compare these two alternative formulations of the algorithm for the Waterbirds dataset. The performance for these two versions is similar to the JSE algorithm as outlined in Section 3.2. We suggest that in practice, which version of the algorithm should be used could depend on (1) whether the user wants to remove a certain spurious concept, or isolate certain main-task features, and (2) the dimensionality of both subspaces. For example, if the dimension of the main-task subspace is much lower than that of the spurious concept subspace, it might be worthwhile to project onto $\boldsymbol{z}_{\mathrm{mt}}$ to improve the bias-variance trade-off, rather than removing $\boldsymbol{z}_{\mathrm{sp}}$.

Table 6. **Results for the Waterbirds dataset for different versions of the JSE algorithm**: Table shows the average, worst-group, and per-group accuracy on a test set where $p_{\mathrm{OOD}}(y_{\mathrm{mt}} = y | y_{\mathrm{sp}} = y) = 0.5$, with $y \in \{0, 1\}$. Each accuracy is obtained by averaging over 5 runs. Standard error is reported between brackets.

				$p_{\mathrm{train}}(y_{\mathrm{mt}} = y \| y_{\mathrm{sp}} = y)$		
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9
JSE with projecting onto $\boldsymbol{z}_{\mathrm{mt}}$	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 0$	90.98 (0.45)	91.90 (0.46)	89.06 (0.72)	89.39 (0.24)	92.39 (0.65)
	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 1$	88.52 (0.59)	89.38 (0.39)	89.37 (0.32)	91.54 (0.23)	89.22 (0.91)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 0$	91.00 (0.18)	90.81 (0.49)	92.15 (0.41)	91.56 (0.25)	88.32 (0.80)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 1$	89.72 (0.40)	88.94 (0.39)	90.12 (0.38)	88.01 (0.48)	88.82 (0.75)
	Worst-group	88.43 (0.55)	88.47 (0.17)	88.61 (0.45)	88.01 (0.48)	86.86 (0.26)
	Average	89.89 (0.35)	90.47 (0.24)	89.64 (0.33)	90.31 (0.09)	90.31 (0.42)
JSE with Swapped Loops	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 0$	91.59 (0.72)	90.71 (1.91)	91.16 (0.32)	91.49 (0.64)	90.83 (0.67)
	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 1$	88.39 (0.79)	87.65 (1.80)	89.05 (0.72)	90.40 (0.35)	89.18 (0.90)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 0$	89.91 (0.33)	90.53 (0.54)	90.81 (0.29)	90.28 (0.20)	89.28 (0.61)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 1$	89.25 (0.46)	90.16 (0.65)	89.28 (0.26)	88.82 (0.47)	87.91 (0.55)
	Worst-group	87.83 (0.59)	87.16 (1.58)	88.34 (0.46)	88.75 (0.44)	87.11 (0.30)
	Average	89.90 (0.50)	89.44 (1.31)	90.09 (0.35)	90.64 (0.19)	89.69 (0.41)

C.3. Using JSE With and Without First Applying PCA to the Embeddings

For our results in Section 4.2, we first apply PCA to the embeddings to reduce their respective dimension. This section serves to show that it is possible for JSE to be successful without first applying PCA to the embeddings. We compare the performance of JSE on the Waterbirds dataset with and without PCA in Figure 11, and observe that there is little to no difference in performance. However, the version using PCA is computationally more efficient due to the reduced dimensionality (2048 for the original last-layer embeddings of Resnet, vs. 300 after PCA). Aplying PCA to the embeddings can potentially be useful in cases where there is a limited set of datapoints available, and the dimensionality of the embeddings is very large.

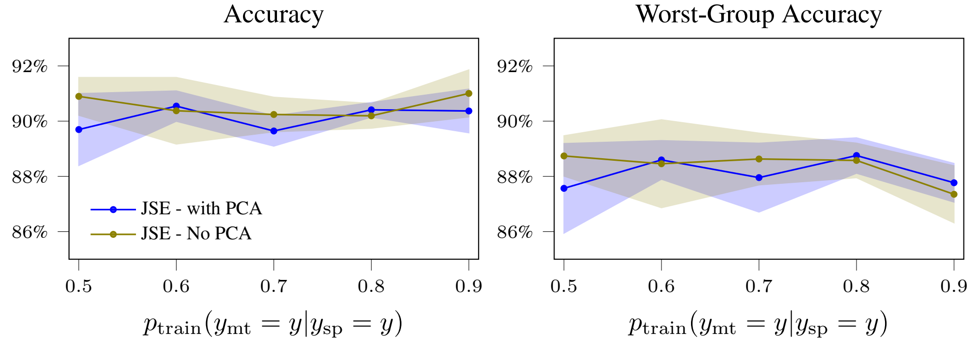

Figure 11. **Effect of using JSE with and without PCA for the Waterbirds dataset**: We plot the (worst-group) accuracy on an OOD test set where $p_{\mathrm{OOD}}(y_{\mathrm{mt}} = y | y_{\mathrm{sp}} = y) = 0.5$, as a function of $p_{\mathrm{train}}(y_{\mathrm{mt}} = y | y_{\mathrm{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. The shaded area reflects the 95% confidence interval. When using PCA, the dimension of the embeddings is reduced to $d = 300$.

D. Details on the Testing Procedure of JSE

In this section, we provide a detailed explanation of the testing procedure used in the JSE algorithm for breaking the for-loops. Section D.1 will introduce notation, and a general set-up that is applicable for each test. In Section D.2 we provide the derivations that are needed for our subsequent test statistics. In Section D.3 we define the test statistics that are used throughout the JSE method. In Section D.4 we study the effect of our choice to equally weight across the four groups in the data when measuring the test statistics, and in Section D.5 we investigate how one can adjust the test in cases where one set of labels is much harder to predict than the other.

D.1. Notation and Set-up

Recall from Section 3.3 that we are interested in testing the difference between two binary cross-entropies (BCE). For example, we can define the difference in BCE between a logistic regression that is trained on our embeddings $\boldsymbol{z}$ to predict the spurious concept labels y_{sp} and one that just uses an intercept (referred to as 'random' classifier).

For a particular sample, one could simply estimate this difference by weighting the individual observations equally. However, for our tests, we noticed that there was a considerable improvement if the difference was weighted equally within the subgroups specified by the pair of labels $(y_{\text{mt}}, y_{\text{sp}})$. This is examined in Section D.4. Below, we formulate how one, in general, can test for such a weighted difference between BCE's of two classifiers.

Based on the main task and spurious labels $y_{\text{mt}}, y_{\text{sp}}$, we define four groups: (1) $y_{\text{mt}} = 0, y_{\text{sp}} = 0$, (2) $y_{\text{mt}} = 0, y_{\text{sp}} = 1$, (3) $y_{\text{mt}} = 1, y_{\text{sp}} = 0$, and (4) $y_{\text{mt}} = 1, y_{\text{sp}} = 1$. The spurious and main-task labels are used to define a new random variable $G \in \{1, 2, 3, 4\}$, indicating group membership. The probability of a group g is noted as π_g.

We consider the difference between the binary cross-entropy (BCE) as a random variable d. We assume that this difference is a mixture of four random variables: d_1, d_2, d_3, d_4. The random variable d_g corresponds to the difference in BCE for group g. We assume that the expectation and variance of each of these four random variables is different, e.g. $\mathbb{E}[d_g] \neq \mathbb{E}[d_h]$, $\text{Var}[d_g] \neq \text{Var}[d_h]$, for $g \neq h$. However, we do assume that they are independent: $\text{Cov}(d_g, d_h) = 0$. We use the use the notation $\mathbb{E}[d_g] = \mu_g$ and $\text{Var}[d_g] = \sigma_g^2$.

We are interested in the weighted average of d_1, d_2, d_3, d_4, where each group receives an equal weight. Concretely, we can define a new random variable

$$d_w = \frac{1}{4}\sum_{g=1}^{4} d_g.$$

The subscript of w will be used to refer to the equally weighted sum of difference d. We are interested in the following hypotheses

$$H_0 : \mathbb{E}[d_w] = 0, \quad H_1 : \mathbb{E}[d_w] < 0.$$

We draw a sample of n independent and identical (IID) observations from d and G.

Because we observe G, we know for each observation which random variable we are observing - e.g. if $G = 1$, we observe d_1. This means that after drawing the n observations, we observe the n_g observations for group g. The observations of d from group g are denoted as $d_{g,i}$ for $i = 1, 2, ..., n_g$. However, we do not know the value of each π_g. We do assume that each π_g is strictly positive.

D.2. Derivation of Test Statistic

The expectation of d_w is

$$\mu_w = \mathbb{E}[d_w] = \frac{1}{4}\mathbb{E}\left[\sum_{g=1}^{4} d_g\right] = \frac{1}{4}\sum_{g=1}^{4}\mu_g.$$

Let $\bar{d}_w$ denote an estimator of d_w

$$\bar{d}_w = \frac{1}{4}\sum_{g=1}^{4}\bar{d}_g, \quad \text{with} \quad \bar{d}_g = \frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g},$$

where n_g is a random variable. We can show this estimator is unbiased via the law of total expectation.

$$\begin{aligned}
\mathbb{E}[\bar{d}_g] &= \mathbb{E}\left[\frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g}|n_g\right]\right] \\
&= \mathbb{E}\left[\frac{1}{n_g}\sum_{i=1}^{n_g} \mathbb{E}\left[d_{i,g}|n_g\right]\right] \\
&= \mathbb{E}\left[\frac{1}{n_g} n_g \mu_g\right] && \text{(Since they are IID)} \\
&= \mu_g, \\
\mathbb{E}[\bar{d}_w] &= \frac{1}{4}\sum_{g=1}^{4} \mu_g.
\end{aligned}$$

We now turn to the variance of $\bar{d}_w$

$$\begin{aligned}
\mathrm{Var}(\bar{d}_w) &= \mathrm{Var}\left(\frac{1}{4}\sum_{g=1}^{4} \bar{d}_g\right) \\
&= \frac{1}{16}\mathrm{Var}\left(\sum_{g=1}^{4} \bar{d}_g\right) \\
&= \frac{1}{16}\sum_{g=1}^{4}\sum_{h=1}^{4} \mathrm{Cov}(\bar{d}_g, \bar{d}_h).
\end{aligned}$$

First, it is shown first show that $\mathrm{Cov}(\bar{d}_g, \bar{d}_h) = 0$ for $g \neq h$ via the law of total covariance:

$$\mathrm{Cov}(\bar{d}_g, \bar{d}_h) = \mathbb{E}[\mathrm{Cov}(\bar{d}_g, \bar{d}_h|n_g, n_h)] + \mathrm{Cov}(\mathbb{E}[\bar{d}_g|n_g, n_h], \mathbb{E}[\bar{d}_h|n_g, n_h])$$

Since the sample means in expectation are the constants μ_g, μ_h, their covariance is 0:

$$\mathrm{Cov}(\mathbb{E}[\bar{d}_g|n_g, n_h], \mathbb{E}[\bar{d}_h|n_g, n_h]) = \mathrm{Cov}(\mu_g, \mu_h) = 0.$$

Next, we define

$$\begin{aligned}
\mathbb{E}[\mathrm{Cov}(\bar{d}_g, \bar{d}_h|n_g, n_h)] &= \mathbb{E}\left[\frac{1}{n_g}\frac{1}{n_h}\mathrm{Cov}\left(\sum_{i=1}^{n_g} d_{i,g}, \sum_{i=1}^{n_h} d_{i,h}|n_g, n_h\right)\right] \\
&= \mathbb{E}\left[\frac{1}{n_g}\frac{1}{n_h} \times 0\right] = 0,
\end{aligned}$$

This last step can be made because we assume $d_{g,i}$ is independent of $d_{h,j}$ for $g \neq h$, $i = 1, ..., n_g$ and $j = 1, ..., n_h$. The variance of $\bar{d}_w$ becomes

$$\mathrm{Var}\left(\bar{d}_w\right) = \frac{1}{16}\sum_{g=1}^{4} \mathrm{Var}\left(\bar{d}_g\right).$$

We can define $\mathrm{Var}\left(\bar{d}_g\right)$ via the law of total variance

$$\begin{aligned}
\mathrm{Var}\left(\bar{d}_g\right) &= \mathrm{Var}\left(\frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g}\right)\\
&= \mathbb{E}\left[\mathrm{Var}\left(\frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g}|n_g\right)\right] + \mathrm{Var}\left(\mathbb{E}\left[\frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g}|n_g\right]\right)\\
&= \mathbb{E}\left[\mathrm{Var}\left(\frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g}|n_g\right)\right] + \mathrm{Var}\left(\mu_g\right)\\
&= \mathbb{E}\left[\mathrm{Var}\left(\frac{1}{n_g}\sum_{i=1}^{n_g} d_{i,g}|n_g\right)\right] && \text{(Since variance of constant is 0)}\\
&= \mathbb{E}\left[\frac{1}{n_g^2}\sum_{i=1}^{n_g} \mathrm{Var}\left(d_{i,g}\right)\right]\\
&= \mathbb{E}[\frac{1}{n_g^2} n_g \sigma_g^2] && \text{(Since they are IID)}\\
&= \mathbb{E}\left[\frac{1}{n_g}\right]\sigma_g^2.
\end{aligned}$$

For $n \to \infty$, n_g approximately follows a binomial distribution with $\mathbb{E}[n_g] = n\pi_g$ and variance $n\pi_g(1-\pi_g)$. We define a second order Taylor expansion to approximate $\frac{1}{n_g}$ around $\mathbb{E}[n_g] = n\pi_g$

$$\mathbb{E}\left[\frac{1}{n_g}\right] \approx \frac{1}{n\pi_g} + \frac{(1-\pi_g)}{n^2\pi_g^2}.$$

This means:

$$\begin{aligned}
\mathrm{Var}\left(\bar{d}_g\right) &\approx \left(\frac{1}{n\pi_g} + \frac{(1-\pi_g)}{n^2\pi_g^2}\right)\sigma_g^2\\
&= \frac{1}{n\pi_g}\sigma_g^2 + \mathcal{O}(n^{-2}).
\end{aligned}$$

Using this, we approximate the variance of the weighted sum d_w via

$$\mathrm{Var}(\bar{d}_w) \approx \frac{1}{16}\sum_{g=1}^{4}\frac{1}{n\pi_g}\sigma_g^2.$$

Using the expectation and variance of $\bar{d}_w$, we now proceed to its asymptotic distribution. Assuming that (i) the sample means $\bar{d}_g$ are based on IID random variables and (ii) σ_g^2 is bounded, we can use the central limit theorem (CLT) for each of the four sample means

$$\sqrt{n}\left(\bar{d}_g - \mu_g\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{\sigma_g^2}{\pi_g}\right).$$

This holds because $lim_{n\to\infty} n_g/n = \pi_g > 0$ for all g; see for instance Rényi (1957). Given that the CLT holds for each sample mean, joint convergence follows by independence of the sample means. Hence, the distribution of the linear combination directly follows

$$\sqrt{n}(\bar{d}_w - \mathbb{E}[d_w]) = \frac{1}{4}\sum_{g=1}^{4}\sqrt{n}(\bar{d}_g - \mu_g) \xrightarrow{d} \mathcal{N}\left(0, \frac{1}{16}\sum_{g=1}^{4}\frac{\sigma_g^2}{\pi_g}\right).$$

Hence, the variance of $\bar{d}_w$

$$\mathrm{Var}(\bar{d}_w) \approx \frac{1}{16} \sum_{g=1}^{4} \frac{\sigma_g^2}{n\pi_g}$$

can be consistently estimated via

$$\widehat{\mathrm{Var}}(\bar{d}_w) = \frac{1}{16} \sum_{g=1}^{4} \frac{s_g^2}{n\hat{\pi}_g} = \frac{1}{16} \sum_{g=1}^{4} \frac{s_g^2}{n_g},$$

with $\hat{\pi}_g = n_g/n$, and $s_g^2 = \frac{1}{n_g-1} \sum_{i=1}^{n_g} (d_{g,i} - \bar{d}_g)^2$. Using this, we can define a test statistic t_w, which for $n \to \infty$

$$t_w = \frac{\bar{d}_w - \mathbb{E}[d_w]}{\widehat{\mathrm{Var}}(\bar{d}_w)} \xrightarrow{d} \mathcal{N}(0, 1).$$

D.3. Test statistics for JSE

In the previous section, we defined a test statistic for the equally weighted weighted average of d. Here, we will use this derivation for the test statistics in the inner and outer loop of JSE.

We start with the first criterion, namely that the vector $\boldsymbol{v}_{\mathrm{sp}}$ ($\boldsymbol{v}_{\mathrm{mt}}$) is informative about the spurious label (main-task label). This criterion is operationalised as follows: the $\boldsymbol{v}_{\mathrm{sp}}$ should contain more information about y_{sp} than a majority-rule ‘random classifier’. The coefficients for a logistic regression can be written as a combination of a unit vector and a scalar: $\boldsymbol{w} = \boldsymbol{v}\gamma$. Consider a logistic regression model $\hat{y}_{\mathrm{sp}}^{(\boldsymbol{v}_{\mathrm{sp}})} = \mathrm{Logit}^{-1}\left(\gamma_{\mathrm{sp}} \boldsymbol{z}^\top \boldsymbol{v}_{\mathrm{sp}} + b_{\mathrm{sp}}\right)$. This is a predictor for the label y_{sp} based on the embeddings projected onto $\boldsymbol{v}_{\mathrm{sp}}$. Let $\hat{y}_{\mathrm{sp}}^{(\mathrm{rnd})}$ denote a random classifier. We can define the difference between these two classifiers as

$$d_{\mathrm{sp,rnd}}^{(\boldsymbol{v}_{\mathrm{sp}})} = \mathcal{L}_{\mathrm{BCE}}(\hat{y}_{\mathrm{sp}}^{(\boldsymbol{v}_{\mathrm{sp}})}, y_{\mathrm{sp}}) - \mathcal{L}_{\mathrm{BCE}}(\hat{y}_{\mathrm{sp}}^{(\mathrm{rnd})}, y_{\mathrm{sp}}).$$

The first criterion translates into the following hypothesis

$$H_0 : \mathbb{E}[d_{w,\mathrm{sp,rnd}}^{(\boldsymbol{v}_{\mathrm{sp}})}] = 0 \quad \text{versus} \quad H_1 : \mathbb{E}[d_{w,\mathrm{sp,rnd}}^{(\boldsymbol{v}_{\mathrm{sp}})}] < 0,$$

which means that under the null hypothesis, there is no difference in the BCE of these two classifiers. Under the alternative hypothesis, the BCE of a random classifier is higher.

We use the following test statistic, where under the null hypothesis

$$t_{\mathrm{sp,rnd}} = \frac{\bar{d}_{w,\mathrm{sp,rnd}}^{(\boldsymbol{v}_{\mathrm{sp}})}}{\mathrm{Var}\left(\bar{d}_{w,\mathrm{sp,rnd}}^{(\boldsymbol{v}_{\mathrm{sp}})}\right)} \xrightarrow{d} \mathcal{N}(0, 1).$$

The previous test can also be defined for a main-task vector $\boldsymbol{v}_{\mathrm{mt}}$ and main-task labels y_{mt}. Consider a logistic regression model $\hat{y}_{\mathrm{mt}}^{(\boldsymbol{v}_{\mathrm{mt}})} = \mathrm{Logit}^{-1}\left(\gamma_{\mathrm{mt}} \boldsymbol{z}^\top \boldsymbol{v}_{\mathrm{mt}} + b_{\mathrm{mt}}\right)$. The difference between the BCE of this classifier and a random classifier $\hat{y}_{\mathrm{mt}}^{(\mathrm{rnd})}$ is

$$d_{\mathrm{mt,rnd}}^{(\boldsymbol{v}_{\mathrm{mt}})} = \mathcal{L}_{\mathrm{BCE}}(\hat{y}_{\mathrm{mt}}^{(\boldsymbol{v}_{\mathrm{mt}})}, y_{\mathrm{mt}}) - \mathcal{L}_{\mathrm{BCE}}(\hat{y}_{\mathrm{mt}}^{(\mathrm{rnd})}, y_{\mathrm{mt}}),$$

which we can use to test the hypothesis

$$H_0 : \mathbb{E}[d_{w,\mathrm{mt,rnd}}^{(\boldsymbol{v}_{\mathrm{mt}})}] = 0 \quad \text{versus} \quad H_1 : \mathbb{E}[d_{w,\mathrm{mt,rnd}}^{(\boldsymbol{v}_{\mathrm{mt}})}] < 0.$$

For these hypotheses, we define a test statistic similar to the previous one, only now for the main-task vector and labels

$$t_{\mathrm{mt,rnd}} = \frac{\bar{d}_{w,\mathrm{mt,rnd}}^{(\boldsymbol{v}_{\mathrm{mt}})}}{\mathrm{Var}\left(\bar{d}_{w,\mathrm{mt,rnd}}^{(\boldsymbol{v}_{\mathrm{mt}})}\right)} \xrightarrow{d} \mathcal{N}(0, 1).$$

We now turn to the second criterion, which is that the $\boldsymbol{v}_{\text{sp}}$ is more predictive of the spurious concept than the main-task concept (and vice-versa for $\boldsymbol{v}_{\text{mt}}$). This criterion is operationalised as follows: The BCE of a spurious vector $\boldsymbol{v}_{\text{sp}}$ should be lower for the spurious concept than the main-task, and the vice-versa. We compare the BCE's of $\hat{y}_{\text{sp}}^{(\boldsymbol{v}_{\text{sp}})}$ and $\hat{y}_{\text{mt}}^{(\boldsymbol{v}_{\text{sp}})} = \text{Logit}^{-1}\left(\gamma'_{\text{mt}}\, \boldsymbol{z}^{\top}\boldsymbol{v}_{\text{sp}} + b'_{\text{mt}}\right)$, where the latter is a predictor for the main-task label, based on the embeddings projected onto $\boldsymbol{v}_{\text{sp}}$. The model parameters γ_{mt} and b_{mt} are to be trained by minimizing the BCE. We define the difference

$$d_{\text{sp,mt}}^{(\boldsymbol{v}_{\text{sp}})} = \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{sp}}^{(\boldsymbol{v}_{\text{sp}})}, y_{\text{sp}}) - \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{mt}}^{(\boldsymbol{v}_{\text{sp}})}, y_{\text{mt}}),$$

and use this difference to test the following hypotheses

$$H_0 : \mathbb{E}[d_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{sp}})}] = \Delta \quad \text{versus} \quad H_1 : \mathbb{E}[d_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{sp}})}] < \Delta,$$

where the parameter Δ can be used to adjust for the fact that one label is harder to predict than the other. We give an example of its usefulness in Appendix D.5, and a heuristic for setting the parameter value. Under the null hypothesis, the difference in the BCE for the spurious concept and main-task labels, for a logistic regression based on the embeddings projected onto $\boldsymbol{v}_{\text{sp}}$, is Δ. We can use the following test statistic, where under the null hypothesis

$$t_{\text{sp,mt}}^{(\boldsymbol{v}_{\text{sp}})} = \frac{\bar{d}_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{sp}})} - \Delta}{\text{Var}\left(\bar{d}_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{sp}})}\right)} \xrightarrow{d} \mathcal{N}(0,1).$$

We can then conduct the same test, but now for the main-task vector $\boldsymbol{v}_{\text{mt}}$ instead of $\boldsymbol{v}_{\text{sp}}$. Define $\hat{y}_{\text{sp}}^{(\boldsymbol{v}_{\text{mt}})} = \text{Logit}^{-1}\left(\gamma'_{\text{sp}}\, \boldsymbol{z}^{\top}\boldsymbol{v}_{\text{mt}} + b'_{\text{sp}}\right)$. This test uses the following difference

$$d_{\text{sp,mt}}^{(\boldsymbol{v}_{\text{mt}})} = \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{sp}}^{(\boldsymbol{v}_{\text{mt}})}, y_{\text{sp}}) - \mathcal{L}_{\text{BCE}}(\hat{y}_{\text{mt}}^{(\boldsymbol{v}_{\text{mt}})}, y_{\text{mt}}),$$

and the hypotheses become

$$H_0 : \mathbb{E}[d_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{mt}})}] = \Delta \quad \text{versus} \quad H_1 : \mathbb{E}[d_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{mt}})}] > \Delta.$$

Under H_1 we now test if the difference is greater than Δ compared to the previous test. We use the following test statistic:

$$t_{\text{sp,mt}}^{(\boldsymbol{v}_{\text{mt}})} = \frac{\bar{d}_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{mt}})} - \Delta}{\text{Var}\left(\bar{d}_{w,\text{sp,mt}}^{(\boldsymbol{v}_{\text{mt}})}\right)} \xrightarrow{d} \mathcal{N}(0,1).$$

For each of the test statistics, given a large enough sample size, we can acquire our critical values from the standard normal distribution for a given significance level α.

In general, we used the validation set for the test statistics in order to mitigate the effect of overfitting. Unless otherwise mentioned, we set $\alpha = 0.05$ and $\Delta = 0$.

D.4. Weighted Average vs. Unweighted Average for Test Statistics of JSE

As stated in the previous section, we use a weighted average for the test statistics of JSE, where the difference in BCE's is weighted equally across the four combinations of y_{sp} and y_{mt}. We argue that this is helpful in distinguishing whether or not a spurious concept vector $\boldsymbol{v}_{\text{sp}}$ contains a spurious concept or not (vice versa for $\boldsymbol{v}_{\text{mt}}$).

For example, consider a sample where 90% of the water or landbirds coincide with a water or land background ($p_{\text{train}}(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.9$), and we are interested in determining if a vector $\boldsymbol{v}$ contains information about the spurious or main-task features. If $\boldsymbol{v}$ contains information about the background features, a logistic regression for the label y_{sp} based on the embeddings projected onto $\boldsymbol{v}$ will have a low BCE for spurious concept labels. However, this logistic regression will likely also have a low BCE for the main-task labels, due to the correlation between the labels. We can address this problem by weighting the BCE equally across the four combinations of y_{sp} and y_{mt}, where the groups with a small sample (and where y_{sp} and y_{mt} do not coincide) will have a greater influence on the overall average

We verify this argument empirically by comparing two versions of JSE: one with the equally weighted average for the test statistics (as presented in this paper), and one which uses a simple average. Figure 12 shows the results of applying these two versions of the JSE algorithm to the Waterbirds dataset. When the spurious correlation is high, the version with a simple average performs worse in terms of both overall and worst-group accuracy.

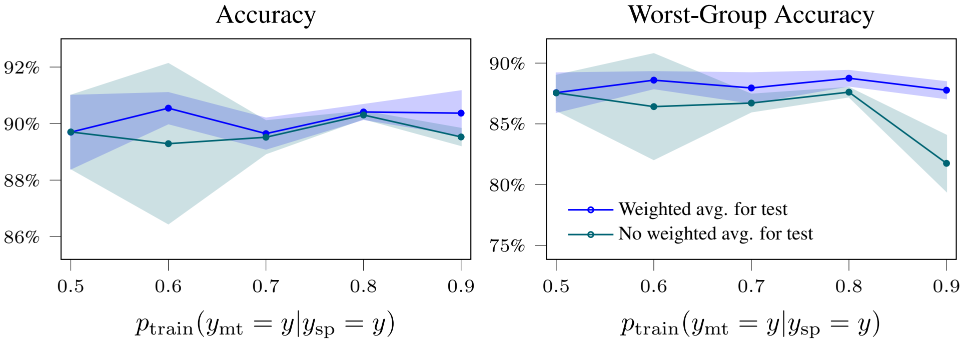

Figure 12. **Effect of using an equally weighted average for the tests of JSE for the Waterbirds dataset**: We plot the (worst-group) accuracy on an OOD test set where $p_{\text{OOD}}(y_{\text{mt}} = y|y_{\text{sp}} = y) = 0.5$, as a function of $p_{\text{train}}(y_{\text{mt}} = y|y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. The shaded area reflects the 95% confidence interval.

D.5. Adjusting for Different Difficulty in Predicting Labels

A potential issue with comparing two BCE's is that it might be fundamentally harder to predict one set of labels over the other. Consider the example of distinguishing cows vs. penguins as given in the introduction. If it is much easier to predict the background than the animal type, then we might wrongly attribute spurious vectors to $\mathcal{Z}_{\text{mt}}$. If the spurious concept is always easier to predict, then even if $\boldsymbol{v}$ represents the main-task features (e.g. the animal shape), the vector might be attributed to the spurious concept subspace, since it has a lower binary cross-entropy for the spurious concept label than the main task label. To address this, we add the Δ term when testing for the second criterion mentioned in Section 3.3. By having a non-zero Δ, we can account for the fact that one binary cross-entropy is always likely to be lower (or higher) than the other.

This naturally leads to the question how one should determine Δ. We provide a simple heuristic. First, we optimize Equation 2 and obtain a first pair of spurious and main-task vectors, $\hat{\boldsymbol{v}}_{\text{sp}}, \hat{\boldsymbol{v}}_{\text{mt}}$. We compare the difference between these two orthogonal vectors through measuring the following:

$$d^*_{\text{sp,mt}} = \mathcal{L}_{\text{BCE}}(\hat{y}^{(\hat{\boldsymbol{v}}_{\text{sp}})}_{\text{sp}}, y_{\text{sp}}) - \mathcal{L}_{\text{BCE}}(\hat{y}^{\hat{\boldsymbol{v}}_{\text{mt}}}_{\text{mt}}, y_{\text{mt}}).$$

We measure the weighted average of this term, defined $\bar{d}^*_{w,\text{sp,mt}}$, for the validation set. This gives an indication if one set of labels is harder to predict than the other, and its value can be used to set Δ. In order to demonstrate the usefulness of Δ and the heuristic, consider the Toy dataset, outlined Section 4.1. In the original set-up, both sets of labels were equally hard to predict, since $\gamma_{\text{sp}} = \gamma_{\text{mt}} = 3$ for both. We change that for this section, and set $\gamma_{\text{sp}} = 6$, $\gamma_{\text{mt}} = 2$, making the spurious concept labels much more separable than the main-task labels.

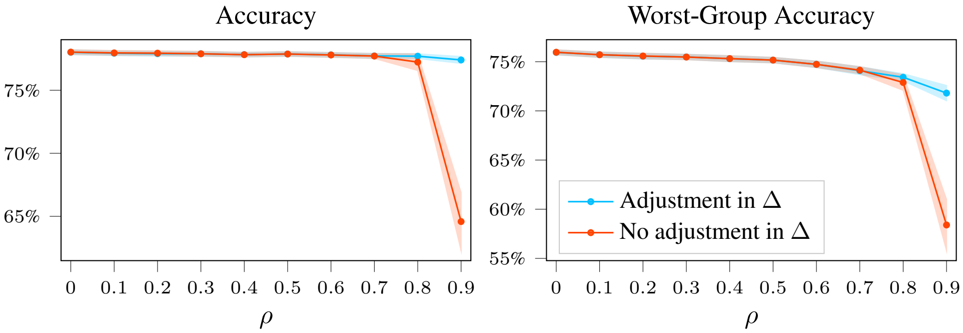

Figure 13. **Effect of adjusting Δ for JSE**: We plot the (worst-group) accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training data. Each accuracy is obtained by averaging over 100 runs. The shaded area reflects the 95% confidence interval.

In Figure 13 we show how JSE performs on a Toy dataset where the separability of the spurious and main-task labels differs. If we do not adjust Δ (e.g. keep it 0), we observe that overall and worst-group accuracy drop once the correlation between the spurious and main-task features becomes high ($\rho = 0.9$). If we adjust Δ via our heuristic, this problem is avoided.

E. Grad-CAM Images for Waterbirds dataset

For each method, we start with the same model, finetuned on the Waterbirds dataset. Then, after applying PCA ($d = 300$), concept removal, and re-training the last linear layer, we apply Grad-CAM to the last convolutional layer of the ResNet50 model. The model and concept-removal methods uses data were $p_{\text{train}}(y_{\text{mt}} = y|y_{\text{sp}} = y) = 0.9$. We use the `Captum` library for calculating the Grad-CAM saliency maps. From each of the four groups in the data (defined by the combinations of $y_{\text{mt}}, y_{\text{sp}}$), we sample 3 images at random. The results are shown in Figure 14. For each group (landbirds on land, landbirds on water, waterbirds on land, waterbirds on water) we observe that JSE ignores the background, while other methods rely on the background for prediction.

F. Concept Removal of Black and White CelebA Images

For each method, we work with 8,000 images in the training set, and 2,000 in the validation set. The black and white images are sampled such that $p(y_{\text{mt}} = y) = 0.5$. We set $p(y_{\text{mt}} = y|y_{\text{sp}} = y) = 0.8$ to create a spurious correlation between the main-task features and spurious concept features. We run JSE, INLP and RLACE with a learning rate of 0.01, weight decay 0f 0.01, and a batch size of 128, and optimize via the outlined training procedure in Section I.2. When using the original dimension (2500), we were unable to get RLACE to converge. For this method, we used PCA to the reduce the dimensionality of the images ($d = 500$) before applying the projection. The vectors are then transformed back to the original dimension. The results are shown for several images across Figure 15 and 16.

G. Comparing JSE to other Instance-reweighting Methods

In this section, we compare JSE to four powerful instance-reweighting methods: (i) group-weighted empirical risk minimization (GW-ERM), which uses a sampling scheme such that the spurious concept and main-task labels are balanced (Idrissi et al., 2022), (ii) group distributional robust optimization (GDRO), which aims to minimize the worst-group loss over possible combinations of the spurious and main-task labels (Sagawa et al., 2020a), (iii) sub-group resampling (SUBG), in which we create a new sample of the dataset where each group is equal in size to the smallest group, and them apply ERM (Sagawa et al., 2020b), and finally (iv) just train twice (JTT), which puts a greater weight on samples that were wrongly identified by an initial model (Liu et al., 2021). This last method does not require the use of spurious concept labels, except for the validation set.

Concept-removal methods such as JSE, but also INLP, RLACE and LEACE are applied post-hoc, e.g. when the full model has already been trained. For a fair comparison in terms of computational resources, each method is applied to the last layer of a model that was finetuned on data with the spurious correlation. Previous work finds that this strategy is particular effective at dealing with spurious correlations (Kirichenko et al., 2023). The results of this comparison with the on benchmark datasets is discussed in Section G.1, and in Section G.2 we compare the methods when limited spurious concept labels are available.

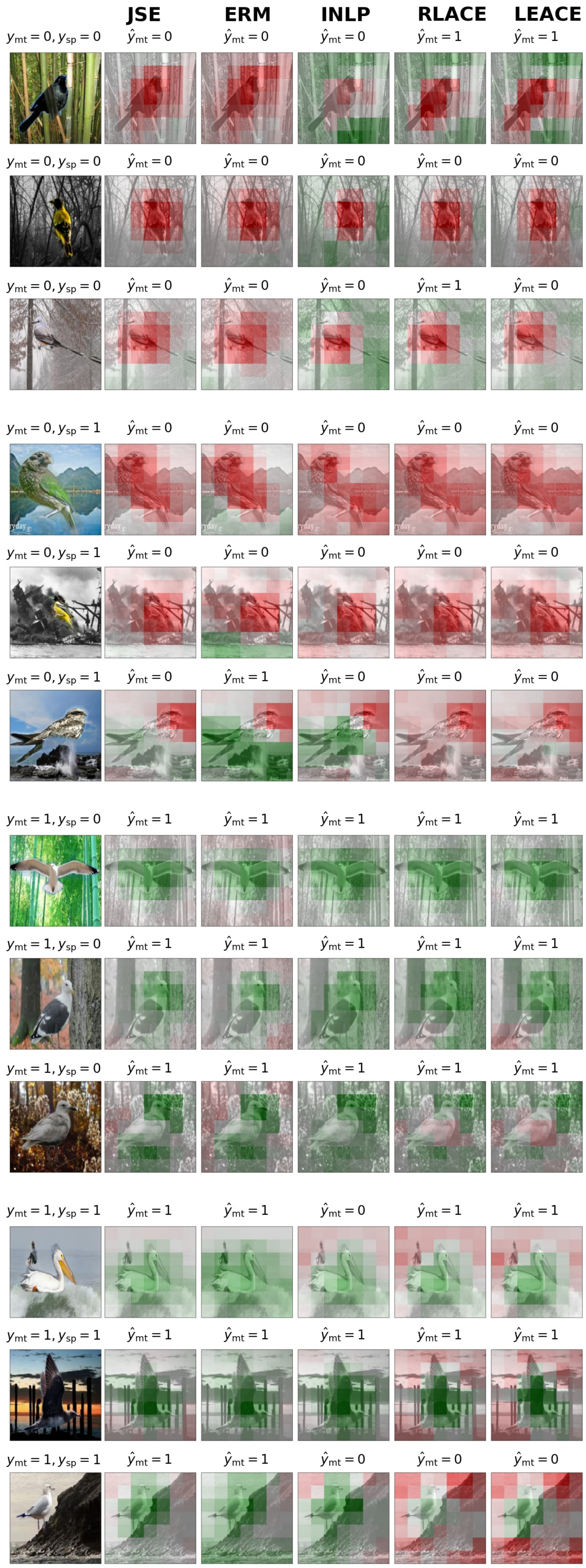
JSE
ERM
INLP
RLACE
LEACE
$y_{\text{mt}} = 0, y_{\text{sp}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$y_{\text{mt}} = 0, y_{\text{sp}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$y_{\text{mt}} = 0, y_{\text{sp}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 0$
$y_{\text{mt}} = 0, y_{\text{sp}} = 1$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$y_{\text{mt}} = 0, y_{\text{sp}} = 1$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$y_{\text{mt}} = 0, y_{\text{sp}} = 1$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$y_{\text{mt}} = 1, y_{\text{sp}} = 0$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$y_{\text{mt}} = 1, y_{\text{sp}} = 0$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$y_{\text{mt}} = 1, y_{\text{sp}} = 0$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$y_{\text{mt}} = 1, y_{\text{sp}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$y_{\text{mt}} = 1, y_{\text{sp}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$y_{\text{mt}} = 1, y_{\text{sp}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 1$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$
$\hat{y}_{\text{mt}} = 0$

Figure 14. **Grad-CAM images for the Waterbirds dataset**: Red (green) patches indicate contribution towards a prediction $y_{\text{mt}} = 0$ ($y_{\text{mt}} = 1$). The model and respective concept-removal methods were trained on data where $p_{\text{train}}(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.9$.

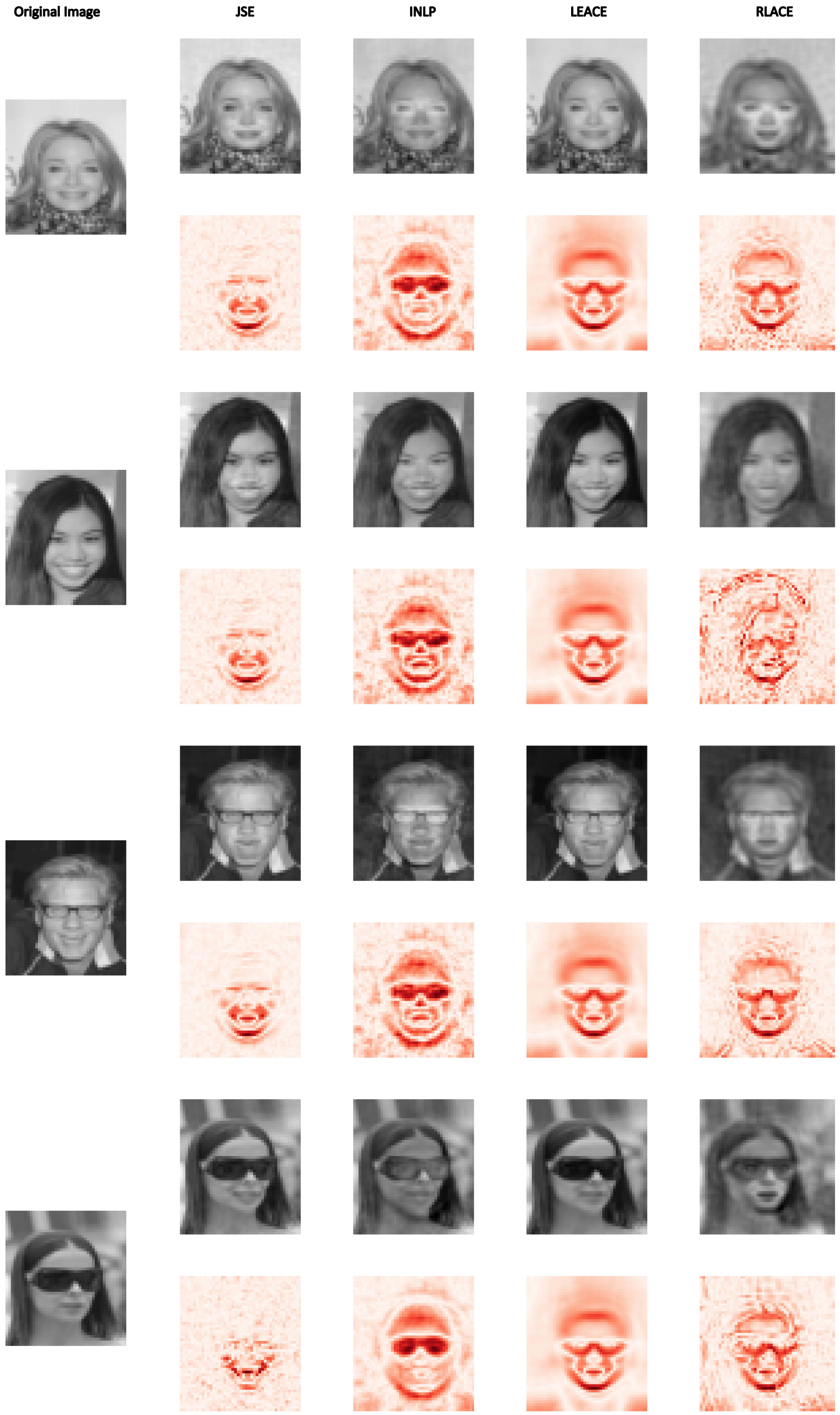

Figure 15. **Application of concept-removal methods to raw pixel data (image 1-4)**: The first row shows the image, after it is transformed by the concept-removal method. The second row shows the absolute difference between the transformed and original image to indicate which pixels have been changed.

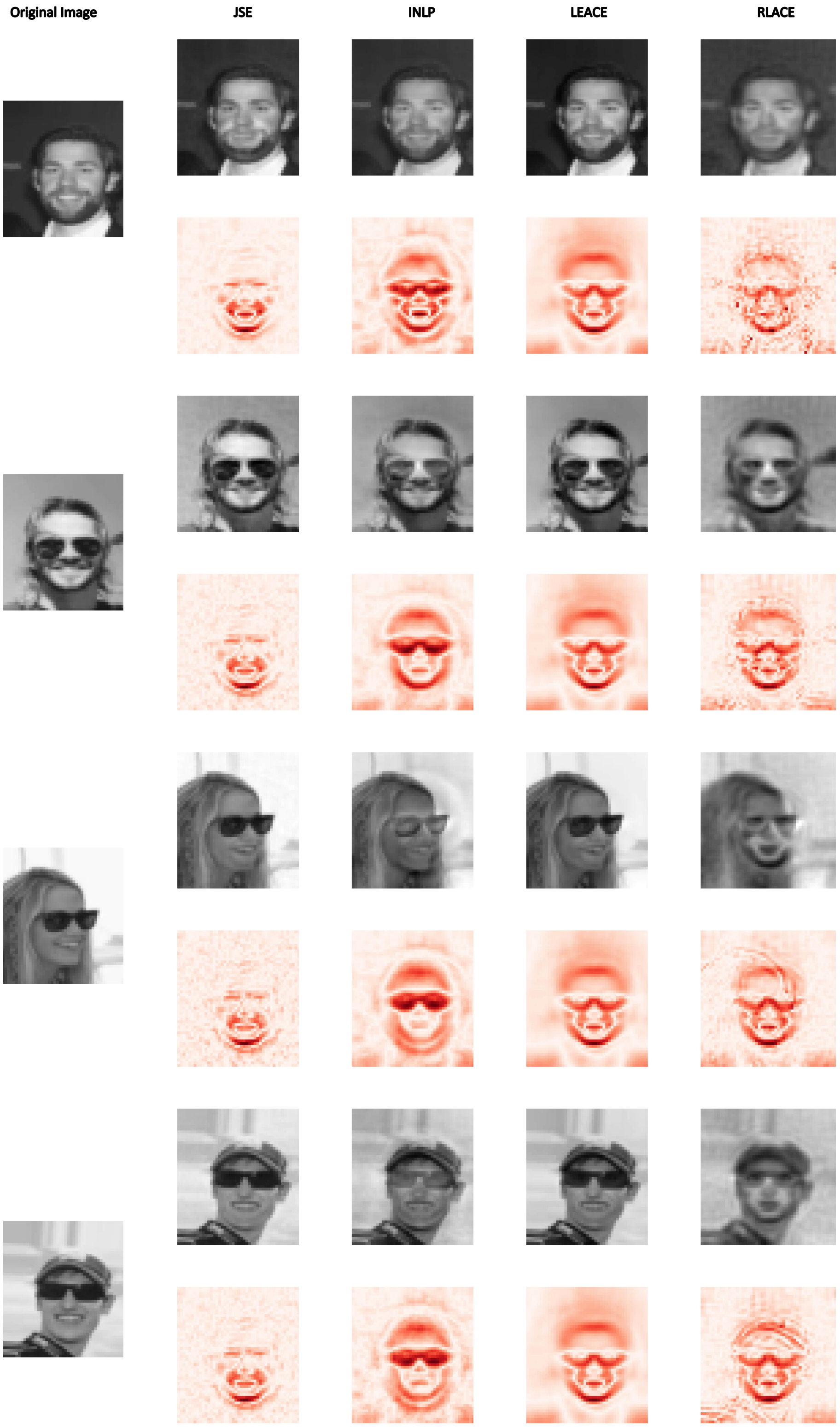

Figure 16. **Application of concept-removal methods to raw pixel data (image 5-8)**: The first row shows the image, after it is transformed by the concept-removal method. The second row shows the absolute difference between the transformed and original image to indicate which pixels have been changed.

G.1. Comparison on Benchmark Datasets

We show the comparison between JSE and instance-reweighting methods with the datasets considered in this paper (Toy, Waterbirds, CelebA, multiNLI) in Figure 17. The numerical results are shown in Table 7-11. JSE is comparable or outperforms other instance-reweighting methods, except in the case where the spurious correlation is very high - e.g. $p(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.95$ for the Waterbirds and CelebA datasets.

We suspect the relatively worse performance of JSE at $p(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.95$ is related to finite-sample estimation noise. If the spurious correlation strength is high, there are relatively less datapoints where the spurious vector is not predictive of the main-task, and vice-versa - making it harder for JSE to separate the spurious and main-task features. We further describe this issue in Section B.2 of the appendix using the Toy dataset.

Furthermore, we emphasize a conceptual difference between JSE and instance-reweighting methods. JSE removes the spurious embedding, which makes it robust to changes at the more fundamental level of the conditional distribution $p(y_{\text{mt}} | \boldsymbol{z}_{\text{sp}})$. The other methods aim for robustness against changes in $p(y_{\text{mt}} | y_{\text{sp}})$. This explains why for the Toy dataset JSE outperforms other methods, because in this setting the conditional distribution $p(y_{\text{mt}} | \boldsymbol{z}_{\text{sp}})$ is modified for the OOD data.

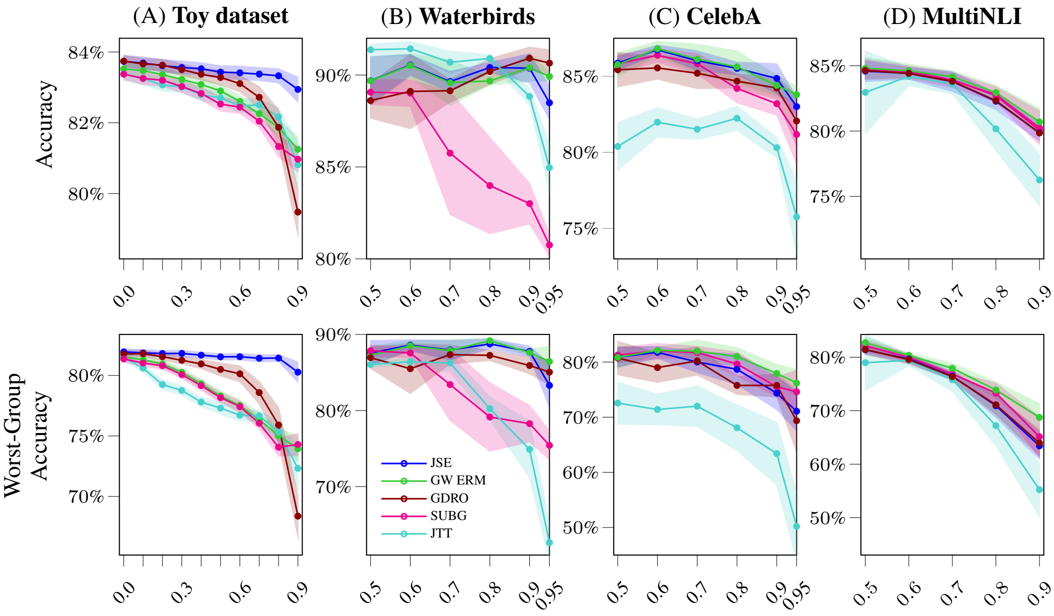

Figure 17. **OOD generalization, compared to instance-reweighting methods**: We plot the (worst-group) accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training set (ρ for the Toy dataset, $p_{\text{train}}(y_{\text{mt}} = y | y_{\text{sp}} = y)$ for the other datasets). Averages based on 100, 5, 5 and 5 runs, respectively. The shaded area reflects the 95% confidence interval.

G.2. Comparison with Limited Spurious Concept Labels

In this section, we evaluate the ability of JSE and instance-reweighting methods to deal with spurious correlations when limited spurious concept labels are available. We compare methods that rely on such labels for training (and thus leave out JTT). In the experiment, there is a limited budget of size n_{sp} to gather spurious concept labels - e.g. one can label 1500, 1000, or 500 images with y_{sp}. The limited dataset (of size n_{sp}) exhibits the same spurious correlation as in the original training set. In the case of GW-ERM, GDRO, this means we can only train on a limited set of labels. For SUBG, this means we still create a dataset where each group has the size of the smallest group - only now based on the smaller dataset. For JSE, we first estimate the spurious concept subspace based on the limited dataset. We use this estimate to remove the spurious concept subspace from all the embeddings, and subsequently train the model to predict y_{mt} on the full dataset.

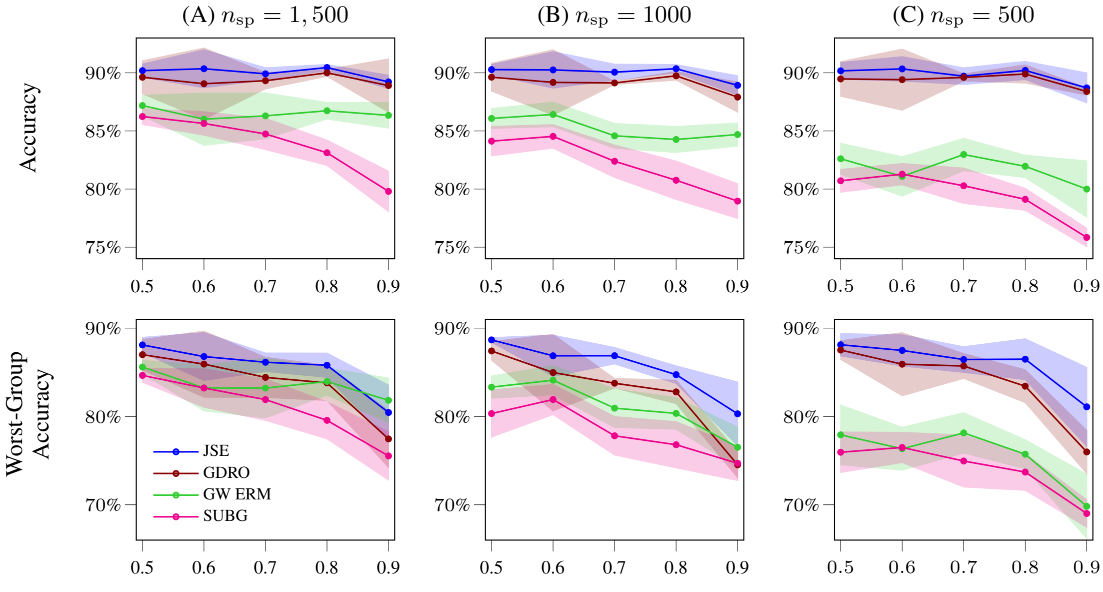

Figure 18. **Performance with limited available spurious concept labels for the Waterbirds dataset**: We plot the (worst-group) accuracy on an OOD test set where $p_{\text{OOD}}(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.5$, as a function of $p_{\text{train}}(y_{\text{mt}} = y | y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. The shaded area reflects the 95% confidence interval.

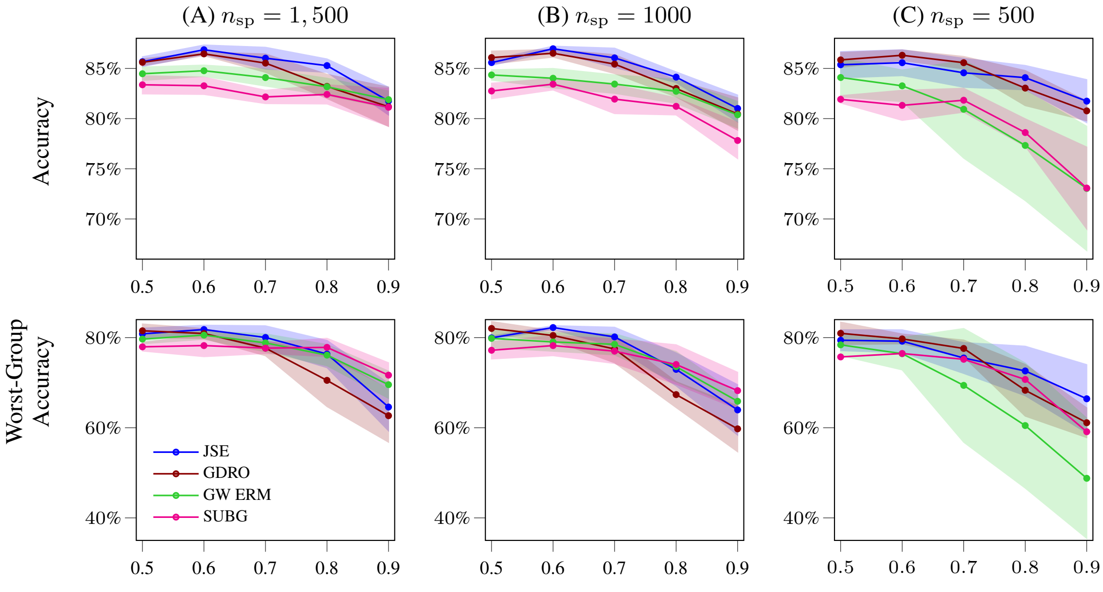

Figure 19. **Performance with limited available spurious concept labels for the CelebA dataset**: We plot the (worst-group) accuracy on an OOD test set where $p_{\text{OOD}}(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.5$, as a function of $p_{\text{train}}(y_{\text{mt}} = y | y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. The shaded area reflects the 95% confidence interval.

In Figures 18 and 19 we compare JSE to instance-reweighting methods on the Waterbirds and CelebA datasets with varying numbers of avilable spurious concept labels. For the Waterbirds dataset, JSE outperforms other methods in overall accuracy. In terms of worst-group accuracy, it is on average slightly better than GW-ERM for $n_{\text{sp}} = 1000$ and $n_{\text{sp}} = 500$. For the CelebA dataset, JSE performs similar or slightly better than GDRO. In terms of worst-group accuracy, GW-ERM and SUBG perform on average better than JSE and GDRO for $n_{\text{sp}} = 1500$ and $n_{\text{sp}} = 1000$, although the variance of the worst-group accuracy is quite high. When $n_{\text{sp}} = 500$, JSE on average outperforms GW-ERM and SUBG in terms of overall and worst-group accuracy. Overall, the results indicate that JSE can have an advantage when limited spurious concept labels are available.

Table 7. **Results of instance-reweighting methods for the Waterbirds dataset**: Table shows the average, worst-group, and per-group accuracy on a test set where $p_{\mathrm{OOD}}(y_{\mathrm{mt}} = y | y_{\mathrm{sp}} = y) = 0.5$, with $y \in \{0, 1\}$, as a function of $p_{\mathrm{train}}(y_{\mathrm{mt}} = y | y_{\mathrm{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. Standard error is reported between brackets.

		$p_{\mathrm{train}}(y_{\mathrm{mt}} = y \| y_{\mathrm{sp}} = y)$					
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9	0.95
JSE	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 0$	90.72 (0.83)	92.43 (0.44)	90.58 (0.27)	91.46 (0.76)	91.85 (0.75)	91.08 (0.64)
	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 1$	88.16 (1.08)	89.05 (0.50)	87.96 (0.64)	89.64 (0.50)	89.56 (0.76)	83.99 (1.71)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 0$	91.43 (0.24)	90.16 (0.46)	91.25 (0.28)	90.25 (0.41)	89.63 (0.81)	84.98 (2.09)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 1$	89.75 (0.57)	89.56 (0.33)	90.69 (0.32)	89.60 (0.53)	88.75 (0.66)	90.44 (0.80)
	Worst-group	87.57 (0.83)	88.60 (0.36)	87.96 (0.64)	88.76 (0.33)	87.77 (0.36)	83.30 (1.46)
	Average	89.70 (0.66)	90.55 (0.28)	89.64 (0.28)	90.41 (0.13)	90.37 (0.40)	88.49 (0.46)
GW-ERM	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 0$	90.16 (0.71)	91.95 (0.50)	89.35 (1.28)	89.42 (0.28)	91.29 (0.77)	92.57 (0.71)
	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 1$	88.79 (1.21)	89.34 (0.58)	89.16 (0.66)	89.61 (0.11)	90.42 (0.66)	88.57 (0.33)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 0$	91.71 (0.29)	90.65 (0.33)	91.87 (0.57)	90.93 (0.10)	89.69 (0.56)	87.04 (1.30)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 1$	89.13 (0.63)	89.53 (0.57)	89.63 (0.16)	89.60 (0.16)	87.91 (0.48)	88.26 (0.24)
	Worst-group	87.27 (0.60)	88.51 (0.33)	87.89 (0.72)	89.18 (0.18)	87.62 (0.22)	86.44 (1.02)
	Average	89.69 (0.66)	90.52 (0.33)	89.59 (0.63)	89.68 (0.12)	90.40 (0.38)	89.92 (0.13)
SUBG	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 0$	88.86 (0.54)	89.58 (0.55)	84.06 (2.29)	79.42 (2.53)	78.28 (1.22)	76.76 (1.40)
	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 1$	88.87 (0.57)	88.20 (0.56)	84.24 (2.52)	84.39 (1.37)	83.48 (0.59)	79.28 (1.30)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 0$	90.81 (0.39)	90.16 (0.50)	92.74 (0.57)	92.49 (0.67)	92.12 (0.45)	91.68 (0.26)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 1$	88.75 (0.33)	88.79 (0.55)	90.00 (0.96)	90.09 (0.97)	88.85 (0.46)	89.00 (0.97)
	Worst-group	87.86 (0.37)	87.57 (0.32)	83.41 (2.41)	79.15 (2.30)	78.28 (1.22)	75.42 (0.98)
	Average	89.07 (0.34)	89.02 (0.37)	85.75 (1.71)	83.99 (1.34)	83.01 (0.56)	80.75 (0.32)
GDRO	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 0$	88.20 (1.11)	89.35 (2.64)	89.60 (0.70)	92.46 (0.75)	94.02 (0.61)	93.53 (0.99)
	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 1$	87.73 (0.56)	88.31 (1.01)	87.95 (0.85)	88.44 (1.01)	90.28 (0.82)	90.30 (0.52)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 0$	91.65 (0.26)	90.75 (0.69)	91.28 (0.41)	88.75 (0.70)	86.17 (0.68)	85.39 (1.17)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 1$	90.12 (0.42)	89.44 (0.69)	89.63 (0.70)	89.81 (0.82)	86.98 (0.65)	87.01 (0.67)
	Worst-group	86.94 (0.71)	85.50 (1.66)	87.35 (0.62)	87.26 (0.34)	85.92 (0.55)	85.05 (0.85)
	Average	88.61 (0.49)	89.11 (1.03)	89.14 (0.32)	90.19 (0.36)	90.91 (0.30)	90.65 (0.37)
JTT	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 0$	92.63 (0.37)	95.14 (0.37)	94.29 (0.38)	97.08 (0.21)	98.16 (0.26)	98.93 (0.12)
	$y_{\mathrm{mt}} = 0, y_{\mathrm{sp}} = 1$	92.53 (0.22)	90.12 (0.58)	88.52 (0.72)	87.96 (0.29)	82.94 (0.90)	75.34 (1.98)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 0$	88.26 (0.31)	86.54 (0.30)	86.64 (0.74)	80.25 (0.77)	74.89 (1.89)	62.65 (1.26)
	$y_{\mathrm{mt}} = 1, y_{\mathrm{sp}} = 1$	86.04 (0.14)	87.88 (0.44)	89.75 (0.32)	90.16 (0.46)	90.75 (0.47)	91.99 (0.57)
	Worst-group	86.04 (0.14)	86.42 (0.23)	86.25 (0.61)	80.25 (0.77)	74.89 (1.89)	62.65 (1.26)
	Average	91.38 (0.15)	91.43 (0.20)	90.69 (0.31)	90.90 (0.10)	88.84 (0.23)	84.96 (0.67)

Table 8. **Results instance-reweighting methods for the CelebA dataset**: Table shows the average, worst-group, and per-group accuracy on a test set where $p_{\text{OOD}}(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.5$, with $y \in \{0, 1\}$, as a function of $p_{\text{train}}(y_{\text{mt}} = y | y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. Standard error is reported between brackets.

		$p_{\text{train}}(y_{\text{mt}} = y \| y_{\text{sp}} = y)$					
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9	0.95
JSE	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.40 (0.35)	87.56 (0.71)	87.52 (0.36)	88.32 (1.57)	90.36 (0.43)	91.08 (0.64)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	82.72 (1.23)	82.24 (0.56)	81.12 (1.18)	82.56 (0.77)	81.84 (1.37)	78.72 (0.92)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.04 (0.70)	84.48 (1.23)	82.72 (1.10)	79.12 (1.74)	74.40 (1.55)	71.08 (1.80)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	92.32 (0.37)	92.68 (0.30)	92.76 (0.23)	92.08 (0.57)	92.84 (0.37)	91.16 (0.53)
	Worst-group	80.88 (0.90)	81.76 (0.56)	80.00 (0.96)	78.68 (1.41)	74.36 (1.51)	71.08 (1.80)
	Average	85.87 (0.23)	86.74 (0.15)	86.03 (0.33)	85.52 (0.21)	84.86 (0.49)	83.01 (0.30)
GW-ERM	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.63 (0.87)	87.64 (0.50)	85.88 (0.88)	86.72 (0.98)	86.64 (0.62)	88.08 (0.55)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.05 (1.15)	82.72 (0.67)	82.68 (1.39)	83.08 (0.76)	81.36 (1.46)	78.36 (0.67)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.66 (1.52)	84.88 (1.20)	84.80 (1.08)	81.00 (0.81)	78.56 (0.84)	76.76 (1.50)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	92.08 (0.46)	92.00 (0.40)	91.16 (0.63)	91.68 (0.42)	91.04 (0.75)	92.00 (1.10)
	Worst-group	80.85 (1.01)	82.12 (0.30)	81.96 (1.02)	81.00 (0.81)	77.92 (0.85)	76.20 (1.26)
	Average	85.77 (0.39)	86.81 (0.25)	86.13 (0.50)	85.62 (0.54)	84.40 (0.34)	83.80 (0.28)
SUBG	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.72 (0.71)	87.00 (0.67)	85.60 (0.75)	85.44 (0.61)	86.64 (0.60)	83.72 (0.61)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	82.92 (0.62)	83.76 (0.62)	82.84 (0.85)	82.20 (0.59)	80.96 (1.13)	77.96 (0.56)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	81.76 (0.79)	83.40 (1.51)	84.04 (1.31)	79.76 (1.18)	75.52 (0.92)	75.12 (2.23)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	91.92 (0.37)	91.36 (0.74)	90.84 (0.26)	89.40 (0.64)	89.68 (0.82)	87.92 (0.95)
	Worst-group	81.28 (0.68)	82.00 (0.70)	81.72 (0.56)	79.64 (1.10)	75.52 (0.92)	74.60 (1.98)
	Average	85.83 (0.30)	86.38 (0.23)	85.83 (0.34)	84.20 (0.48)	83.20 (0.39)	81.18 (0.99)
GDRO	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	85.16 (1.45)	88.28 (0.67)	85.68 (1.14)	88.96 (0.80)	88.12 (0.94)	90.64 (1.20)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	82.88 (1.29)	80.68 (0.34)	80.84 (1.54)	80.76 (0.70)	80.68 (1.31)	73.76 (2.07)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.56 (1.62)	80.36 (2.00)	82.20 (1.80)	75.76 (0.83)	75.76 (1.06)	69.40 (2.91)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	91.12 (0.27)	92.88 (0.61)	92.08 (0.95)	93.24 (0.49)	92.36 (0.92)	94.44 (0.79)
	Worst-group	80.72 (1.54)	79.00 (1.37)	80.20 (1.31)	75.76 (0.83)	75.76 (1.06)	69.36 (2.88)
	Average	85.43 (0.57)	85.55 (0.38)	85.20 (0.51)	84.68 (0.23)	84.23 (0.38)	82.06 (0.84)
JTT	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	82.16 (1.64)	85.64 (1.09)	85.48 (1.27)	89.20 (0.49)	91.04 (0.50)	94.68 (0.63)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	83.04 (0.74)	83.80 (1.06)	79.08 (1.11)	79.92 (0.71)	71.72 (0.85)	63.04 (2.13)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	72.56 (1.95)	71.40 (1.41)	72.00 (1.87)	68.08 (2.12)	63.40 (2.92)	50.20 (3.13)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.80 (2.30)	87.08 (0.62)	89.52 (1.29)	91.76 (0.44)	95.08 (0.77)	95.12 (0.45)
	Worst-group	72.56 (1.95)	71.40 (1.41)	72.00 (1.87)	68.08 (2.12)	63.40 (2.92)	50.20 (3.13)
	Average	80.39 (0.80)	81.98 (0.49)	81.52 (0.34)	82.24 (0.41)	80.31 (0.35)	75.76 (1.23)

Table 9. **Results of instance-reweighting methods for the MultiNLI dataset**: Table shows the average, worst-group, and per-group accuracy on a test set where $p_{\text{OOD}}(y_{\text{mt}} = y|y_{\text{sp}} = y) = 0.5$, with $y \in \{0, 1\}$, as a function of $p_{\text{train}}(y_{\text{mt}} = y|y_{\text{sp}} = y)$. Each accuracy is obtained by averaging over 5 runs. Standard error is reported between brackets.

		$p_{\text{train}}(y_{\text{mt}} = y\|y_{\text{sp}} = y)$				
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9
GW-ERM	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.51 (0.50)	88.69 (0.46)	90.21 (0.38)	91.34 (0.40)	92.54 (0.54)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	86.05 (0.51)	83.55 (0.74)	81.20 (0.64)	77.87 (1.27)	71.73 (1.87)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	83.46 (0.68)	80.35 (0.21)	77.94 (0.47)	73.87 (0.76)	68.75 (1.31)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.22 (0.43)	85.87 (0.49)	87.28 (0.38)	88.69 (0.78)	89.79 (0.93)
	Worst-group	82.72 (0.54)	80.35 (0.21)	77.94 (0.47)	73.87 (0.76)	68.75 (1.31)
	Average	84.81 (0.42)	84.62 (0.26)	84.16 (0.22)	82.94 (0.28)	80.70 (0.52)
SUBG	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	86.94 (0.51)	88.72 (0.50)	90.54 (0.34)	91.44 (0.78)	93.60 (0.62)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	86.56 (0.30)	83.52 (0.70)	81.01 (0.85)	78.61 (1.56)	73.41 (1.54)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.70 (0.76)	79.97 (0.36)	76.88 (0.22)	73.31 (0.93)	65.14 (1.92)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	82.75 (0.39)	85.84 (0.45)	87.18 (0.46)	87.81 (0.93)	88.86 (0.87)
	Worst-group	82.13 (0.55)	79.97 (0.36)	76.88 (0.22)	73.31 (0.93)	65.14 (1.92)
	Average	84.74 (0.39)	84.51 (0.24)	83.90 (0.29)	82.79 (0.30)	80.25 (0.66)
GDRO	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	87.47 (0.66)	89.07 (0.64)	90.75 (0.29)	92.62 (0.34)	94.05 (0.46)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	87.50 (0.40)	84.10 (0.70)	81.17 (0.95)	77.49 (1.20)	71.70 (1.17)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	81.84 (0.54)	79.55 (0.41)	76.37 (0.33)	71.10 (0.90)	63.97 (1.25)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	81.71 (0.43)	84.93 (0.62)	86.94 (0.82)	88.11 (0.60)	89.70 (0.86)
	Worst-group	81.42 (0.44)	79.55 (0.41)	76.37 (0.33)	71.10 (0.90)	63.97 (1.25)
	Average	84.63 (0.42)	84.41 (0.28)	83.81 (0.34)	82.33 (0.32)	79.85 (0.41)
JTT	$y_{\text{mt}} = 0, y_{\text{sp}} = 0$	83.52 (1.64)	88.29 (0.83)	90.99 (0.26)	92.10 (2.08)	96.11 (0.83)
	$y_{\text{mt}} = 0, y_{\text{sp}} = 1$	82.42 (2.40)	81.97 (1.03)	78.05 (1.33)	70.02 (2.20)	57.92 (3.39)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 0$	82.82 (3.15)	79.95 (0.48)	76.00 (1.06)	68.58 (2.23)	56.14 (2.28)
	$y_{\text{mt}} = 1, y_{\text{sp}} = 1$	83.09 (2.65)	86.54 (0.66)	89.34 (0.32)	90.00 (2.81)	94.85 (0.66)
	Worst-group	78.98 (2.64)	79.41 (0.50)	75.78 (1.10)	67.20 (1.93)	55.23 (2.55)
	Average	82.96 (1.61)	84.19 (0.36)	83.60 (0.38)	80.17 (0.84)	76.26 (1.01)

Table 10. **Results of instance-reweighting methods for the Toy dataset for** $\rho \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$. Table shows the average, worst-group, and per-group accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training data. Each accuracy is obtained by averaging over 100 runs. Standard error is reported between brackets.

				ρ		
Method	**Accuracy**	0.0	0.1	0.2	0.3	0.4
GW-ERM	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 0$	83.26 (0.20)	82.64 (0.19)	82.15 (0.21)	81.25 (0.22)	80.44 (0.25)
	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 1$	83.56 (0.17)	83.96 (0.19)	84.55 (0.18)	84.69 (0.19)	85.47 (0.18)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 0$	83.79 (0.19)	84.25 (0.20)	84.56 (0.17)	85.13 (0.19)	85.57 (0.20)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 1$	83.45 (0.17)	83.05 (0.18)	82.21 (0.19)	81.82 (0.18)	80.87 (0.21)
	Worst-group	81.52 (0.14)	81.30 (0.14)	80.95 (0.17)	80.26 (0.17)	79.31 (0.19)
	Average	83.52 (0.09)	83.48 (0.08)	83.36 (0.08)	83.22 (0.09)	83.08 (0.09)
SUBG	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 0$	83.54 (0.20)	82.66 (0.20)	82.21 (0.20)	81.57 (0.20)	80.64 (0.22)
	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 1$	83.42 (0.20)	83.80 (0.19)	84.22 (0.20)	84.86 (0.24)	85.00 (0.22)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 0$	83.11 (0.17)	83.84 (0.18)	84.21 (0.18)	84.46 (0.21)	85.17 (0.21)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 1$	83.43 (0.17)	82.73 (0.19)	82.18 (0.21)	81.25 (0.21)	80.56 (0.26)
	Worst-group	81.37 (0.14)	81.03 (0.13)	80.81 (0.15)	80.07 (0.16)	79.16 (0.20)
	Average	83.37 (0.08)	83.26 (0.09)	83.20 (0.09)	83.02 (0.09)	82.83 (0.09)
GDRO	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 0$	83.63 (0.16)	83.49 (0.15)	83.57 (0.17)	83.44 (0.20)	83.45 (0.21)
	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 1$	83.93 (0.18)	83.78 (0.18)	83.81 (0.20)	83.49 (0.21)	83.13 (0.25)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 0$	83.54 (0.20)	83.54 (0.20)	83.42 (0.21)	83.21 (0.22)	82.91 (0.26)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 1$	83.81 (0.19)	83.80 (0.17)	83.66 (0.20)	83.82 (0.21)	83.94 (0.22)
	Worst-group	81.78 (0.15)	81.81 (0.14)	81.55 (0.14)	81.25 (0.16)	80.95 (0.20)
	Average	83.74 (0.08)	83.66 (0.08)	83.63 (0.08)	83.50 (0.08)	83.37 (0.08)
JTT	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 0$	83.23 (0.18)	81.81 (0.23)	80.60 (0.23)	80.07 (0.24)	79.59 (0.30)
	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 1$	83.71 (0.17)	84.83 (0.18)	85.80 (0.18)	86.06 (0.19)	86.22 (0.25)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 0$	83.12 (0.18)	84.34 (0.20)	85.21 (0.17)	85.64 (0.20)	85.77 (0.25)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 1$	83.43 (0.19)	81.99 (0.17)	80.66 (0.25)	80.27 (0.25)	79.60 (0.30)
	Worst-group	81.49 (0.13)	80.60 (0.16)	79.25 (0.20)	78.76 (0.19)	77.80 (0.20)
	Average	83.38 (0.08)	83.25 (0.09)	83.07 (0.09)	83.01 (0.08)	82.80 (0.09)

Table 11. **Results of instance-reweighting methods for the Toy dataset for** $\rho \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. Table shows the average, worst-group, and per-group accuracy on a test set without spurious correlation, as a function of the spurious correlation in the training data. Each accuracy is obtained by averaging over 100 runs. Standard error is reported between brackets.

				ρ		
Method	**Accuracy**	0.5	0.6	0.7	0.8	0.9
GW-ERM	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 0$	79.55 (0.30)	78.88 (0.30)	78.37 (0.41)	77.57 (0.46)	79.78 (0.57)
	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 1$	85.65 (0.23)	85.63 (0.27)	85.73 (0.40)	85.59 (0.50)	82.78 (0.78)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 0$	86.05 (0.22)	86.19 (0.29)	86.06 (0.35)	85.99 (0.47)	82.69 (0.80)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 1$	80.33 (0.24)	79.72 (0.33)	78.90 (0.37)	78.28 (0.45)	79.68 (0.59)
	Worst-group	78.31 (0.22)	77.52 (0.24)	76.33 (0.29)	75.03 (0.31)	73.94 (0.53)
	Average	82.90 (0.09)	82.60 (0.08)	82.26 (0.10)	81.86 (0.11)	81.25 (0.18)
SUBG	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 0$	80.03 (0.28)	78.79 (0.27)	77.71 (0.31)	77.63 (0.46)	78.75 (0.48)
	$y_{\mathrm{mt}} = 0$, $y_{\mathrm{sp}} = 1$	85.19 (0.28)	86.09 (0.24)	86.44 (0.31)	85.10 (0.62)	83.39 (0.71)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 0$	85.02 (0.28)	86.15 (0.23)	86.44 (0.25)	85.29 (0.53)	83.22 (0.68)
	$y_{\mathrm{mt}} = 1$, $y_{\mathrm{sp}} = 1$	79.92 (0.27)	78.79 (0.29)	77.68 (0.33)	77.42 (0.49)	78.57 (0.49)
	Worst-group	78.17 (0.20)	77.42 (0.23)	76.05 (0.25)	74.07 (0.37)	74.29 (0.46)
	Average	82.53 (0.10)	82.44 (0.09)	82.04 (0.09)	81.33 (0.15)	80.97 (0.18)
GDRO	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 0$	83.64 (0.24)	83.89 (0.23)	84.62 (0.29)	85.70 (0.30)	88.46 (0.29)
	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 1$	83.01 (0.34)	82.25 (0.39)	80.94 (0.50)	77.92 (0.78)	69.99 (0.90)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 0$	82.51 (0.33)	82.07 (0.36)	80.51 (0.55)	77.70 (0.76)	70.26 (0.85)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 1$	83.93 (0.25)	84.20 (0.28)	84.74 (0.31)	86.10 (0.33)	89.00 (0.28)
	Worst-group	80.50 (0.27)	80.14 (0.33)	78.59 (0.47)	75.90 (0.73)	68.37 (0.87)
	Average	83.28 (0.10)	83.11 (0.12)	82.72 (0.15)	81.87 (0.25)	79.48 (0.32)
JTT	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 0$	79.66 (0.33)	78.52 (0.29)	78.66 (0.34)	79.67 (0.48)	82.96 (0.56)
	$y_{\mathrm{mt}} = 0$ and $y_{\mathrm{sp}} = 1$	85.97 (0.31)	86.65 (0.29)	86.70 (0.25)	84.78 (0.57)	78.26 (1.14)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 0$	85.42 (0.36)	86.13 (0.34)	85.97 (0.30)	84.26 (0.67)	78.29 (1.11)
	$y_{\mathrm{mt}} = 1$ and $y_{\mathrm{sp}} = 1$	79.71 (0.36)	78.64 (0.32)	78.72 (0.32)	79.92 (0.46)	83.64 (0.57)
	Worst-group	77.30 (0.25)	76.71 (0.25)	76.66 (0.23)	75.33 (0.40)	72.32 (0.90)
	Average	82.70 (0.09)	82.48 (0.09)	82.50 (0.09)	82.17 (0.13)	80.81 (0.32)

H. JSE and Multiple Spurious Concepts

In this section we aim to provide a brief example of how one can use JSE to deal with multiple spurious concepts. In order to do so, we conduct an experiment on the CelebA dataset. Here, the first spurious concept $y_{\mathrm{sp}}^{(1)} \in \{0, 1\}$ is the sex of the person in the image (female-male). We take the version of the dataset where the spurious correlation between the main-task and spurious concept of gender is $p(y_{\mathrm{mt}} = y | y_{\mathrm{sp}}^{(1)} = y) = 0.8$. In the test set, $p(y_{\mathrm{mt}} = y | y_{\mathrm{sp}}^{(1)} = y) = 0.5$. The second spurious concept $y_{\mathrm{sp}}^{(2)} \in \{0, 1\}$ indicates whether or not someone wears glasses or not. This second spurious concept also happens to be spuriously correlated with the main-task of blond vs. non-blond. For instance, in this dataset, $p(y_{\mathrm{mt}} = 1 | y_{\mathrm{sp}}^{(2)} = 1) = 0.18$, and $p(y_{\mathrm{mt}} = 0 | y_{\mathrm{sp}}^{(2)} = 1) = 0.82$.

To remove the influence of both concepts, we first apply JSE to estimate the subspace related to $y_{\mathrm{sp}}^{(1)}$, and then related to $y_{\mathrm{sp}}^{(2)}$. Then, we project the embeddings on the orthogonal complement of both subspaces. For determining the hyper-parameters of JSE in each instance, we use the same approach as our experiments on OOD generalization. We compare this approach to using standard ERM, as well as group-weighted ERM (GW-ERM). With group-weighted ERM, the groups are defined over all 8 combinations of $y_{\mathrm{mt}}, y_{\mathrm{sp}}^{(1)}, y_{\mathrm{sp}}^{(2)}$.

The results of this experiment are reported in Table 12. If JSE is applied for both spurious concepts, it improves upon ERM for worst-group accuracy for both spurious concepts. There appears to be a small trade-off with only intervening for a single concept. For instance, if we only remove the subspace related to $y_{\mathrm{sp}}^{(1)}$, the average worst-group accuracy (over combinations of $y_{\mathrm{mt}}, y_{\mathrm{sp}}^{(1)}$) is 80.04, vs. 78.72 if we intervene for both spurious concepts.

Table 12. **Results of applying JSE to multiple spurious concepts:** Table shows the average worst-group, and per-group accuracy on a test set where $p(y_{\text{mt}} = y | y_{\text{sp}}^{(1)} = y) = 0.5$. In the training set, $p(y_{\text{mt}} = y | y_{\text{sp}}^{(1)} = y) = 0.8$. The worst-group accuracy is either taken over combinations of $y_{\text{mt}}, y_{\text{sp}}^{(1)}$ or $y_{\text{mt}}, y_{\text{sp}}^{(2)}$. The n refers to the number of samples in the test set. We show three versions of JSE: when removing the subspace related to $y_{\text{sp}}^{(1)}$, $y_{\text{sp}}^{(2)}$, and both. The learning rate used for JSE to remove $y_{\text{sp}}^{(1)}$ was 0.001, with a weight decay of 0.001, and for removing $y_{\text{sp}}^{(2)}$ it was 0.01, with a weight decay of 0.01. Each accuracy is obtained by averaging over 5 runs. Standard error is reported in brackets.

Accuracy type	n	**Both** $y_{\text{sp}}^{(1)}$**,** $y_{\text{sp}}^{(2)}$	**Only for** $y_{\text{sp}}^{(1)}$	**Only for** $y_{\text{sp}}^{(2)}$	**ERM**	**GW-ERM**
Overall	2000	85.6 (0.2)	85.99 (0.24)	84.02 (0.29)	83.68 (0.22)	84.45 (0.57)
$y_{\text{mt}} = 1, y_{\text{sp}}^{(1)} = 1$	500	92.16 (0.72)	92.2 (0.6)	95.84 (0.33)	95.88 (0.45)	89.64 (0.49)
$y_{\text{mt}} = 1, y_{\text{sp}}^{(1)} = 0$	500	78.72 (0.69)	80.24 (1.12)	68.64 (0.7)	67.8 (0.64)	80.0 (0.48)
$y_{\text{mt}} = 0, y_{\text{sp}}^{(1)} = 1$	500	82.64 (0.53)	83.08 (0.39)	77.68 (0.87)	76.96 (0.68)	81.56 (0.73)
$y_{\text{mt}} = 0, y_{\text{sp}}^{(1)} = 0$	500	88.88 (0.99)	88.44 (0.92)	93.92 (0.5)	94.08 (0.59)	86.6 (1.96)
Worst-group $(y_{\text{mt}}, y_{\text{sp}}^{(1)})$	-	78.72 (0.69)	80.04 (0.98)	68.64 (0.7)	67.8 (0.64)	79.36 (0.39)
$y_{\text{mt}} = 1, y_{\text{sp}}^{(2)} = 1$	30	73.11 (3.6)	73.5 (3.4)	75.04 (2.65)	63.96 (2.34)	83.5 (2.66)
$y_{\text{mt}} = 1, y_{\text{sp}}^{(2)} = 0$	970	86.0 (0.15)	86.81 (0.39)	82.58 (0.43)	82.64 (0.61)	84.93 (0.33)
$y_{\text{mt}} = 0, y_{\text{sp}}^{(2)} = 1$	75	83.94 (4.65)	84.43 (2.96)	88.65 (2.7)	90.38 (2.76)	83.78 (3.88)
$y_{\text{mt}} = 0, y_{\text{sp}}^{(2)} = 0$	925	85.95 (0.33)	85.9 (0.44)	85.55 (0.56)	85.1 (0.41)	84.14 (1.08)
Worst-group $(y_{\text{mt}}, y_{\text{sp}}^{(2)})$	-	69.04 (2.82)	71.84 (3.04)	74.15 (2.25)	63.96 (2.34)	78.74 (2.81)

I. Details on Datasets, Models, and Parameter Selection

I.1. Datasets

Toy: For a given dataset size (e.g. $n =$2,000) the data is split into an 80% training and 20% validation set, and a test set of the same size is kept apart for evaluation. We describe the data-generating process of this dataset in Section 4.1.

Waterbirds: this dataset from Sagawa et al. (2020b) is a combination of the Places dataset (Zhou et al., 2016) and the CUB dataset (Welinder et al., 2010). A 'water background' is set by selecting an image from the lake and ocean categories in the places dataset, and the 'land background' is set based on the broadleaf and bamboo forest categories. A waterbird/land is then pasted in front of the background. When creating new versions of the dataset, we change the $p(y_{\text{mt}} = y | y_{\text{sp}} = y)$, and keep the size of the training set at 4,775 samples, and 1,199 for the validation set. For the test set, we select 5,796 samples where $p(y_{\text{mt}} = y | y_{\text{sp}} = y) = 0.5$.

For this dataset when training ERM or adversarial removal, we sample in each batch such that $p(y_{\text{sp}} = 1) = 0.5$. When training JSE, INLP or RLACE, in each batch we sample such that $p(y_{\text{sp}} = 1) = 0.5$. When training ERM on the embeddings transformed by JSE, INLP or RLACE, we sample again such that in each batch $p(y_{\text{mt}} = 1) = 0.5$.

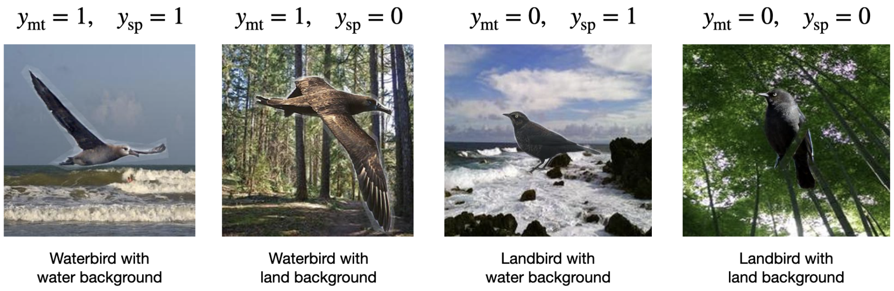
$y_{\text{mt}} = 1, \quad y_{\text{sp}} = 1$
$y_{\text{mt}} = 1, \quad y_{\text{sp}} = 0$
$y_{\text{mt}} = 0, \quad y_{\text{sp}} = 1$
$y_{\text{mt}} = 0, \quad y_{\text{sp}} = 0$
Waterbird with water background
Waterbird with land background
Landbird with water background
Landbird with land background

Figure 20. **Examples of the different images in the Waterbirds dataset**

CelebA: this dataset contains images of celebrity faces (Liu et al., 2015). The total size of the dataset is 202,599, from which we sample smaller versions in order to control the strength of the spurious correlation. For these smaller versions, we select 4,500 observations for the training set, 2,000 for the validation set, and 2,000 for the test set. We set the $p(y_{\mathrm{mt}} = y|y_{\mathrm{sp}} = y)$ for the training and validation set, while we set $p(y_{\mathrm{mt}} = y|y_{\mathrm{sp}} = y) = 0.5$ for the test set. We set $p(y_{\mathrm{mt}} = 1) = 0.5$ for the training, validation and test set.

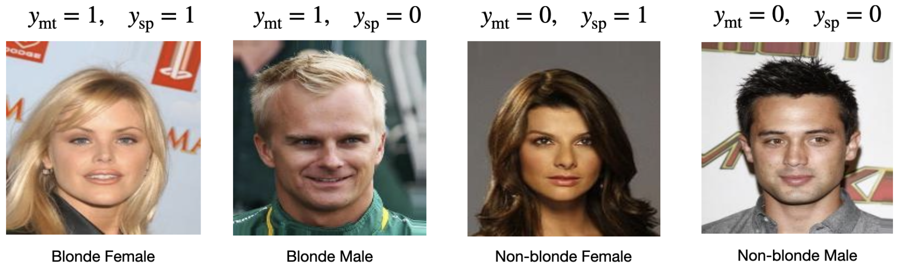
$y_{\mathrm{mt}} = 1, \quad y_{\mathrm{sp}} = 1$
$y_{\mathrm{mt}} = 1, \quad y_{\mathrm{sp}} = 0$
$y_{\mathrm{mt}} = 0, \quad y_{\mathrm{sp}} = 1$
$y_{\mathrm{mt}} = 0, \quad y_{\mathrm{sp}} = 0$
Blonde Female
Blonde Male
Non-blonde Female
Non-blonde Male

Figure 21. **Examples of the different images in the CelebA dataset**

MultiNLI: the MultiNLI dataset (Williams et al., 2018) contains pairs of sentences, with examples shown in Table 13. The sentences are pasted together with a [SEP] token in between. We change the dependent variable to a binary label, with $y_{\mathrm{mt}} = 1$ indicating the first sentence (the premise) contradicting the second sentence (the hypothesis), and y_{sp} otherwise. We chose for the punctuation marks as a spurious correlation rather than the commonly used ‘negation word’ (see for instance Sagawa et al. (2020a)) since it is questionable to what extent a negation word such as ’no’, ’never’ or ’none’ is spuriously associated with a contradiction (Joshi et al., 2022). The original dataset contains 206,175 pairs of sentences. We sample smaller versions of the dataset where $p(y_{\mathrm{mt}} = 1) = 0.5$ and change the $p(y_{\mathrm{mt}} = y|y_{\mathrm{sp}} = y)$ for the training and validation set. For the test set, we set $p(y_{\mathrm{mt}} = y|y_{\mathrm{sp}} = y) = 0.5$. The training set contains 50,000 datapoints and the validation and test set each contain 5,000 datapoints.

Table 13. **Examples from sentence pairs in the MultiNLI dataset.**

y_{sp}	y_{mt}	**Premise**	**Hypothesis**
0	0	Conceptually cream skimming has two basic dimensions - product and geography.	Product and geography are what make cream skimming work !!
1		One of our number will carry out your instructions minutely.	A member of my team will execute your orders with immense precision.
0	1	Fun for adults and children.	Fun for only children.
1		This analysis pooled estimates from these two studies to develop a C-R function linking PM to chronic bronchitis.	The analysis proves that there is no link between PM and bronchitis !!

For simplicity, we perform no data augmentation for any of the datasets. When training logistic regressions on the last-layer representations, we demean the data based on the mean of the training set. This is based on previous work from Chen et al. (2020), which states that demeaning is a necessary step before determining concept vectors.

I.2. Models & Training Procedure

Models: For the Waterbirds and CelebA dataset, we use the ResNet50 architecture implemented in the `torchvision` package: `torchvision.models.ResNet50(pretrained=True)`. More details on the model can be found in the original paper from He et al. (2016). We finetune the model using the parameters of Kirichenko et al. (2023). For waterbirds, this means using a learning rate of 10^{-3}, a weight decay of 10^{-3}, a batch size of 32, and for 100 epochs without early stopping. For CelebA, this means using a learning rate of 10^{-3}, a weight decay of 10^{-4}, a batch size of 128, and for 50 epochs without early stopping. We use stochastic gradient descent (SGD) with a momentum parameter of 0.9. After this, PCA is applied to the embeddings of the layer of the architecture, to reduce the dimensionality from 2048 to 300.

For the MultiNLI dataset, we use the base BERT model implemented in the `transformers` package (Wolf et al., 2019): `BertModel.from_pretrained("bert-base-uncased")`. The model was pre-trained on BookCorpus, a dataset consisting of 11,038 unpublished books, as well as the English Wikipedia (excluding lists, tables and headers). More details on the model can be found in the original paper: Devlin et al. (2019). For finetuning the BERT model on MultiNLI we use early stopping, and stop the procedure if we observe no improvement after 1 epoch. We use the Adam optimizer (Kingma & Ba, 2015) with the standard settings in *Pytorch*. When finetuning, we train for a maximum of 10 epochs, use a batch size of 32, a learning rate of 10^{-5}, and a weight decay of 10^{-4}.

Training procedure of concept-removal methods: For JSE, ERM INLP and RLACE, we use early stopping to prevent overfitting. If we observe no improvement on the validation set for a certain model after 5 epochs, the training procedure is stopped and the model with the lowest loss on the validation set is selected. We use stochastic gradient descent (SGD) with a momentum parameter of 0.9 for JSE, ERM INLP and RLACE, and train for a maximum of 50 epochs.

I.3. Implementation Details & Parameter Selection

We only assume access to a validation dataset that follows the same distribution as the training dataset. This means that the conditional probability $p(y_{\text{mt}} = y|y_{\text{sp}} = y)$ is the same across the training and validation set. For experiments where we change the $p(y_{\text{mt}} = y|y_{\text{sp}} = y)$ in the training set, we do not select new parameters for each case, but select them based on the scenario where $p(y_{\text{mt}} = y|y_{\text{sp}} = y) = 0.9$.

For JSE, ERM, INLP, GDRO, JTT, ERM-GW and SUBG we set the batch size at 128. If we work with these methods with the smaller datasets in Section G.2, we set the batch size to 64. We also do this for RLACE, except in the case of the MultiNLI dataset - we only observed convergence for this dataset and method with a batch size of 512. After setting the batch size, we select the best combination of the learning rate and weight decay. For the learning rate, we assess the values 10^{-1}, 10^{-2}, 10^{-3} and 10^{-4}. For the weight decay, we assess the values 0, 10^{-3}, 10^{-2}, 10^{-1} and 1. In the case of the Toy dataset we always set the weight decay to 0. For each method, we use the weighted binary cross-entropy on the validation set to measure performance. For the Toy dataset, the performance is measured across 10 runs, and for the Waterbirds, CelebA and multiNLI dataset the performance is measured across 5 runs, each time with a different finetuned model.

Below, we detail how the parameters were selected for each method, as well as implementation details for RLACE and adversarial removal. The selected combinations of the learning rate and weight decay can be found in Table 14.

ERM: We select the learning rate and weight-decay combination that has the best performance for the main-task labels. These parameters are also used when fitting a logistic regression on the transformed representations from JSE, INLP or RLACE. We keep the parameters the same for group-weighted ERM and subgroup sampling.

INLP: we select the combination of parameters that has the best performance for the spurious concept labels, based on the first spurious concept vector found by INLP. We continue projecting out the spurious concept vectors found by INLP until the accuracy of the spurious concept classifier is no better than a majority rule classifier. Whether or not the BCE is statistically significantly different from that of a majority rule classifier is tested via an t-test of the difference in the BCE's for both classifiers, where the critical value is determined based on $\alpha = 0.05$.

JSE: we select the combination of parameters that has the best performance for the spurious concept and main-task concept labels, weighing each equally. This performance is based on the first set of spurious and main-task concept vectors. The critical value of the tests in Section D is determined based on $\alpha = 0.05$.

LEACE: we use the original code from Belrose et al. (2023) for implementation of the method.

RLACE: we use the original code from Ravfogel et al. (2022a), and run the algorithm for a maximum of 50,000 iterations. For the spurious concept classifier and optimizing the projection matrix, we use the same parameters as INLP. We optimize the projection matrix until the accuracy of the classifier is lower than 51%. Similar to Ravfogel et al. (2022a), in each case we find a matrix of rank 1.

Adversarial removal: The weight of the adversary loss is set to $\lambda = 1$. The entire architecture is finetuned, and the adversary is trained using the gradient reversal method (Ganin & Lempitsky, 2015). For the vision datasets, we observe that the accuracy of the adversary converges to below 55%, which is commonly accepted as success for the method. After the adversarial removal method, we apply standard ERM including PCA, as for the other methods. For MultiNLI, we experimented extensively with hyper-parameters in order to get both a high accuracy on the main task, and the accuracy of the adversary to converge to below 55%. We did not observe this, even after lowering the weight of the adversary loss from

1 to 10^{-1} or even 10^{-2}. We used the Adam optimizer when performing adversarial removal.

GDRO: we fix the hyper-parameter $\eta_g = 0.1$, similar to Idrissi et al. (2022). We grid-search the best combination of the learning rate, weight decay and the hyper-parameter of C, with possible values of 0, 1, 2, 3, 4, 5. We provided initial parameters for the GDRO model based on an fitted ERM model.

JTT: We grid-search the best combination of the learning rate, weight decay and hyper-parameter of λ, which is the weight for the misclassified samples. For λ, we consider the values of 2, 5, 10, and 25.

Table 14. **Selected combinations of learning rate and weight decay.** For the C parameter of GDRO, we use the values $1, 2, 5$ and 5 for respectively the Waterbirds, CelebA, multiNLI and Toy datasets. For the λ parameter of JTT, we use the values of $2, 5, 10, 2$ for respectively the Toy, Waterbirds, CelebA, and multiNLI datasets.

Dataset	Method	Learning Rate	Weight Decay
Toy	JSE	10^{-2}	0
	ERM	10^{-1}	0
	INLP	10^{-1}	0
	RLACE	10^{-1}	0
	GDRO	10^{-3}	0
	JTT	10^{-3}	0
Waterbirds	JSE	10^{-3}	10^{-3}
	ERM	10^{-2}	10^{-2}
	INLP	10^{-2}	10^{-3}
	RLACE	10^{-2}	10^{-3}
	ADV	10^{-3}	10^{-4}
	GDRO	10^{-3}	10^{-1}
	JTT	10^{-2}	10^{-2}
CelebA	JSE	10^{-2}	10^{-3}
	ERM	10^{-2}	10^{-2}
	INLP	10^{-2}	10^{-3}
	RLACE	10^{-2}	10^{-3}
	ADV	10^{-3}	10^{-4}
	GDRO	10^{-3}	10^{-3}
	JTT	10^{-2}	10^{-2}
MultiNLI	JSE	10^{-2}	10^{-2}
	ERM	10^{-2}	1
	INLP	10^{-2}	10^{-3}
	RLACE	10^{-2}	10^{-2}
	GDRO	10^{-4}	1
	JTT	10^{-4}	10^{-3}