Probing Classifiers are Unreliable for Concept Removal and Detection

Abhinav Kumar [1] **Chenhao Tan** [2] **Amit Sharma** [1]

Abstract

Neural network models trained on text data have been found to encode undesired linguistic or sensitive attributes in their representation. Removing such attributes is non-trivial because of a complex relationship between the attribute, text input, and the learnt representation. Recent work has proposed post-hoc and adversarial methods to remove such unwanted attributes from a model's representation. Through an extensive theoretical and empirical analysis, we show that these methods can be counter-productive: they are unable to remove the attributes entirely, and in the worst case may end up destroying all task-relevant features. The reason is the methods' reliance on a probing classifier as a proxy for the attribute which we prove is difficult to train correctly in presence of spurious correlation.

1. Introduction

Neural network classifiers built using text data have been shown to learn spuriously correlated features (Gururangan et al., 2018a; McCoy et al., 2019) or embed sensitive attributes like gender or race (Bolukbasi et al., 2016; Conneau et al., 2018) at their representation layer. Classifiers that use such sensitive or spurious concepts (henceforth *concepts*) raise concerns of model unfairness and affects out-of-distribution generalization (Sagawa et al., 2020a; Arjovsky et al., 2020). Because the classifiers are based on hard-to-interpret deep neural networks, removing the influence of these concepts is non-trivial. Since many concepts cannot be modified at the input tokens level, removal methods that work at the representation layer have been proposed: post-hoc removal (Bolukbasi et al., 2016; Xu et al., 2017) on a pre-trained model (e.g., null space projection (Ravfogel et al., 2020)), and adversarial removal (Ganin & Lempitsky, 2015; Elazar & Goldberg, 2018) by jointly training the main task classifier with an (adversarial) classifier for the concept.

In this paper, we theoretically show that both classes of methods can be counter-productive in real-world settings where the main-task label is correlated with the concept. Examples include natural language inference (main-task) where the presence of negative words (spurious concept) may be correlated with the "contradicts" label. Our key result is based on the observation that both these methods internally use an auxiliary (probing) classifier (Shi et al., 2016; Adi et al., 2017) that aims to predict the spurious feature based on the representation learnt by the main classifier.

We show that the auxiliary classifier cannot be a reliable signal on whether the representation includes features that are causally derived from the concept. As previous work has argued (Tsipras et al., 2019; Sagawa et al., 2020c; Arjovsky et al., 2020; Belinkov, 2022), if the representation features causally derived from the concept are not predictive enough, the probing classifier for the concept can be expected to rely on correlated features to obtain a higher accuracy. However, we show a stronger result by considering a setting where there is no potential accuracy gain and the concept's features are easily learnable. Namely, even when the concept's causally-related features alone can provide 100% accuracy, the probing classifier may still learn non-zero weights for the main-task relevant features. Based on this result, under some simplifying assumptions, we prove that both post-hoc and adversarial training methods can fail to remove the undesired concept, remove useful task-relevant features in addition to the undesired concept, or do both. As an extreme case, we show how post-hoc methods can lead to a random main-task classifier, removing all task-relevant information from the representation [1]. Our contributions include:

- Theoretical analysis of null space and adversarial methods, showing they fail to remove an unwanted concept from a model's representation, even under favorable conditions.
- Empirical results on four datasets showing that the null-space and adversarial removal methods are unable to remove concept-related features fully or end up unnecessarily removing task-relevant features.
- A practical spuriousness score motivated by theory for evaluating the output of removal methods.

[1]Microsoft Research, India [2]Department of Computer Science, University of Chicago. Correspondence to: Abhinav Kumar <abhinavkumar.sieger@gmail.com>.

Published at the ICML 2022 Workshop on Spurious Correlations, Invariance, and Stability. Baltimore, Maryland, USA. Copyright 2022 by the author(s).

[1]Full paper version could be found at ArXiv with same title.

2. Concept removal can be counter-productive

For a classification task, let $(\boldsymbol{x}^i, y_m^i)_{i=1}^n$ be set of examples in the dataset $\mathcal{D}_m$, where $\boldsymbol{x}^i \in \boldsymbol{X}$ is the input feature and $y_m^i \in Y_m$ the label. We call this as the *main task* and label y_m^i as the main task label. The main task classifier can be written as $c_m(h(\boldsymbol{x}))$ where $h : \boldsymbol{X} \to \boldsymbol{Z}$ is an encoder mapping the input $\boldsymbol{X}$ to the latent representation $\boldsymbol{Z}$ and $c_m : \boldsymbol{Z} \to Y_m$ is the classifier on top of the representation $\boldsymbol{Z}$. Additionally, we are given labels for a spurious or sensitive concept, $y_p^i \in Y_p$, and our goal is to ensure that the representation $h(\boldsymbol{x})$ learnt by the main classifier does not include concept-causal features defined as:

Definition 2.1. A feature $Z_j \in \boldsymbol{Z}$ (j^{th} dimension of $h(\boldsymbol{x})$) at the representation layer is causally derived from a concept (*concept-causal* for short) if upon changing the value of the concept label, the corresponding change in the input's value $\boldsymbol{x}$ will lead to a change in the feature's (Z_j) value.

The above definition of *concept-causal* feature requires assumption of some underlying causal mechanism generating the text which is often not available. Hence these latent space based removal methods (Sec 2.1) use an auxiliary probing classifier to estimate the concept-causal feature present in the latent representation from the concept label.

2.1. Challenge with learning a *clean* probing classifier

Given $\boldsymbol{z} = h(\boldsymbol{x})$ and the concept label y_p, the goal of probing task is to learn a classifier $c_p(\boldsymbol{z})$ such that it only uses the concept-causal features and the accuracy for y_p is maximized. If there are features in $\boldsymbol{z}$ outside concept-causal that improve the accuracy for the classifier, a classifier trained on standard losses such as cross-entropy or max-margin is expected to use those features too, as argued in the probing literature (Hewitt & Liang, 2019; Belinkov, 2022). In this section, however, we create a setup that is the most favorable for the classifier to use only concept-causal features—no accuracy gain on using features outside concept-causal, disentangled representation so that no further representation learning is needed, and concept labels that are linearly separable using concept-causal features—yet we find that a trained classifier uses non-concept-causal features.

Assumption 2.1 (Disentangled Frozen Latent Representation)**.** The latent representation $\boldsymbol{z}$ is disentangled and is of form $[\boldsymbol{z}_m, \boldsymbol{z}_p]$, where $\boldsymbol{z}_p \in \mathbb{R}^{d_p}$ are the features causally derived from the concept label and $\boldsymbol{z}_m \in \mathbb{R}^{d_m}$ are the features causally derived from the main task label along with rest of the features in latent representation. Here d_m and d_p are the dimensions of $\boldsymbol{z}_m$ and $\boldsymbol{z}_p$ respectively. Also the encoder $h(\boldsymbol{x})$ which maps the input $\boldsymbol{X}$ to latent representation $\boldsymbol{Z}$ is frozen or non-trainable.

While it may be less surprising that a spurious feature is used if the invariant feature (concept-causal) does not give 100% accuracy, we show that spurious feature will be used even if the invariant feature is fully predictive of the label, following (Nagarajan et al., 2021). We formalize it as,

Assumption 2.2 (concept-causal feature Linear Separability)**.** The concept-features of the latent representation ($\boldsymbol{z}_p$) for every point are linearly separable/fully predictive for the concept labels y_p, i.e $y_p^i \cdot (\hat{\boldsymbol{\epsilon}}_p \cdot \boldsymbol{z}_p^i + b_p) > 0$ for all datapoints $(\boldsymbol{x}^i, y_p^i)$ for some $\hat{\boldsymbol{\epsilon}}_p \in \mathbb{R}^{d_p}$. For the case of zero-centered latent space we have $b_p = 0$.

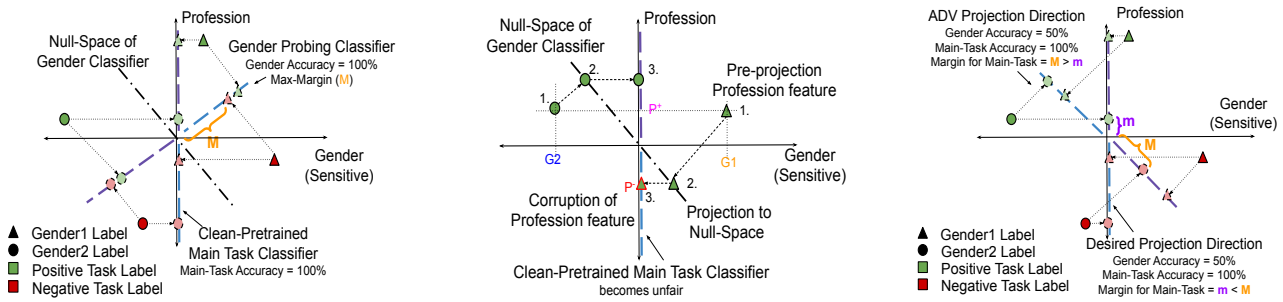
In addition, we assume that there is some correlation between the concept and main-task label: a function of $\boldsymbol{z}_m$ can also classify correctly a non-empty subset of points.

Assumption 2.3 (Spurious Correlation)**.** For a subset of training points $\mathcal{S} \subset \mathcal{D}_p$, $\boldsymbol{z}_m$ is linearly-separable with respect to concept label y_p i.e $y_p^i \cdot (\hat{\boldsymbol{\epsilon}}_m \cdot \boldsymbol{z}_m^i + b_m) > 0$ for some $\hat{\boldsymbol{\epsilon}}_m \in \mathbb{R}^{d_m}$ and $b_m \in \mathbb{R}$. For the case of zero-centered latent space we have $b_m = 0$.

Now we state the key lemma: if non-concept-causal features $\boldsymbol{z}_m$ are linearly separable w.r.t. to concept label y_p for only a *few special* points, then the probing classifier $c_p(\boldsymbol{z})$ will use those features. As in (Nagarajan et al., 2021), for ease of exposition we use max-margin training loss.

Lemma 2.1. Let the latent representation be frozen and disentangled such that $\boldsymbol{z} = [\boldsymbol{z}_m, \boldsymbol{z}_p]$ (Assm 2.1), where concept-features $\boldsymbol{z}_p$ be fully predictive (Assm 2.2). Let $c_p^*(\boldsymbol{z}) = \boldsymbol{w}_p \cdot \boldsymbol{z}_p$ be the desired clean/purely-invariant linear classifier trained using max-margin objective (Suppl C.1) which only uses $\boldsymbol{z}_p$ for its prediction. Let $\boldsymbol{z}_m$ be the spurious feature s.t. for the margin points of $c_p^*(\boldsymbol{z})$, $\boldsymbol{z}_m$ be linearly-separable w.r.t. probing task label y_p (Assm 2.3). Then, assuming the latent space is centered around $\mathbf{0}$, the concept-probing classifier trained using max-margin objective will be of form $c_p(\boldsymbol{z}) = \boldsymbol{w}_p \cdot \boldsymbol{z}_p + \boldsymbol{w}_m \cdot \boldsymbol{z}_m$ where $\boldsymbol{w}_m \neq \mathbf{0}$.

For detailed proof see Suppl C.2. In Suppl C.3 we prove that the assumptions for Lemma 2.1 are both sufficient and necessary for a classifier to use non-concept-causal features $\boldsymbol{z}_m$ when $\boldsymbol{z}_p$ is 1-dimensional. Lemma 2.1 generalizes a result from (Nagarajan et al., 2021) by using fewer assumptions (we do not assume that z_m to be binary, that z_m and z_p are conditionally independent given y_m or y_p, and do not assume monotonicity of classifier norm with dataset size). We also present similar result for main-task classifier in Suppl C.4. Sagawa et al. (2020b) shows that subsampling the dataset help in learning *clean* classifier by breaking spurious correlation. Note that even after subsampling as in Sagawa et al. (2020b) Assm 2.3 could still be satisfied if it is satisfied in the original dataset, since margin points are few and may not be subsampled.



(a) Clean Main-task Classifier (b) Main Classifier becomes unfair (c) Failure of Adversarial Removal

Figure 1: **Failure Mode of Removal Methods**: Assume a 2-dimensional representation, $\boldsymbol{z}$. Main Task feature is Profession and sensitive feature is Gender to be removed. **(a)** A "clean" main-task classifier—vertical line which only uses profession—that is input to null-space (INLP) for removal. From Lemma 2.1, INLP trains a gender classifier which is shown by slanted line. **(b)** For the two points having same profession but different gender feature (marked '1'), post projection to the null-space ('2') has their profession feature reversed ('3'), thus making the fair pretrained classifier become unfair. **(c)** For the adversarial removal method, there can be two projection directions that yield a *random-guess* 50% accuracy on gender and the same main-task accuracy. However, the slanted projection direction has bigger margin for main-task and will be preferred, again leading to a final classifier that uses the Gender feature.

2.2. Failure mode of Null-space removal (INLP)

The null space method (Ravfogel et al., 2020; Elazar et al., 2021), henceforth referred as *INLP*, first estimates the subspace in the latent space discriminative of the concept we want to remove by training a probing classifier $c_p(\boldsymbol{z}^i) \rightarrow y_p^i$, where y_p^i is the concept label. Then the projection is done on the null-space of this probing classifier which is expected to be non-discriminative of the concept. (Ravfogel et al., 2020) used a linear probing classifier ($c_p(\boldsymbol{z})$) to ensure that the any linear classifier cannot recover the removed concept from modified latent representation $\boldsymbol{z}'$ and hence the main task classifier ($c_m(\boldsymbol{z}')$) also become invariant to removed attribute. Details for the method are in Suppl D.1. Below we state the failure of null-space removal under the assumptions of Lemma 2.1 (proof in Suppl D.2). We use $\boldsymbol{z}^{i(k)}$ to denote the representation $\boldsymbol{z}^i$ after k steps of INLP.

Theorem 2.2. Let the latent representation $\boldsymbol{z}$ be disentangled (Assm 2.1), $\boldsymbol{c}_m(\boldsymbol{z})$ be the pre-trained main-task classifier, and $\boldsymbol{c}_p(\boldsymbol{z})$ the probing classifier used by INLP to remove the unwanted feature $\boldsymbol{z}_p$ from the latent representation. Let the assumptions of Lemma 2.1 be true for the probing classifier $c_p(\boldsymbol{z})$, trained using max-margin objective.

1. **Mixing:** After the first projection step of linear-INLP, the dimensions of $\boldsymbol{z}$ gets mixed such that $\boldsymbol{z}^{i(1)} = [g(\boldsymbol{z}_m^i, \boldsymbol{x}_p^i), f(\boldsymbol{z}_p^i, \boldsymbol{z}_m^i)] \neq [\boldsymbol{z}_m^i, \boldsymbol{z}_p^i]$. Also, this mixing is non-invertible with subsequent projection steps. Thus, the latent space is no longer disentangled and removal of concept-causal features will also lead to removal of task-specific features.

2. **Removal:** The L2-norm of the latent representation $\boldsymbol{z}$ decreases with every projection step as long as the parameters of probing classifier at a step ($\boldsymbol{w}^k$) does not lie completely in the space spanned by parameters of previous probing classifiers i.e. span($\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{k-1}$). Thus, after sufficiently many steps, INLP can destroy all information in the representation, $\boldsymbol{z}^{i(\infty)} = [\boldsymbol{0}, \boldsymbol{0}]$.

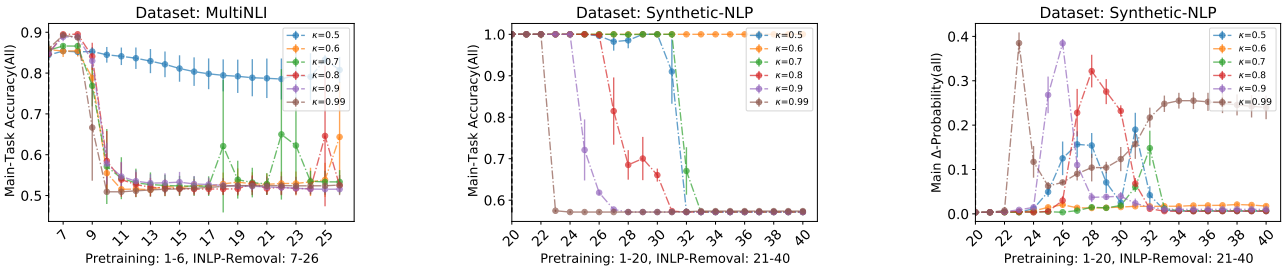
2.3. Failure mode of adversarial removal methods

To remove the unwanted features $\boldsymbol{z}_p$ from the latent representation, the adversarial removal method jointly trains the main classifier $c_m(\boldsymbol{z})$ and the probing classifier $c_p : \boldsymbol{z} \rightarrow y_p$ by specifying c_p's loss as an adversarial loss (Elazar & Goldberg, 2018) . For details refer to Suppl E.1.

Since the encoder $h : \boldsymbol{x} \rightarrow \boldsymbol{z}$ mapping the input the latent representation $\boldsymbol{z}$ is frozen (Assm 2.1), to allow training for the main-task classifier we introduce additional representation layers after it. For simplicity in the proof, we assume a linear transformation to the latent representation $h_2 : \boldsymbol{z} \rightarrow \boldsymbol{\zeta}$. After this, we have a linear main-task classifier $c_m : \boldsymbol{\zeta} \rightarrow y_m$, as before. The probing classifier $c_p : \boldsymbol{\zeta} \rightarrow y_p$ is trained adversarially to remove $\boldsymbol{z}_p$ from latent representation $\boldsymbol{\zeta}$. Thus, the goal of the adversarial method can be stated as removing the information of $\boldsymbol{z}_p$ from $\boldsymbol{\zeta}$. Let the main-task classifier satisfy the assumption of the generalized version of Lemma 2.1 (Lemma C.5 in Suppl C.4). In addition we need an assumption on the hard-to-classify margin points which says that main-task labels and adversarial concept label are correlated on margin point of *clean* main-task classifier.

Assumption 2.4 (Label Correlation of Margin Point)**.** For the margin points of *clean* classifier for the main task, the adversarial-probing label y_p and the main task labels y_m are correlated, i.e w.l.o.g. $y_m^i = y_p^i$ for all margins points of clean main-task classifier.

Theorem 2.3. Let the latent representation $\boldsymbol{Z}$ satisfy Assm 2.1, $h_2(\boldsymbol{z})$ be a linear transformation s.t. $h_2 : \boldsymbol{Z} \rightarrow \boldsymbol{\zeta}$,



(a) Main-Task Acc(All) (b) Main-Task Acc(All) (c) Main-Task Δ Prob(All)

Figure 2: **Null-Space and Adversarial Removal Failure**: For discussion see Sec 3.

main classifier be $c_m(\boldsymbol{\zeta}) = \boldsymbol{w}_{c_m} \cdot \boldsymbol{\zeta}$ and the adversarial classifier be $c_p(\boldsymbol{\zeta}) = \boldsymbol{w}_{c_p} \cdot \boldsymbol{\zeta}$. Let all the assumptions of generalized Lemma 2.1 (Lemma C.5 in Suppl C.4) be satisfied for main-classifier $c_m(\cdot)$ when using $\boldsymbol{Z}$ directly as input and Assm 2.2 be satisfied on $\boldsymbol{Z}$ w.r.t. adversarial-task. Let $h_2^*(\boldsymbol{z})$ be the desired encoder which is successful in removing $\boldsymbol{z}_p$ from $\boldsymbol{\zeta}$. Then there exists an undesired/incorrect encoder $h_2(\boldsymbol{z})$ such that $\boldsymbol{\zeta} = h_2(\boldsymbol{z}) = f(\boldsymbol{z}_m, \boldsymbol{z}_p)$ for some function "f" and the main classifier using it $c_m(h_2(\boldsymbol{z}))$ has bigger margin than $c_m(h_2^*(\boldsymbol{z}))$ and has,

1. $Acc(c_p(h_2(\boldsymbol{z})), y_p) = Acc(c_p(h_2^*(\boldsymbol{z})), y_p)$ when adversarial probing classifier $c_p(\cdot)$ is trained using any learning objective. Thus the undesired encoder $h_2(\boldsymbol{z})$ is indistinguishable from desired encoder $h_2^*(\boldsymbol{x})$ in terms of adversarial task prediction accuracy but better for main-prediction task in terms of max-margin objective.

2. $L(c_p(h_2(\boldsymbol{z})), y_p) > L(c_p(h_2^*(\boldsymbol{z})), y_p)$ under Assm 2.2 and Assm 2.4 where L is either max-margin or cross-entropy loss used for training $c_p(\cdot)$. Thus undesired encoder $h_2(\boldsymbol{z})$ is preferable over desired encoder $h_2^*(\boldsymbol{z})$ for both main and adversarial task objective.

Spuriousness Score. We next define a new metric which will help us access the quality of a given classifier. To define spuriousness score over binary labels and concepts, we consider two subgroups, S_{min} and S_{maj} where S_{maj} contains the dominant correlation exhibited in dataset while S_{min} contains the rest of the data.

Definition 2.2. Given a dataset, $\mathcal{D}_{m,p} = S_{min} \cup S_{maj}$ containing binary task label and binary concept, let $Acc^{c_m}(S_{min})$ be the accuracy of the given main-task classifier (c_m) on the minority group S_{min} and $Acc^*(S_{min})$ be the accuracy on the minority group of a *clean* classifier that does not use the unwanted spurious concept. We define spuriousness score of "c_m" as: $\psi(c_m) = \left|1 - \frac{Acc^{c_m}(S_{min})}{Acc^*(S_{min})}\right|$

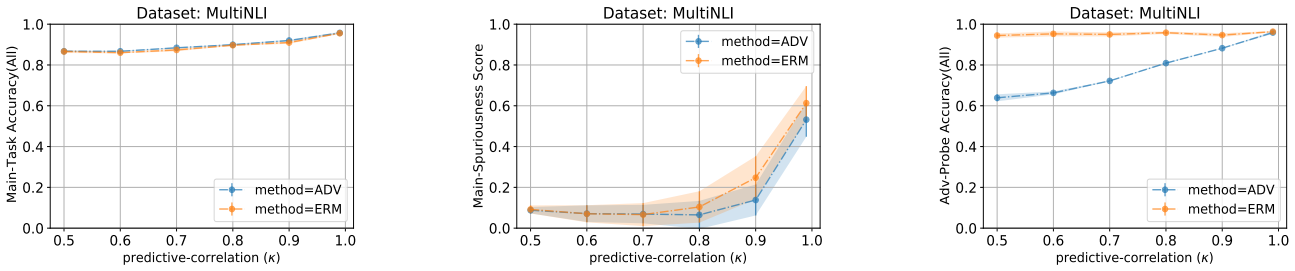
To estimate $Acc^*(S_{min})$, we subsample the dataset such that y_p takes a single value in the sample and train the main classifier on it, as done by (Ravichander et al., 2021). Thus, probing label y_p no longer is correlated with the main-task label y_m. There can be other ways to estimate it, e.g., by reweighting the data or using the accuracy on S_{maj}. However, we found that the former had high variance and the latter requires an equal-noise assumption such that $Acc(S_{maj}) = Acc(S_{min})$ on a *clean* main-task classifier.

3. Experimental Results

Dataset. We demonstrate the failure of null-space (INLP) and adversarial removal (AR) on three real-world dataset – MultiNLI (Williams et al., 2018), Twitter-PAN16 (Rangel et al., 2016) and Twitter-AAE (Blodgett et al., 2016). For brevity, only show the results for MultiNLI dataset in this section (see Suppl G for complete results). In MultiNLI dataset, given two sentences–premise and hypothesis–the main task is to predict whether hypothesis *entails*, *contradicts* or is *neutral* to premise. We simplify to a binary task of predicting whether a hypothesis *contradicts* the premise or not. Since negation words like *nobody,no,never* and *nothing* have been reported to be spuriously correlated with with the *contradiction* label (Gururangan et al., 2018b), we create a 'negation' concept denoting the presence of these words. The goal is to remove the negation concept from the NLI model's representation. In general the *negation* feature might be useful for the task and should not be removed completely. But for simplicity here we consider complete removal of this feature.

Synthetic-Text. To understand the reasons for failure, we introduce a Synthetic-Text dataset where it is possible to change the input text based on a change in concept (thus implementing Def. 2.1). Here we can directly evaluate whether the concept is being used by the main-task classifier by intervening on the concept (adding or removing) and observing the change in prediction scores. The main-task is to predict whether a sentence contains a numbered word (e.g., *one, fifteen, etc.*). We introduce spurious concept (length) by increasing the length of sentences that contain *numbered* words.

Predictive correlation. To assess robustness of removal methods, we create multiple datasets with different *predictive* correlation between two labels y_m and y_p. The predictive correlation (κ) measures how informative one



(a) Main-Task Acc(All) (b) Main Spuriousness Score (c) Adv-Probe Acc(All)

Figure 3: **Adversarial Removal Failure**: For discussion see Adversarial Removal result in Sec 3.

variable is for predicting the other, $\kappa = Pr(y_m \cdot y_p > 0) = \frac{\sum_{i=1}^{N} \mathbf{1}[y_m \cdot y_p > 0]}{N}$, where N is the size of dataset and $\mathbf{1}[\cdot]$ is the indicator function which is 1 if the argument is true otherwise 0. Predictive-correlation lies in $\kappa \in [0.5, 1]$ where $\kappa = 0.5$ means no-correlation and $\kappa = 1$ means the attributes are completely correlated.

Results: Null Space Removal In general, for any model given as input to INLP, it may be difficult to verify whether the removal method removed the correct features. Hence, we construct a benchmark where the input classifier is *clean*, i.e., it does not use the concept at all. We do so by training on a subset of data with the same value of the property, as in (Ravichander et al., 2021).

In Fig 2a, iteration 7-26 shows individual steps of null-space removal for MultiNLI dataset. Since, the given pre-trained classifier was *clean* i.e not using the sensitive features, null-space removal shouldn't have any effect on it. Colored lines show different datasets used by INLP with different level of predictive correlation κ. We observe that for most value of κ, the main-task classifier's accuracy eventually goes to 50% random guess accuracy implying that the main-task related attribute has been removed by the removal process, as predicted by Theorem 2.2. Higher the value of correlation κ, faster the removal of main-task attribute happens.

To avoid the full collapse, a stopping criteria of INLP is to stop when the main-task classifier's performance drops (Ravfogel et al., 2020). In Fig 2c we measure the sensitivity of Synthetic-Text main-task classifier w.r.t. to sensitive concept by changing the feature in input sentences corresponding to sensitive concept and measuring the change in main-task classifier prediction probability (ΔProb). At lower iterations of INLP, the change in main-task output due to change in sensitive concept's value, ΔProb, is higher than that of the input classifier. For example, for $\kappa = 0.8$, the main-task classifier's performance drops for the first time at iteration 27, but it has a high $\Delta\text{Prob} \approx 25\%$ as shown in Fig 2c. Hence it is possible that stopping prematurely will lead us to a classifier which is more unfair or reliant to sensitive concept than it was before, consistent with the first statement in Theorem 2.2 stating that INLP will lead to mixing of features in latent space.

Results: Adversarial Removal We compare the standard ERM training of the main classifier with adversarial-removal (AR) method over the same number of epochs (20). For MultiNLI dataset, Fig 3b shows the spuriousness score (Def 2.2) of ERM and AR classifiers as we vary the predictive correlation (κ) between the main-task label and sensitive concept label in the training dataset. While the spuriousness score for classifier trained using AR (blue curve) is lower than that of ERM for all values of κ, it is substantially away from zero. Thus, the AR method fails to completely remove the sensitive/spurious concept completely from the latent representation. Fig 3c shows the possible reason using the sensitive concept probing classifier accuracy for ERM and AR. The probe accuracy after adversarial training doesn't decrease to 50% but stops at accuracy proportional to the predictive correlation κ. This is expected since even if the AR would have been successful in removing the sensitive feature, the main-task feature would still be predictive of sensitive concept label by κ due to the spurious correlation between them. However, the converse is not true: an accuracy of κ does not imply that sensitive concept is fully removed. The results substantiate the first statement of Theorem 2.3: given two representations where one (*desired*) does not have concept features while the other (*undesired*) contains the sensitive concept features, the undesired one may be better for the main task loss even as both may have the same probing accuracy. While previous empirical work ((Elazar & Goldberg, 2018)) showed a similar limitation for adversarial methods, we extend it through theory and empirical experiments that measure the use of unwanted concept by classifier through the spuriousness score, not just its presence in the representation.

4. Conclusion and Discussion

Using both theoretical analysis and empirical results over real-world datasets, we presented a critique of recent work that claims to remove spurious or sensitive concepts from a text classifier. Given the practical relevance of the removal problem, we also provided a spuriousness metric to evaluate any classifier on its use of unwanted concepts.

References

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=BJh6Ztuxl.

Agarap, A. F. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization, 2020.

Belinkov, Y. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, 04 2022. ISSN 0891-2017. doi:10.1162/coli_a_00422. URL https://doi.org/10.1162/coli_a_00422.

Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

Blodgett, S. L., Green, L., and O'Connor, B. Demographic Dialectal Variation in Social Media: A Case Study of African-American English. In *Proceedings of EMNLP*, 2016.

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., and Kalai, A. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4356–4364, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi:10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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.

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. *Trans. Assoc. Comput. Linguistics*, 9:160–175, 2021. URL https://transacl.org/ojs/index.php/tacl/article/view/2423.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 1180–1189. JMLR.org, 2015.

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 2018a. Association for Computational Linguistics. doi:10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

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 2018b. Association for Computational Linguistics. doi:10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

Hewitt, J. and Liang, P. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

McCoy, T., Pavlick, E., and Linzen, T. Right for the wrong reasons: Diagnosing syntactic heuristics in natural lan-

guage 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.

Nagarajan, V., Andreassen, A. J., and Neyshabur, B. Understanding the failure modes of out-of-distribution generalization. *ArXiv*, abs/2010.15775, 2021.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Rangel, F., Rosso, P., Verhoeven, B., Daelemans, W., Potthast, M., and Stein, B. Pan16 author profiling, September 2016. URL https://doi.org/10.5281/zenodo.3745963.

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.

Ravichander, A., Belinkov, Y., and Hovy, E. Probing the probing paradigm: Does probing accuracy entail task relevance? In *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.

Rehurek, R. and Sojka, P. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

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 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. 8346–8356. PMLR, 13–18 Jul 2020b. URL https://proceedings.mlr.press/v119/sagawa20a.html.

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, 2020c.

Shi, X., Padhi, I., and Knight, K. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1526–1534, Austin, Texas, November 2016. Association for Computational Linguistics. doi:10.18653/v1/D16-1159. URL https://www.aclweb.org/anthology/D16-1159.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

Weisstein, E. W. "symmetric matrix." from mathworld–a wolfram web resource, https://mathworld.wolfram.com/symmetricmatrix.html. URL https://mathworld.wolfram.com/SymmetricMatrix.html.

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., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface's transformers: State-of-the-art natural language processing, 2019. URL https://arxiv.org/abs/1910.03771.

Xie, Q., Dai, Z., Du, Y., Hovy, E., and Neubig, G. Controllable invariance through adversarial feature learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 585–596, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

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.

A. Broader Impact and Ethical Consideration

Removal of spurious or sensitive concepts is an important problem to ensure that machine learning classifiers generalize better to new data and are fair towards all groups. We found multiple limitations with current removal methods and recommend caution against the use of these methods in practice.

B. Limitations of our work

One of the limitations of our theoretical work is assuming frozen or non-trainable latent representation which makes the analysis of task-classifier trained on top of them relatively easier. We address this limitation in our empirical work where we do not make such assumptions. Also, our work addresses failure modes of probing based removal on two popular methods, post-hoc removal (Null-Space Removal) and Adversarial Removal and we conjecture that similar failure could be expected for other methods in those classes. It would be interesting to show that any concept removal method based on a probing classifier will fail in general.

C. Probing and Main Classifier Failure Proofs

C.1. Notation and Setup: Max-margin Classifier

Assm 2.1 says encoder $h : \boldsymbol{X} \rightarrow \boldsymbol{Z}$, mapping the input to latent representation is frozen/non-trainable. Thus for every input $\boldsymbol{x}^i$ in the dataset $\mathcal{D}$, we have a corresponding latent representation $\boldsymbol{z}^i$ which is fixed. Also, the latent representation $\boldsymbol{Z}$ is disentangled i.e $\boldsymbol{z} = [\boldsymbol{z}_m, \boldsymbol{z}_p]$ where $\boldsymbol{z}_m$ are the feature causally derived from main task label and $\boldsymbol{z}_p$ are the features causally derived from the concept label. Let $c_p(\boldsymbol{z}) = \boldsymbol{w}_p \cdot \boldsymbol{z}_p + \boldsymbol{w}_m \cdot \boldsymbol{z}_m$ be the linear probing classifier which we train using max-margin objective. The hyperplane $c_p(\boldsymbol{z}) = 0$ is the decision boundary of this linear classifier. The points which falls on one side of the decision boundary ($c_p(\boldsymbol{z}) > 0$) are assigned one label (say positive label 1) and the rest are assigned another label (say negative label -1). The margin $\mathcal{M}_{c_p}$ of this hypothesis function ($c_p(\boldsymbol{z})$) is the distance of the closest latent representation from the decision boundary. The points which are closest to the decision boundary is called the margin points. The distance of a given latent representation $\boldsymbol{z}^i$ having class label y^i, where $y^i \in \{-1, 1\}$, from decision boundary is given by

$$\mathcal{M}_{c_p}(\boldsymbol{z}^i) := \frac{m_{c_p}(\boldsymbol{z}^i)}{\|\boldsymbol{w}\|} = \frac{y^i \cdot c_p(\boldsymbol{z}^i)}{\|\boldsymbol{w}\|} = \frac{y^i \cdot (\boldsymbol{w} \cdot \boldsymbol{z}^i + b)}{\|\boldsymbol{w}\|} \tag{1}$$

Max-Margin (MM): Then the max-margin classifier is trained by optimizing the following objective:

$$\underset{\boldsymbol{w},b}{\operatorname{argmax}} \left\{ \min_i \mathcal{M}_{c_p}(\boldsymbol{z}^i) \right\} \tag{2}$$

For ease of exposition we could convert this objective into multiple equivalent forms. To do this we observe that scaling the parameters of $c_p(\boldsymbol{z})$ by a factor of γ i.e $\boldsymbol{w} \rightarrow \gamma \boldsymbol{w}$ and $b \rightarrow \gamma b$ does not change the distance of the point from the decision boundary ($\mathcal{M}_{c_p}(\boldsymbol{z}^i)$).

MM-Denominator Version: We can use this freedom of scaling the parameters to set $m_{c_p}(\boldsymbol{z}^i) = 1$ for the closest point of any given hypothesis function, thus all the data points will satisfy the constraint:

$$m_{c_p}(\boldsymbol{z}^i) = y^i \cdot c_p(\boldsymbol{z}^i) \geq 1 \tag{3}$$

and giving us the final max-margin objective:

$$\underset{\boldsymbol{w}}{\operatorname{argmax}} \left\{ \frac{1}{\|\boldsymbol{w}\|} \right\} \tag{4}$$

under the constraint $m_{c_p}(\boldsymbol{z}^i) \geq 1$ corresponding to all the points in the dataset.

MM-Numerator Version: Alternatively, one can choose γ such that $\|\boldsymbol{w}\| = c$ where $c \in \mathbb{R}$ is some constant value. The the modified objective becomes:

$$\underset{\boldsymbol{w},b}{\operatorname{argmax}} \left\{ m_{c_p}(\boldsymbol{z}^i) \right\} \tag{5}$$

under constrain $\|\boldsymbol{w}\| = c$ which is usually set to 1.

We will use different formulation in our proofs later based on the ease of exposition and give a clear indication when we do so. One can refer to Chapter 7, Section 7.1 of (Bishop, 2006) for furthur details about max-margin classifier.

C.2. Proof of Sufficient Condition: Lemma 2.1 and Lemma C.5

Lemma 2.1. Let the latent representation be frozen and disentangled such that $\boldsymbol{z} = [\boldsymbol{z}_m, \boldsymbol{z}_p]$ (Assm 2.1), where concept-features $\boldsymbol{z}_p$ be fully predictive (Assm 2.2). Let $c_p^*(\boldsymbol{z}) = \boldsymbol{w}_p \cdot \boldsymbol{z}_p$ be the desired clean/purely-invariant linear classifier trained using max-margin objective (Suppl C.1) which only uses $\boldsymbol{z}_p$ for its prediction. Let $\boldsymbol{z}_m$ be the spurious feature s.t. for the margin points of $c_p^*(\boldsymbol{z})$, $\boldsymbol{z}_m$ be linearly-separable w.r.t. probing task label y_p (Assm 2.3). Then, assuming the latent space is centered around $\mathbf{0}$, the concept-probing classifier trained using max-margin objective will be of form $c_p(\boldsymbol{z}) = \boldsymbol{w}_p \cdot \boldsymbol{z}_p + \boldsymbol{w}_m \cdot \boldsymbol{z}_m$ where $\boldsymbol{w}_m \neq \mathbf{0}$.

In this section we prove that, given the assumption in Lemma 2.1 is satisfied, they are sufficient for a concept probing classifier $c_p(\boldsymbol{z})$ to use the spuriously correlated main-task feature $\boldsymbol{z}_m$. See Sec C.1 for detailed setup and max-margin training objective. Also, we could use the same line of reasoning to prove similar result for the main-task classifier i.e. when condition in Lemma C.5 is satisfied, the main-task classifier will use the spurious concept-feature $\boldsymbol{z}_p$. To keep the proof general for both the lemmas, we prove the result for a general classifier $c(\boldsymbol{z})$ trained to predict a task label y. Here the latent representation be of form $\boldsymbol{z} = [\boldsymbol{z}_{inv}, \boldsymbol{z}_{sp}]$ where $\boldsymbol{z}_{inv}$ is the feature which is causally-derived from/invariant to task-label y and $\boldsymbol{z}_{sp}$ be the feature spurious to the task label y. With respect to concept-probing classifier $c_p(\boldsymbol{z})$ in Lemma 2.1 $\boldsymbol{z}_{inv} \coloneqq \boldsymbol{z}_p$ and $\boldsymbol{z}_{sp} \coloneqq \boldsymbol{z}_m$. Similarly, for the main-task classifier in Lemma C.5, $\boldsymbol{z}_{inv} \coloneqq \boldsymbol{z}_m$ and $\boldsymbol{z}_{sp} \coloneqq \boldsymbol{z}_p$. For ease of exposition, we define two categories of classifiers based on which features they use:

Definition C.1 (Purely-Invariant Classifier)**.** Given a linear classifier of form $c(\boldsymbol{z}) = \boldsymbol{w}_{inv}\boldsymbol{z}_{inv} + \boldsymbol{w}_{sp}\boldsymbol{z}_{sp} + b$ is called *"purely-invariant"* if it does not use the spurious features $\boldsymbol{z}_{sp}$ i.e., $\boldsymbol{w}_{sp} = \mathbf{0}$.

Definition C.2 (Spurious-Using Classifier)**.** Given a linear classifier of form $c(\boldsymbol{z}) = \boldsymbol{w}_{inv}\boldsymbol{z}_{inv} + \boldsymbol{w}_{sp}\boldsymbol{z}_{sp} + b$ is called *"spurious-using"* if it uses the spurious features $\boldsymbol{z}_{sp}$ i.e., both $\boldsymbol{w}_{sp} \neq \mathbf{0}$ and $\boldsymbol{w}_{inv} \neq \mathbf{0}$.

Proof of Lemma 2.1 and C.5. Let $c_{inv}(\boldsymbol{z}) = \boldsymbol{w}_{inv} \cdot \boldsymbol{z}_{inv}$ be the *clean*/purely-invariant classifier trained using the max-margin objective using the *MM-Denominator* formulation given in Eq 4 such that $\boldsymbol{w}_{inv} \neq \mathbf{0}$. The classifier $c_{inv}(\boldsymbol{z})$ is 100% predictive of the task labels y (from Assm 2.2 for the probing task or Assm C.1 for the main-task). Here the bias term $b = 0$ since we assume the latent representation $\boldsymbol{z}$ is centered. The norm of this classifier is $\|\boldsymbol{w}_{inv}\|$ and the distance of each input latent representation ($\boldsymbol{z}^i$) with class label y^i ($y^i \in \{-1, 1\}$) from the decision boundary ($c_{inv}(\boldsymbol{z}) = 0$) is given by:

$$\mathcal{M}_{inv}(\boldsymbol{z}^i) = \frac{m_{inv}(\boldsymbol{z}^i)}{\|\boldsymbol{w}_{inv}\|} = \frac{y^i \cdot c_{inv}(\boldsymbol{z}^i)}{\|\boldsymbol{w}_{inv}\|} = \frac{y^i \cdot (\boldsymbol{w}_{inv} \cdot \boldsymbol{z}^i_{inv})}{\|\boldsymbol{w}_{inv}\|} \tag{6}$$

Since we have used the *MM-Denominator* version of max-margin to train $C_{inv}(\boldsymbol{z})$ from Eq 3 we have $m_{inv}(\boldsymbol{z}^i) = 1$ for the margin-points and greater than 1 for rest of the points. Next we will construct a new classifier parameterized by $\alpha \in [0, 1]$ by perturbing the clean/purely-invariant classifier $c_{inv}(\boldsymbol{z})$ such that:

$$c_\alpha(\boldsymbol{z}) = \alpha\big(\boldsymbol{w}_{inv} \cdot \boldsymbol{z}_{inv}\big) + \|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2}\big(\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}\big) \tag{7}$$

where $\hat{\boldsymbol{\epsilon}}_{sp} \in \mathbb{R}^{d_{sp}}$ is a unit vector in spurious subspace of features, d_{sp} is the dimension of the spurious subspace. We observe that the norm of this perturbed classifier $c_\alpha(\boldsymbol{x})$ is $\|\boldsymbol{w}_{inv}\|$, which is same as the clean/purely-invariant classifier $c_{inv}(\boldsymbol{x})$. Thus the distance of any input $\boldsymbol{z}^i$ with class label y^i from the decision boundary of this perturbed classifier, $c_\alpha(\boldsymbol{z}) = 0$ is given by:

$$\mathcal{M}_\alpha(\boldsymbol{z}^i) = \frac{m_\alpha(\boldsymbol{x}_i)}{\|\boldsymbol{w}_{inv}\|} = \frac{y^i \cdot c_\alpha(\boldsymbol{z}^i)}{\|\boldsymbol{w}_{inv}\|} \tag{8}$$

The perturbed classifier will be *spurious-using* i.e use the spurious feature $\boldsymbol{z}_{sp}$ when $\alpha \in [0, 1)$ since $(\|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2}) \neq 0$ for these setting of α. Thus to show that there exist a *spurious-using* classifier which has a margin greater than the margin of the *purely-invariant* classifier, we need to prove that there exist an $\alpha \in [0, 1)$ such that $c_\alpha(\boldsymbol{z})$ has bigger margin than $c_{inv}(\boldsymbol{z})$ i.e. $\min_z \mathcal{M}_\alpha(\boldsymbol{z}) > \min_z \mathcal{M}_{inv}(\boldsymbol{z})$. Since norm of parameters of both the classifier is same, substituting the expression of

$\mathcal{M}_\alpha$ and $\mathcal{M}_{inv}$ from Eq 6 and 8, we need to show $m_\alpha(\boldsymbol{z}^i) > 1$ for all $\boldsymbol{z}_i$. We have:

$$m_\alpha(\boldsymbol{z}^i) = y^i \cdot \left(\alpha(\boldsymbol{w}_{inv} \cdot \boldsymbol{z}^i_{inv}) + \|\boldsymbol{w}_{inv}\| \sqrt{1-\alpha^2} (\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}^i_{sp}) \right) \tag{9}$$

$$= \alpha \cdot m_{inv}(\boldsymbol{z}^i) + y^i \|\boldsymbol{w}_{inv}\| \sqrt{1-\alpha^2} (\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}^i_{sp}) \tag{10}$$

Let $\mathcal{S}^m_y$ denote the set of margin-points of purely-invariant classifier $c_{inv}(\boldsymbol{z})$ with class label y having $m_{inv}(\boldsymbol{z}) = 1$ and $\mathcal{S}^r_y$ contain rest of non-margin points having $m_{inv}(\boldsymbol{z}) > 1$ with the label y. Here "m" stands for margin-point in superscript of $\mathcal{S}$ and "r" stands for rest of point with label y. Below we show that for all non-margin points we can always choose $\alpha \in [0, 1)$ such that $m_\alpha(\boldsymbol{z}^i) > 1$. But we need spurious feature of margin point to be linearly separable for having $m_\alpha(\boldsymbol{z}^i) > 1$, given by Assm 2.3 for probing task or C.2 for main-task.

Margin Points For the margin-points in latent space, $\boldsymbol{z}^m \in \mathcal{S}^m_y$ we have $m_{inv}(\boldsymbol{z}^m) = 1$ and we need to show that there exist $\alpha \in [0, 1)$ such that $m_\alpha(\boldsymbol{z}^m) > 1$ for all $\boldsymbol{z}^m \in \mathcal{S}^m_y$. From Eq 10 we have:

$$m_\alpha(\boldsymbol{z}^m) = \alpha \cdot 1 + y \|\boldsymbol{w}_{inv}\| \sqrt{1-\alpha^2} (\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{x}^m_{sp}) > 1 \tag{11}$$

$$\left(\|\boldsymbol{w}_{inv}\| \sqrt{1-\alpha^2}\right) y (\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{x}^m_{sp}) > 1 - \alpha \tag{12}$$

From Assm 2.3 for probing task or C.2 for the main-task, we know that spurious-feature $\boldsymbol{z}_{sp}$ of margin-points are linearly-seperable w.r.t to task label y. Thus there exist an unit vector $\hat{\boldsymbol{\epsilon}}_{sp} \in \mathbb{R}^{d_{sp}}$ such that $y(\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}^m_{sp}) > 0$ for all $\boldsymbol{z}^m \in \mathcal{S}^m_y$. Also since $\alpha \in [0, 1)$, we have $(1-\alpha) > 0$ and $(\|\boldsymbol{w}_{inv}\| \sqrt{1-\alpha^2}) > 0$ since $\|\boldsymbol{w}_{inv}\| > 0$. If Assm 2.3 or C.2 (corresponding to the task) wouldn't have been satisfied then the above equation would have been inconsistent since right hand side is > 0 but left hand side wouldn't have been always greater than 0. Then achieving $m_\alpha(\boldsymbol{z}^m) > 1$ wouldn't have been possible. Let $\beta := (y(\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}))$ then squaring both sides and cancelling $(1-\alpha)$ since $\alpha \in [0, 1)$ we get:

$$\|\boldsymbol{w}_{inv}\|^2 \cancel{(1-\alpha)} (1+\alpha) \left(y(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{sp}} \cdot \boldsymbol{x}^m_{sp}) \right)^2 > \cancel{(1-\alpha)} (1-\alpha) \tag{13}$$

$$\|\boldsymbol{w}_{inv}\|^2 (1+\alpha) \beta^2 > (1-\alpha) \tag{14}$$

$$\|\boldsymbol{w}_{inv}\|^2 \beta^2 + \alpha \|\boldsymbol{w}_{inv}\|^2 \beta^2 > 1 - \alpha \tag{15}$$

$$\alpha \left(1 + \|\boldsymbol{w}_{inv}\|^2 \beta^2 \right) > \left(1 - \|\boldsymbol{w}_{inv}\|^2 \beta^2 \right) \tag{16}$$

After substituting back the value of β and rearranging we get:

$$\alpha > \frac{1 - \|\boldsymbol{w}_{inv}\|^2 \cdot \left(y(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{sp}} \cdot \boldsymbol{x}^m_{sp}) \right)^2}{1 + \|\boldsymbol{w}_{inv}\|^2 \cdot \left(y(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{sp}} \cdot \boldsymbol{x}^m_{sp}) \right)^2} = \alpha^1_{lb} \tag{17}$$

Since $\|\boldsymbol{w}_{inv}\|^2 \cdot \left(y(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{sp}} \cdot \boldsymbol{x}^m_{sp}) \right)^2 > 0$, the right hand side of above equation $\alpha^1_{lb} < 1$ which sets a new lower bound on allowed value of α. Thus there exist an $\alpha \in [0, 1)$ such that $m_p(\boldsymbol{x}^m) > 1$ for all $\boldsymbol{x}^m \in \mathcal{S}^m_y$.

Non-Margin Points For the non-margin points $\boldsymbol{z}^r \in \mathcal{S}^r_y$ in the latent space we have $m_{inv}(\boldsymbol{z}^r) > 1$. Let $\gamma := \min_{\boldsymbol{z}^r \in \mathcal{S}^r_y} \left(m_{inv}(\boldsymbol{z}^r) \right)$ thus we also have $\gamma > 1$. Let $\alpha \neq 0$ and we choose α such that:

$$\frac{1}{\alpha} < \gamma \tag{18}$$

$$\alpha > \frac{1}{\gamma} \tag{19}$$

Substituting the value of γ we get:

$$\alpha > \frac{1}{\min_{\boldsymbol{z}^r \in \mathcal{S}_y^r} \left(m_{inv}(\boldsymbol{z}^r)\right)} = \alpha_{lb}^2 \tag{20}$$

Since $\gamma > 1$, thus right hand side in above equation $\alpha_{lb}^2 < 1$ which sets a new lower bound on allowed values of α. Since $m_{inv}(\boldsymbol{z}^r) > \gamma > \frac{1}{\alpha}$ for all $\boldsymbol{x}^r \in \mathcal{S}_y^r$ for $\alpha \in (\alpha_{lb}^2, 1)$ (Eq 20), we can write $m_{inv}(\boldsymbol{z}^r) = \frac{1}{\alpha} + \eta(\boldsymbol{z}^r)$ where $\eta(\boldsymbol{z}^r) > 0$ for all $\boldsymbol{z}^r \in \mathcal{S}_y^r$. Now we need to show that there exist an $\alpha \in (\alpha_{lb}^2, 1)$ such that $m_p(\boldsymbol{z}^r) > 1$ for all $\boldsymbol{z}^r \in \mathcal{S}_y^r$. Thus from Eq 10 we have:

$$m_p(\boldsymbol{z}^r) = \alpha \cdot m_{inv}(\boldsymbol{z}^r) + \|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2}\Big(y\big(\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}\big)\Big) > 1 \tag{21}$$

$$\alpha \cdot (\frac{1}{\alpha} + \eta(\boldsymbol{z}^r)) + \|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2}\Big(y\big(\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}\big)\Big) > 1 \tag{22}$$

$$\|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2}\Big(y\big(\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}\big)\Big) > -\big(\alpha \cdot \eta(\boldsymbol{z}^r)\big) \tag{23}$$

Since $\alpha \in (\alpha_{lb}^2, 1)$, we have $(\alpha \cdot \eta(\boldsymbol{z}^r)) > 0$ and $\|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2} > 0$. Let's define $\delta(\boldsymbol{z}^r) := y\big(\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}\big)$. Thus for the latent-points $\boldsymbol{z}^r \in \mathcal{S}_y^r$ which have $\delta(\boldsymbol{z}^r) \geq 0$, Eq 23 is always satisfied since left side of inequality is greater than or equal to zero and right side is always less than zero. For the points for which $\delta(\boldsymbol{z}^r) < 0$ we have:

$$\|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2} \cdot \cancel{(-1)} \cdot |\delta(\boldsymbol{z}^r)| > \cancel{-}\big(\alpha \cdot \eta(\boldsymbol{z}^r)\big) \tag{24}$$

$$\|\boldsymbol{w}_{inv}\|\sqrt{1-\alpha^2}|\delta(\boldsymbol{z}^r)| < \big(\alpha \cdot \eta(\boldsymbol{z}^r)\big) \tag{25}$$

$$\|\boldsymbol{w}_{inv}\|^2\big(1-\alpha^2\big)\delta(\boldsymbol{z}^r)^2 < \big(\alpha \cdot \eta(\boldsymbol{z}^r)\big)^2 \tag{26}$$

$$\|\boldsymbol{w}_{inv}\|^2\delta(\boldsymbol{z}^r)^2 < \alpha^2 \cdot \Big(\eta(\boldsymbol{z}^r)^2 + \|\boldsymbol{w}_{inv}\|^2\delta(\boldsymbol{z}^r)^2\Big) \tag{27}$$

$$\alpha > \sqrt{\frac{\|\boldsymbol{w}_{inv}\|^2\delta(\boldsymbol{z}^r)^2}{\eta(\boldsymbol{z}^r)^2 + \|\boldsymbol{w}_{inv}\|^2\delta(\boldsymbol{z}^r)^2}} \tag{28}$$

Now different $\boldsymbol{z}^r$ will have different $\eta(\boldsymbol{z}^r)$ which will give different lower bound of α. Since the $m_\alpha(\boldsymbol{z}^r) > 1$ has to be satisfied for every point in $\boldsymbol{z}^r \in \mathcal{S}_y^r$. Thus it has to be satisfied for the point with minimum value $\eta_{min} = \min_{\boldsymbol{x}^r \in \mathcal{S}_y^r}(\eta(\boldsymbol{x}^r))$ which will give the tightest lower bound which α need to satisfy. Thus we have:

$$\alpha > \sqrt{\frac{\|\boldsymbol{w}_{inv}\|^2\delta(\boldsymbol{z}^r)^2}{\eta_{min}^2 + \|\boldsymbol{w}_{inv}\|^2\delta(\boldsymbol{x}^r)^2}} = \alpha_{lb}^3 \tag{29}$$

Since $\eta_{min} > 0$ and $\|\boldsymbol{w}_{inv}\|^2\delta(\boldsymbol{x}^r)^2 > 0$, the lower bound $\alpha_{lb}^3 < 1$ in Eq 29.

Finally, combining Eq 17, Eq 20 and Eq 29 let the overall lower bound of α be α_{lb} is:

$$\alpha_{lb} = \max\{\alpha_{lb}^1, \alpha_{lb}^2, \alpha_{lb}^3\} \tag{30}$$

This provides a way to construct a spurious-using classifier: given any *purely-invariant*, we can always choose $\alpha \in (\alpha_{lb}, 1)$ and construct a perturbed *spurious-using* classifier from Eq 7 which has a bigger margin than *purely-invariant*. Thus, given all the assumptions, there always exist a *spurious-using* classifier which has greater margin than the *purely-invariant* classifier completing our proof.

□

C.3. Proof of necessary condition

In this section we will show that the concept-probing classifier will use the spurious feature ($\boldsymbol{z}_m$) iff the spurious feature 2.3 is satisfied for the margin points of the clean concept-probing classifier when the concept-feature is 1-dimensional. Also, same line of reasoning will hold for the main-task classifier where we will show that main-task classifier will use the spurious feature ($\boldsymbol{z}_p$) iff spurious feature satisfies C.2 for the margin point of clean main-task classifier. Formally:

Lemma C.1 (Necessary Condition for concept-Probing Classifier)**.** Let the latent representation be frozen/non-trainable and disentangled such that $\boldsymbol{z} = [\boldsymbol{z}_m, z_p]$ where z_p is the concept-feature which is 1-dimensional and fully predictive (Assm 2.2) and $\boldsymbol{z}_m \in \mathbb{R}^{d_m}$. Let $c_p^*(\boldsymbol{z}) = w_p \cdot z_p$ be the desired clean/purely-invariant linear classifier trained using max-margin objective which only uses z_p for prediction. Then the concept-probing classifier trained using max-margin objective will be *spurious-using* i.e. $c_p(\boldsymbol{z}) = w_p \cdot z_p + \boldsymbol{w}_m \cdot \boldsymbol{z}_m$ where $\boldsymbol{w}_m \neq 0$ iff the spurious feature $\boldsymbol{z}_m$ is linearly separable w.r.t to probing task label y_p for the margin point of $c_p^*(\boldsymbol{z})$ (Assm 2.3) .

Lemma C.2 (Necessary Condition for Main-task Classifier)**.** Let the latent representation be frozen/non-trainable and disentangled such that $\boldsymbol{z} = [\boldsymbol{z}_m, z_p]$ where z_m is the main-task feature which is 1-dimensional and fully predictive (Assm C.1) and $\boldsymbol{z}_p \in \mathbb{R}^{d_p}$. Let $c_m^*(\boldsymbol{z}) = w_m \cdot z_m$ be the desired clean/purely-invariant linear classifier trained using max-margin objective which only uses z_m for prediction. Then the main-task classifier trained using max-margin objective will be *spurious-using* i.e. $c_m(\boldsymbol{z}) = w_m \cdot z_m + \boldsymbol{w}_p \cdot \boldsymbol{z}_p$ where $\boldsymbol{w}_p \neq 0$ iff the spurious feature $\boldsymbol{z}_p$ is linearly separable w.r.t to main task label y_m for the margin point of $c_m^*(\boldsymbol{z})$ (Assm C.2) .

Since proof of both Lemma C.1 and C.2 follows same line of reasoning for brevity, following Sec C.2, we will prove the lemma for a general classifier $c(\boldsymbol{z})$ trained using max-margin objective to predict the task-label y. Let the latent representation be of form $\boldsymbol{z} = [z_{inv}, \boldsymbol{z}_{sp}]$ where $z_{inv} \in \mathbb{R}$ is the feature causally derived from the task-label y and $\boldsymbol{z}_{sp} \in \mathbb{R}^d_{sp}$ is the feature spuriously correlated to task label y. With respect to concept-probing classifier $c_p(\boldsymbol{z})$ in Lemma C.1 $\boldsymbol{z}_{inv} := \boldsymbol{z}_p$ and $\boldsymbol{z}_{sp} := \boldsymbol{z}_m$. Similarly, for the main-task classifier in Lemma C.2, $\boldsymbol{z}_{inv} := \boldsymbol{z}_m$ and $\boldsymbol{z}_{sp} := \boldsymbol{z}_p$.

Proof of Lemma C.1 and C.2. Now the goal is to show that Assm 2.3 for probing classifier or Assm C.2 for the main-task classifier is necessary for obtaining a *spurious-using* classifier for the case when $\boldsymbol{z}_{inv}$ is one-dimensional. We show this by assuming that optimal classifier is *spurious-using* even when Assm 2.3 or C.2 breaks and then show that this will lead to contradiction.

Contradiction Assumption: Formally, let's assume that the optimal classifier for the given classification task is *spurious-using* $c_*(\boldsymbol{z})$, where:

$$c_*(\boldsymbol{x}) = w^*_{inv} \cdot z_{inv} + \|\boldsymbol{w}^*_{sp}\|(\hat{\boldsymbol{w}}^*_{sp} \cdot \boldsymbol{z}_{sp}) \tag{31}$$

where $\|\boldsymbol{w}^*_{sp}\| \neq 0$ and $\hat{\boldsymbol{w}}^*_{sp} \in \mathbb{R}^{d_{sp}}$ is a unit vector in spurious subspace with dimension d_{sp}.

Let $c_{inv}(\boldsymbol{z}) = w_{inv} \cdot z_{inv}$ be the optimal *purely-invariant* classifier. Let both $c_*(\boldsymbol{z})$ and $c_{inv}(\boldsymbol{z})$ be trained using the max-margin objective using *MM-Denominator* formulation in Eq 4. Thus from the constraints of this formulation (Eq 3) we have:

$$m_*(\boldsymbol{z}) = y \cdot c_*(\boldsymbol{z}) = y \cdot (w^*_{inv} \cdot z_{inv} + \|\boldsymbol{w}^*_{sp}\|(\hat{\boldsymbol{w}}^*_{sp} \cdot \boldsymbol{z}_{sp})) \geq 1 \ \ ,\& \tag{32}$$

$$m_{inv}(\boldsymbol{z}) = y \cdot c_{inv}(\boldsymbol{z}) = y \cdot (w_{inv} \cdot z_{inv}) \geq 1 \tag{33}$$

From Assm 2.2 or C.1, the invariant feature $\boldsymbol{z}_{inv}$ is 100% predictive and linearly separable w.r.t task label y. Then without loss of generality let's assume that:

$$z_{inv} > 0, \ \text{ when } \ y = +1 \tag{34}$$

$$z_{inv} < 0, \ \text{ when } \ y = -1 \tag{35}$$

From Eq 34 and 35 we have $y \cdot z_{inv} > 0$ thus from Eq 33 we get:

$$w_{inv} \geq 0 \tag{36}$$

Also, from our *contradiction-assumption* the max-margin trained classifier is *spurious-using*, thus the norm of parameters of $c_*(\boldsymbol{z})$ is less or equal to $c_{inv}(\boldsymbol{z})$ (Eq 4). Thus we have:

$$\sqrt{(w^*_{inv})^2 + (\|\boldsymbol{w}^*_{sp}\|)^2} \leq |w_{inv}| \tag{37}$$

$$\implies |w^*_{inv}| < |w_{inv}| \qquad (\|\boldsymbol{w}^*_{sp}\| \neq 0) \tag{38}$$

$$\implies |w^*_{inv}| < w_{inv} \qquad (w_{inv} < 0, Eq\ 36) \tag{39}$$

$$\implies w^*_{inv} < w_{inv} \tag{40}$$

Suppose Assm 2.3 for concept-probing task or Assm C.2 for the main-task breaks, then we have one of the following two case:

1. Opposite Side Failure: When the spurious part of margin point on opposite side of decision-boundary ($c_*(\boldsymbol{z}) = 0$) are not linearly-separable. Formally, there exist two datapoint, $P^{m+} := [z_{inv}^{m+}, \boldsymbol{z}_{sp}^{m+}]$ and $P^{m-} := [z_{inv}^{m-}, \boldsymbol{z}_{sp}^{m-}]$ such that they are margin points of *purely-invariant* classifier $c_{inv}(\boldsymbol{z})$ where P^{m+} has class label $y = +1$ and P^{m-} has class label $y = -1$ and $\forall \hat{\boldsymbol{\epsilon}}_{sp} \in \mathbb{R}^{d_{sp}}$, the spurious feature $\boldsymbol{z}_{sp}$ of both the points lies on same side of $\hat{\boldsymbol{\epsilon}}_{sp}$ i.e:
$$\big((\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}^{m+}) \cdot (\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}^{m-})\big) \geq 0 \tag{41}$$

2. Same Side Failure: When the spurious part of margin point on same side of decision-boundary ($c_*(\boldsymbol{z}) = 0$) are always linearly-separable. There exist two datapoints, $P_y^{m1} := [x_{inv}^{m1}, \boldsymbol{z}_{sp}^{m1}]$ and $P_y^{m2} := [x_{inv}^{m2}, \boldsymbol{z}_{sp}^{m2}]$ such that they are margin point of *purely-invariant* classifier $c_{inv}(\boldsymbol{z})$ and both points have same class label y and $\forall \hat{\boldsymbol{\epsilon}}_{sp} \in \mathbb{R}^{d_{sp}}$, w.l.o.g we have:
$$\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}^{m1} \geq 0, \;\; \& \;\; \hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{z}_{sp}^{m2} \leq 0. \tag{42}$$

We will use the following two lemma to proceed with our proof:

Lemma C.3. If Assm 2.3 or C.2 breaks by *opposite-side failure* mode it leads to contradiction.

Lemma C.4. If Assm 2.3 or C.2 breaks by *same-side failure* mode it leads to contradiction.

This implies that our *contradiction-assumption* which said that the max-margin trained optimal classifier is *spurious-using* even when Assm 2.3 or C.2 breaks, is wrong. Thus Assm 2.3 for concept-probing task or Assm C.2 for main-task is necessary for the optimal max-margin classifier to be *spurious-using* thus completing our proof. □

Proof of Lemma C.3. We have two point $P^{m+} := [z_{inv}^{m+}, \boldsymbol{z}_{sp}^{m+}]$ and $P^{m-} := [z_{inv}^{m-}, \boldsymbol{z}_{sp}^{m-}]$ which breaks the Assm 2.3 or C.2. From Eq 34, $z_{inv} > 0$ for all the points with label $y = 1$, thus we have $x_{inv}^{m+} > 0$ and using Eq 40 ($w_{inv}^* < w_{inv}$) we get:

$$w_{inv}^* < w_{inv} \tag{43}$$
$$w_{inv}^* \cdot z_{inv}^{m+} < w_{inv} \cdot z_{inv}^{m+} \tag{44}$$
$$w_{inv}^* \cdot z_{inv}^{m+} < 1 \tag{45}$$

where the right hand side $w_{inv} \cdot z_{inv}^{m+} = 1$ since P^{m+} is the margin-point of $c_{inv}(\boldsymbol{z})$ (Eq 33). Similarly from Eq 35, $z_{inv} < 0$ for all the points with label $y = -1$, thus we have $z_{inv}^{m-} < 0$ and using Eq 40 ($w_{inv}^* < w_{inv}$) we get:

$$w_{inv}^* < w_{inv} \tag{46}$$
$$(-1) \cdot w_{inv}^* \cdot z_{inv}^{m-} < (-1) \cdot w_{inv} \cdot z_{inv}^{m-} \tag{47}$$
$$(-1) \cdot w_{inv}^* \cdot z_{inv}^{m-} < 1 \tag{48}$$

where the right hand side $(-1) \cdot (w_{inv}^p \cdot x_{inv}^{m-}) = 1$ since P^{m-} is the margin-point of $c_{inv}(\boldsymbol{z})$ (Eq 33).

Next from Eq 32 we have $m_*(\boldsymbol{z}) \geq 1$ for all "$\boldsymbol{z}$" hence it is also true for P^{m+} with $y = 1$ and P^{m-} with $y = -1$. Then:

$$m_*(P^{m+}) = y \cdot c_*(P^{m+}) = 1 \cdot \left\{ w_{inv}^* z_{inv}^{m+} + \|\boldsymbol{w}_{sp}^*\| \big(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{z}_{sp}^{m+}\big) \right\} \geq 1 \tag{49}$$
$$\implies w_{inv}^* z_{inv}^{m+} + \|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m+} \geq 1 \tag{50}$$
$$\implies w_{inv}^* z_{inv}^{m+} \geq 1 - \|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m+} \tag{51}$$

where $\beta^{m+} = \big(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{x}_{sp}^{m+}\big)$. Also we have:

$$m_*(P^{m-}) = y \cdot c_*(P^{m-}) = -1 \cdot \left\{ w_{inv}^* z_{inv}^{m-} + \|\boldsymbol{w}_{sp}^*\| \big(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{z}_{sp}^{m-}\big) \right\} \geq 1 \tag{52}$$
$$\implies -w_{inv}^* z_{inv}^{m-} + \|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m-} \geq 1 \tag{53}$$
$$\implies -w_{inv}^* z_{inv}^{m-} \geq 1 + \|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m-} \tag{54}$$

where $\beta^{m-} = \left(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{x}_{sp}^{m-}\right)$. From Eq 41 we have $\left((\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{x}_{sp}^{m+}) \cdot (\hat{\boldsymbol{\epsilon}}_{sp} \cdot \boldsymbol{x}_{sp}^{m-})\right) \geq 0$ for all $\hat{\boldsymbol{\epsilon}}_{sp} \in \mathbb{R}^{d_{sp}}$ which states the *opposite-side failure* of Assm 2.3 or C.2. Thus:

$$\beta^{m+} \cdot \beta^{m-} \geq 0 \tag{55}$$

Now we will show that Eq 45, 48, 51 and 54 cannot be satisfied simultaneously for any allowed value of β^{m+} and β^{m-} (given by Eq 55) which are:

1. $\beta^{m+} > 0$ and $\beta^{m-} > 0$: From Eq 54 we have $-w_{inv}^* x_{inv}^{m-} > 1$ since $\|\boldsymbol{w}_{sp}^*\| \neq 0$ and $\beta^{m-} > 0$. But from Eq 48 we have $-w_{inv}^* x_{inv}^{m-} < 1$ which is a contradiction.
2. $\beta^{m+} < 0$ and $\beta^{m-} < 0$: From Eq 51 we have $w_{inv}^* x_{inv}^{m+} > 1$ since $\|\boldsymbol{w}_{sp}^*\| \neq 0$ and $\beta^{m+} < 0$. But from Eq 45 we have $w_{inv}^* x_{inv}^{m+} < 1$ which is a contradiction.
3. $\beta^{m+} = 0$ and $\beta^{m-} \in \mathbb{R}$: From Eq 51 we have $w_{inv}^* x_{inv}^{m+} \geq 1$ but from Eq 45 we have $w_{inv}^* x_{inv}^{m+} < 1$ which is a contradiction.
4. $\beta^{m+} \in \mathbb{R}$ and $\beta^{m-} = 0$: From Eq 54 we have $-w_{inv}^* x_{inv}^{m-} \geq 1$ but from Eq 48 we have $-w_{inv}^* x_{inv}^{m-} < 1$ which is a contradiction.

Thus we have a contradiction for all the possible values β^{m+} and β^{m-} could take, completing the proof of this lemma.

□

Proof of Lemma C.4. We have two margin-points $P_y^{m1} := [z_{inv}^{m1}, \boldsymbol{z}_{sp}^{m1}]$ and $P_y^{m2} := [z_{inv}^{m2}, \boldsymbol{z}_{sp}^{m2}]$ which breaks Assm 2.3 or C.2. From Eq 34 and Eq 35 we have $y \cdot z_{inv}^{m1} > 0$ and $y \cdot z_{inv}^{m2} > 0$. Using Eq 40 ($w_{inv}^* < w_{inv}$) we get:

$$w_{inv}^* < w_{inv} \tag{56}$$

$$w_{inv}^* \cdot (y \cdot z_{inv}^{mj}) < w_{inv} \cdot (y \cdot z_{inv}^{mj}) \tag{57}$$

$$y \cdot (w_{inv}^* \cdot z_{inv}^{mj}) < 1 \tag{58}$$

where $j \in \{1, 2\}$ and right hand side $w_{inv} \cdot (y \cdot z_{inv}^{mj}) = 1$ since P_y^{mj} is the margin point of purely-invariant classifier $c_{inv}(\boldsymbol{z})$ (Eq 33).

From Eq 32 we have $m_*(\boldsymbol{x}) \geq 1$ for all "$\boldsymbol{z}$" thus also true for P_y^{m1} and P_y^{m2}. Then:

$$m_*(P_y^{m1}) = y \cdot c_*(P_y^{m1}) = y \cdot \left\{ w_{inv}^* z_{inv}^{m1} + \|\boldsymbol{w}_{sp}^*\| \left(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{z}_{sp}^{m1}\right) \right\} \geq 1 \tag{59}$$

$$\implies y \cdot (w_{inv}^* z_{inv}^{m1}) + y \cdot (\|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m1}) \geq 1 \tag{60}$$

$$\implies y \cdot (w_{inv}^* z_{inv}^{m1}) \geq 1 - y \cdot (\|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m1}) \tag{61}$$

where $\beta^{m1} = \left(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{z}_{sp}^{m1}\right)$. Also we have:

$$m_*(P_y^{m2}) = y \cdot c_*(P_y^{m2}) = y \cdot \left\{ w_{inv}^* z_{inv}^{m2} + \|\boldsymbol{w}_{sp}^*\| \left(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{z}_{sp}^{m2}\right) \right\} \geq 1 \tag{62}$$

$$\implies y \cdot (w_{inv}^* z_{inv}^{m2}) + y \cdot (\|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m2}) \geq 1 \tag{63}$$

$$\implies y \cdot (w_{inv}^* z_{inv}^{m2}) \geq 1 - y \cdot (\|\boldsymbol{w}_{sp}^*\| \cdot \beta^{m2}) \tag{64}$$

where $\beta^{m2} = \left(\hat{\boldsymbol{w}}_{sp}^* \cdot \boldsymbol{z}_{sp}^{m2}\right)$. Now from Eq 42 we have $\beta^{m1} \geq 0$ and $\beta^{m2} \leq 0$ for all unit vector $\hat{\boldsymbol{\epsilon}}_{sp} \in \mathbb{R}^{d_{sp}}$ which states the *same-side* failure mode of Assm 2.3 or C.2. Now we will show that for all allowed value of β^{m1} and β^{m2}, Eq 58 61 and 64 will lead to contradiction. Following are the cases for different allowed values of β^{m1} and β^{m2}:

1. $\beta^{m1} = 0$ and $\beta^{m2} \in \mathbb{R}$: Substituting $\beta^{m1} = 0$ in Eq 61 we get $y \cdot (w_{inv}^* z_{inv}^{m1}) \geq 1$, but from Eq 58 we have $y \cdot (w_{inv}^* z_{inv}^{m1}) < 1$. Thus we have a contradiction.

2. $\beta^{m1} \in \mathbb{R}$ and $\beta^{m2} = 0$: Substituting $\beta^{m2} = 0$ in Eq 64 we get $y \cdot (w^*_{inv} z^{m2}_{inv}) \geq 1$, but from Eq 58 we have $y \cdot (w^*_{inv} z^{m2}_{inv}) < 1$. Thus we have a contradiction.
3. $\beta^{m1} > 0$, $\beta^{m2} < 0$ and $y = (+1)$: Substituting $\beta^{m2} < 0$ and $y = (+1)$ in Eq 64 we get $y \cdot (w^*_{inv} z^{m2}_{inv}) \geq 1$, but from Eq 58 we have $y \cdot (w^*_{inv} z^{m2}_{inv}) < 1$. Thus we have a contradiction.
4. $\beta^{m1} > 0$, $\beta^{m2} < 0$ and $y = (-1)$: Substituting $\beta^{m1} > 0$ and $y = (-1)$ in Eq 61 we get $y \cdot (w^*_{inv} z^{m1}_{inv}) \geq 1$, but from Eq 58 we have $y \cdot (w^*_{inv} z^{m1}_{inv}) < 1$. Thus we have a contradiction.

Thus we have a contradiction for all the possible values β^{m1}, β^{m2} and y could take, completing the proof of this lemma.

□

C.4. Problem with learning a *clean* main-task classifier

In this section we will restate the assumptions and results of 2.1 for the main-task classifier (instead of the probing classifier) and show that the same results will hold.

Assm 2.1 remains the same since it is made on the latent-representation being disentangled and frozen/non-trainable. Next, parallel to Assm 2.2 show that even when main-task feature is 100% predictive of main-task and a linearly separable, the trained main-task classifier will also use the concept-features. Formally,

Assumption C.1 (main-task feature Linear Separability)**.** The main-task features of the latent representation ($\boldsymbol{z}_m$) for every point are linearly separable/fully predictive for the main-task labels y_m, i.e $y^i_m \cdot (\hat{\boldsymbol{\epsilon}}_m \cdot \boldsymbol{z}^i_m + b_m) > 0$ for all datapoints $(\boldsymbol{x}^i, y^i_m)$ for some $\hat{\boldsymbol{\epsilon}}_m \in \mathbb{R}^{d_p}$. For the case of zero-centered latent space, we have $b_m = 0$.

Next similar to Assm 2.3, we define the spurious correlation between main-task and concept label: a function using $\boldsymbol{z}_p$ may also be able to classify correctly w.r.t. main-task label on some non-empty subset of points.

Assumption C.2 (Main-Task Spurious Correlation)**.** For a subset of training points $\mathcal{S} \subset \mathcal{D}_p$, main-task label y_m is linearly separable using $\boldsymbol{z}_p$ i.e $y^i_m \cdot (\hat{\boldsymbol{\epsilon}}_p \cdot \boldsymbol{z}^i_p + b_p) > 0$ for all the points in the dataset, $\hat{\boldsymbol{\epsilon}}_p \in \mathbb{R}^{d_p}$ and $b_p \in \mathbb{R}$. For the case of zero-centered latent space we have $b_p = 0$.

Next we rephrase Lemma 2.1 which shows that for only a *few special* points if the concept-features z_p are linearly-separable w.r.t. to main task classifier y_m, then the main-task classifier $c_m(\boldsymbol{z})$ will use those features.

Lemma C.5 (Sufficient Condition for Main-task Classifier)**.** Let the latent representation be frozen and disentangled such that $\boldsymbol{z} = [\boldsymbol{z}_m, \boldsymbol{z}_p]$ (Assm 2.1), where main-task-features $\boldsymbol{z}_m$ be fully predictive (Assm C.1). Let $c^*_m(\boldsymbol{z}) = \boldsymbol{w}_m \cdot \boldsymbol{z}_m$ be the desired/clean linear main-task classifier trained using max-margin objective (Suppl C.1) which only uses $\boldsymbol{z}_m$ for its prediction. Let $\boldsymbol{z}_p$ be the spurious feature s.t. for the margin points of $c^*_m(\boldsymbol{z})$, $\boldsymbol{z}_p$ be linearly-separable w.r.t. task label y_m (Assm C.2). Then, assuming the latent space is centered around $\mathbf{0}$, the main-task classifier trained using max-margin objective will be of form $c_m(\boldsymbol{z}) = \boldsymbol{w}_m \cdot \boldsymbol{z}_m + \boldsymbol{w}_p \cdot \boldsymbol{z}_p$ where $\boldsymbol{w}_p \neq \mathbf{0}$.

The proof of Lemma C.5 is identical to Lemma 2.1 and is provided in Suppl C.2.

D. Null-Space Removal Failure: Setup and Proof of Theorem 2.2

D.1. Null-Space Setup

As described in Sec 2, the given main-task classifier have an encoder $h : \boldsymbol{X} \rightarrow \boldsymbol{Z}$ mapping the input $\boldsymbol{X}$ to latent representation $\boldsymbol{Z}$. Post that the main-task classifier $c_m : \boldsymbol{Z} \rightarrow Y_m$ is used to predict the main-task label y^i_m from latent representation $\boldsymbol{z}^i$ for every input $\boldsymbol{x}^i$. Given this (pre) trained main-task classifier the goal of post-hoc removal method to remove any undesired/sensitive/spurious concept from the latent representation $\boldsymbol{Z}$ without retraining the encoder h or main-task classifier $c_m(\boldsymbol{z})$.

The null space method (Ravfogel et al., 2020; Elazar et al., 2021), henceforth referred as *INLP*, is one such post-hoc removal method which removes a concept from latent space by projecting the latent space to a subspace that is not discriminative of that attribute. First, it estimates the subspace in the latent space discriminative of the concept we want to remove by training a probing classifier $c_p(\boldsymbol{z}) \rightarrow y_p$, where y_p is the concept label. (Ravfogel et al., 2020) used a linear probing classifier ($c_p(\boldsymbol{z})$) to ensure that the any linear classifier cannot recover the removed concept from modified latent representation $\boldsymbol{z}'$ and hence the main task classifier ($c_m(\boldsymbol{z}')$) also become invariant to removed attribute. Let linear concept-probing classifier $c_p(\boldsymbol{z})$ be

parametrized by a matrix W, and null-space of matrix W is defined as space $N(W) = \{\boldsymbol{z}|W\boldsymbol{z} = \mathbf{0}\}$. Give the basis vectors for the $N(W)$ we can construct the projection matrix $P_{N(W)}$ such that $W(P_{N(W)}\boldsymbol{z}) = \mathbf{0}$ for all $\boldsymbol{z}$. This projection matrix is defined as the guarding operator $g \coloneqq P_{N(\mathcal{W})}$, when applied on the $\boldsymbol{z}$ will remove the features which are discriminative of undesired concept from $\boldsymbol{z}$ as estimated by $c_p(\boldsymbol{z})$. For the setting when Y_p is binary we have:

$$P_{N(W)} = I - \hat{w}\hat{w}^T \tag{65}$$

where I is the identity matrix and $\hat{w}$ is the unit vector in the direction of parameters of classifier $c_p(\boldsymbol{z})$ ((Ravfogel et al., 2020)). Also, the authors recommend running this removal step for multiple iterations to ensure the unwanted concept is removed completely. Thus after S steps of removal, the final guarding function is:

$$g \coloneqq \prod_{i=1}^{S} P^i_{N(W)} \tag{66}$$

where $P^i_{N(W)}$ is the projection matrix at i^{th} removal step. Amnesic Probing ((Elazar et al., 2021)) builds upon this idea for testing whether concept is being used by a given pre-trained classifier or not. The core idea is to remove the concept we want to test from the latent representation and if the prediction of the given classifier is influenced by this removal then the attribute was being used by the given classifier otherwise not.

D.2. Null-Space Removal Failure : Proof of Theorem 2.2

Theorem 2.2. Let the latent representation $\boldsymbol{z}$ be disentangled (Assm 2.1), $\boldsymbol{c}_m(\boldsymbol{z})$ be the pre-trained main-task classifier, and $\boldsymbol{c}_p(\boldsymbol{z})$ the probing classifier used by INLP to remove the unwanted feature $\boldsymbol{z}_p$ from the latent representation. Let the assumptions of Lemma 2.1 be true for the probing classifier $c_p(\boldsymbol{z})$, trained using max-margin objective.

1. **Mixing:** After the first projection step of linear-INLP, the dimensions of $\boldsymbol{z}$ gets mixed such that $\boldsymbol{z}^{i(1)} = [g(\boldsymbol{z}^i_m, \boldsymbol{x}^i_p), f(\boldsymbol{z}^i_p, \boldsymbol{z}^i_m)] \neq [\boldsymbol{z}^i_m, \boldsymbol{z}^i_p]$. Also, this mixing is non-invertible with subsequent projection steps. Thus, the latent space is no longer disentangled and removal of concept-causal features will also lead to removal of task-specific features.

2. **Removal:** The L2-norm of the latent representation $\boldsymbol{z}$ decreases with every projection step as long as the parameters of probing classifier at a step ($\boldsymbol{w}^k$) does not lie completely in the space spanned by parameters of previous probing classifiers i.e. span($\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{k-1}$). Thus, after sufficiently many steps, INLP can destroy all information in the representation, $\boldsymbol{z}^{i(\infty)} = [\mathbf{0}, \mathbf{0}]$.

The proof of Theorem 2.2 proceeds in two steps:

1. We first show that after first step of null-space projection (INLP), both the main-task feature and concept-causal feature gets *emph*. Then we show that the projection operation is non-invertible thus, *mixing* cannot be corrected in subsequent projection steps.

2. Next we show that the projection operation is lossy, i.e removes the norm of latent representation under some condition. Hence after sufficient steps, INLP could destroy all the information in latent representation.

Proof of Theorem 2.2. **First Claim.** Let $c_p(\boldsymbol{z}) = \boldsymbol{w}_p\boldsymbol{z}_p + \boldsymbol{w}_m\boldsymbol{z}_m$ be the linear concept-probing classifier trained to predict the probing label y_p from the latent representation $\boldsymbol{z}$. Since all the assumptions of Lemma 2.1 are satisfied for the probing classifier $c_p(\boldsymbol{z})$, it is *spurious using*, i.e., $\boldsymbol{w}_m \neq \mathbf{0}$ and $\boldsymbol{w}_p \neq \mathbf{0}$. Since the concept-probing label y_p is binary the projection matrix for the first step of INLP removal is defined as $P^1_{N(W)} = I - \hat{\boldsymbol{w}}\hat{\boldsymbol{w}}^T$ where $\hat{\boldsymbol{w}}^T = [\hat{\boldsymbol{w}}_m, \hat{\boldsymbol{w}}_p]$, $\hat{\boldsymbol{w}}_m$ and $\hat{\boldsymbol{w}}_p$ are the unit norm parameters of $c_p(\boldsymbol{z})$ i.e $\boldsymbol{w}_m$ and $\boldsymbol{w}_p$ respectively. On applying this projection on the latent space representation $\boldsymbol{z}^i$

we get:

$$\begin{bmatrix} \boldsymbol{z}_m^{i(1)} \\ \boldsymbol{z}_p^{i(1)} \end{bmatrix} = \left(I - \begin{bmatrix} \hat{\boldsymbol{w}}_m \\ \hat{\boldsymbol{w}}_p \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{w}}_m & \hat{\boldsymbol{w}}_p \end{bmatrix} \right) \begin{bmatrix} \boldsymbol{z}_m \\ \boldsymbol{z}_p \end{bmatrix} \tag{67}$$

$$= \begin{bmatrix} \boldsymbol{z}_m \\ \boldsymbol{z}_p \end{bmatrix} - \left(\frac{c_p(\boldsymbol{z}^i)}{\|\boldsymbol{w}\|} \right) \cdot \begin{bmatrix} \hat{\boldsymbol{w}}_m \\ \hat{\boldsymbol{w}}_p \end{bmatrix} \tag{68}$$

$$= \begin{bmatrix} \boldsymbol{z}_m - \tilde{c}_p(\boldsymbol{z}^i)\hat{\boldsymbol{w}}_m \\ \boldsymbol{z}_p - \tilde{c}_p(\boldsymbol{z}^i)\hat{\boldsymbol{w}}_p \end{bmatrix} \qquad \text{define } \tilde{c}_p(\boldsymbol{z}^i) := \left(\frac{c_p(\boldsymbol{z}^i)}{\|\boldsymbol{w}\|} \right) \tag{69}$$

$$= \begin{bmatrix} g(\boldsymbol{z}_m, \boldsymbol{z}_p) \\ h(\boldsymbol{z}_m, \boldsymbol{x}_p) \end{bmatrix} \tag{70}$$

Next we will show that $g(\boldsymbol{z}_m, \boldsymbol{z}_p) \neq \boldsymbol{z}_m$ and $h(\boldsymbol{z}_m, \boldsymbol{x}_p) \neq \boldsymbol{z}_p$. For $g(\boldsymbol{z}_m, \boldsymbol{z}_p) = \boldsymbol{z}_m$ we need:

$$\cancel{\boldsymbol{z}_m} - \tilde{c}_p(\boldsymbol{z}^i) \cdot \hat{\boldsymbol{w}}_m = \cancel{\boldsymbol{z}_m} \tag{71}$$

$$\tilde{c}_p(\boldsymbol{z}^i) \cdot \hat{\boldsymbol{w}}_m = \boldsymbol{0} \tag{72}$$

We know that $\boldsymbol{w}_m \neq \boldsymbol{0}$ and since the probing classifier is trained using max-margin objective using the *MM-Denominator* formulation we have $y_p^i \cdot c_p(\boldsymbol{z}^i) \geq 1$ from Eq 3. Thus $c_p(\boldsymbol{z}^i) \in (\infty, -1] \cup [1, \infty)$ and $\tilde{c}_p(\boldsymbol{z}^i) \in (\infty, \frac{-1}{\|\boldsymbol{w}\|}] \cup [\frac{1}{\|\boldsymbol{w}\|}, \infty) \neq 0$. Since $\boldsymbol{w}_m \neq 0$ and $\tilde{c}_p(\boldsymbol{z}^i) \neq 0$, Eq 72 cannot be true hence $g(\boldsymbol{z}_m, \boldsymbol{z}_p) \neq \boldsymbol{z}_m$. Similarly for $h(\boldsymbol{z}_m, \boldsymbol{x}_p) = \boldsymbol{z}_p$ we need:

$$\cancel{\boldsymbol{z}_p} - \tilde{c}_p(\boldsymbol{z}^i) \cdot \hat{\boldsymbol{w}}_p = \cancel{\boldsymbol{z}_p} \tag{73}$$

$$\tilde{c}_p(\boldsymbol{z}^i) \cdot \hat{\boldsymbol{w}}_p = \boldsymbol{0} \tag{74}$$

And again since $\boldsymbol{w}_p \neq \boldsymbol{0}$ and $\tilde{c}_p(\boldsymbol{z}^i) \neq 0$, Eq 74 cannot be true hence $h(\boldsymbol{z}_m, \boldsymbol{z}_p) \neq \boldsymbol{z}_p$. Hence both concept-feature $\boldsymbol{z}_p$ and the main-task feature $\boldsymbol{z}_m$ got mixed after the first step of projection.

Next, the following lemma proves that the above projection matrix ($P^1_{N(W)}$) is non-invertible. The subsequent steps of INLP applies projection transformation which can be combined into one single matrix $P^{>1}_{N(W)} = \prod_{j>1} P^j_{N(W)}$. In order for mixing to be reversed, we need $P^{>1}_{N(W)} \times P^1_{N(W)} = I$, thus we need $P^{>1}_{N(W)} = (P^1_{N(W)})^{-1}$ which is not possible from Lemma D.1. Hence the mixing of main-task feature and the concept-feature which happened after first step of projection couldn't be corrected in the subsequent steps of INLP thus completing the first part of our proof.

Lemma D.1. The projection matrix $P^1_{N(W)}$ at any projection step of INLP is non invertible.

Proof of Lemma D.1. The projection matrix for binary target case is defined as $P := P^1_{N(W)} = I - A$ where $A = \hat{w}\hat{w}^T$ be a $n \times n$ matrix. We can see that A is a symmetric matrix. From Eq. 6 in (Weisstein)), every symmetric matrix is diagonalizable. Hence, we can write $A = Q\Lambda Q^T$, where Q is a some orthonormal matrix such that $QQ^T = I$ and $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ be a $n \times n$ diagonal matrix where the diagonal entries $(\lambda_1 \ldots \lambda_n)$ are the eigen-values of A. Since $QQ^T = I$ we can write $P = I - A = QQ^T - Q\Lambda Q^T = Q(I - \Lambda)Q^T$. For the projection matrix P to be invertible P^{-1} should exist. We have:

$$P^{-1} = \left(Q(I - \Lambda)Q^T \right)^{-1} \tag{75}$$

$$= (Q^T)^{-1}(I - \Lambda)^{-1}Q^{-1} \tag{76}$$

$$= Q(I - \Lambda)^{-1}Q^T \tag{77}$$

So projection matrix is only invertible when $(I - \Lambda)$ is invertible. We will show next that $(I - \Lambda)$ is not invertible thus completing our proof. We have $I - \Lambda = diag(1 - \lambda_1, \ldots, 1 - \lambda_n)$, hence:

$$(I - \Lambda)^{-1} = diag(\frac{1}{1 - \lambda_1}, \ldots, \frac{1}{1 - \lambda_2}) \tag{78}$$

Now, if one of the eigen values of A is 1, then the diagonal matrix $(I - \Lambda)$ is not invertible. If one of the eigen values of A is 1, then there exist an eigen-vector $\boldsymbol{x}$ such that $A\boldsymbol{x} = \hat{\boldsymbol{w}}\hat{\boldsymbol{w}}^T \times \boldsymbol{x} = 1 \times \boldsymbol{x}$. The vector $\boldsymbol{x} = \hat{\boldsymbol{w}}$ is the eigen vector of A

with eigen-value 1: $A\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}\hat{\boldsymbol{w}}^T \times \hat{\boldsymbol{w}} = 1 \times \hat{\boldsymbol{w}}$ since $\hat{\boldsymbol{w}}^T \times \hat{\boldsymbol{w}} = 1$ as it is a unit vector. Hence the projection matrix is not invertible.

□

Second Claim. Now for proving these second statement, we will make use of the following lemma:

Lemma D.2. After every projection step of INLP, the norm of every latent representation $\boldsymbol{z}^i$ decreases, i.e., $\|\boldsymbol{z}^{i(k)}\| < \|\boldsymbol{z}^{i(k-1)}\|$ for step k and $k-1$, if 1) $\boldsymbol{z}^{i(k-1)} \neq \mathbf{0}$ and 2) the parameters of probing classifier in step "k" i.e $\hat{\boldsymbol{w}}^k$ don't lie in the space spanned by parameters of previous concept-probing classifier, span($\hat{\boldsymbol{w}}^1, \ldots, \hat{\boldsymbol{w}}^{k-1}$).

Proof of Lemma D.2. After $k-1$-steps of INLP let the latent space representation $\boldsymbol{z}^i$ be denoted as $\boldsymbol{z}^{i(k-1)}$. Let $\hat{\boldsymbol{w}}^k$ be the parameters of classifier $c_p(\boldsymbol{z}^{k-1})$ trained to predict the concept–probing label y_p which we want to remove at step k. Then before the projection step in the kth iteration of the INLP we can write $\boldsymbol{z}^{k-1}$ as:

$$\boldsymbol{z}^{k-1} = \boldsymbol{z}_{\hat{\boldsymbol{w}}^k}\hat{\boldsymbol{w}}^k + \boldsymbol{z}_{N(\hat{\boldsymbol{w}}^k)}N(\hat{\boldsymbol{w}}^k) \tag{79}$$

where $\{\hat{\boldsymbol{w}}^k, N(\hat{\boldsymbol{w}}^k)\}$ is the basis set and $N(\hat{\boldsymbol{w}}^k)$ is the null-space of $\hat{\boldsymbol{w}}^k$. The parameter $\boldsymbol{w}^k$ in this new basis is:

$$\boldsymbol{w}^k = I_{\hat{\boldsymbol{w}}^k}\hat{\boldsymbol{w}}^k + \mathbf{0}N(\hat{\boldsymbol{w}}^k) \tag{80}$$

where $I_{\hat{\boldsymbol{w}}^k}$ is identity matrix with dimension $d_{\hat{\boldsymbol{w}}^k} \times d_{\hat{\boldsymbol{w}}^k}$. Now, in this new basis when we project the $\boldsymbol{z}^{k-1}$ to the null space of $\hat{\boldsymbol{w}}^k$ we have:

$$\begin{aligned}
\boldsymbol{z}^k &= P_{N(\hat{\boldsymbol{w}}^k)}\boldsymbol{z}^{k-1} && (81)\\
&= \left(I - \begin{bmatrix} I_{\hat{\boldsymbol{w}}^k} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} I_{\hat{\boldsymbol{w}}^k} & \mathbf{0} \end{bmatrix}\right) \begin{bmatrix} \boldsymbol{z}_{\hat{\boldsymbol{w}}^k} \\ \boldsymbol{z}_{N(\hat{\boldsymbol{w}}^k)} \end{bmatrix} && (82)\\
&= \begin{bmatrix} \boldsymbol{z}_{\hat{\boldsymbol{w}}^k} \\ \boldsymbol{z}_{N(\hat{\boldsymbol{w}}^k)} \end{bmatrix} - \begin{bmatrix} \boldsymbol{z}_{\hat{\boldsymbol{w}}^k} \\ \mathbf{0} \end{bmatrix} && (83)\\
&= \begin{bmatrix} \mathbf{0} \\ \boldsymbol{z}_{N(\hat{\boldsymbol{w}}^k)} \end{bmatrix} && (84)
\end{aligned}$$

Thus the norm of $\|\boldsymbol{z}^k\| = \sqrt{\|\boldsymbol{z}_{N(\hat{\boldsymbol{w}}^p)}\| + 0}$ is less that $\|\boldsymbol{z}^{k-1}\| = \sqrt{\|\boldsymbol{z}_{\hat{\boldsymbol{w}}^k}\|^2 + \|\boldsymbol{z}_{N(\hat{\boldsymbol{w}}^k)}\|^2}$ if $\boldsymbol{z}_{\hat{\boldsymbol{w}}^k} \neq 0$. Next we will show that $\boldsymbol{z}_{\hat{\boldsymbol{w}}^k}$ cannot be zero. From Eq 84, we observe that at any step "k" INLP removes the part of the representation from $\boldsymbol{z}^{k-1}$ which is in the direction of $\hat{\boldsymbol{w}}^k$ i.e. $\boldsymbol{z}_{\hat{\boldsymbol{w}}^k}$. Consequently, a sequence of removal steps with parameters $\hat{\boldsymbol{w}}^1, \ldots, \hat{\boldsymbol{w}}^{k-1}$ will remove the part of $\boldsymbol{z}$ which lies in the span($\hat{\boldsymbol{w}}^1, \ldots, \hat{\boldsymbol{w}}^{k-1}$). Thus $\boldsymbol{z}_{\hat{\boldsymbol{w}}^k} = \mathbf{0}$ if $\hat{\boldsymbol{w}}^k$ lies in the span of parameters of previous classifier i.e span($\hat{\boldsymbol{w}}^1, \ldots, \hat{\boldsymbol{w}}^{k-1}$) which violates the assumption 2) in Lemma D.2, thus completing our proof.

□

Given parameters of current step concept-probing classifier doesn't lie in the span of previous probing classifier's parameter, the norm of the latent representation is decreasing in every step. Thus given large number of INLP projection iteration $\|\boldsymbol{z}^{i(\infty)}\| \to \mathbf{0}$ thus completing the proof of the second statement. The proof of Lemma D.1 and Lemma D.2 is given below. □

Remark. The following lemma from (Ravfogel et al., 2020) tells us some of the sufficient condition when the parameters of probing classifier at current iteration will not be same as previous step:

Lemma D.3 (Lemma A.1 from (Ravfogel et al., 2020))**.** If the concept-probing classifier is being trained using SGD (stochastic gradient descent) and the loss function is convex then parameter of probing classifier at step k i.e $\boldsymbol{w}^k$ is orthogonal to parameter at step $k-1$ i.e $\boldsymbol{w}^{k-1}$.

We conjecture that Lemma D.3 will be true for any loss since after $k-1$ steps of projection, the component of $\boldsymbol{z}$ in the direction of span($\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{k-1}$) will be removed. Hence the concept-probing classifier should be orthogonal to span($\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{k-1}$) in order to have non-random guess accuracy on probing task.

E. Adversarial Removal: Setup and Proof

E.1. Adversarial Setup

As described in Sec 2.3, let $h : \boldsymbol{X} \rightarrow \boldsymbol{z}$ be an encoder mapping the input $\boldsymbol{x}^i$ to latent representation $\boldsymbol{z}^i$. The main task classifier $c_m : \boldsymbol{Z} \rightarrow Y_m$ is applied to predict the main task label y_m^i from the input latent representation $\boldsymbol{z}^i$ for every input $\boldsymbol{x}^i$.

As described in Sec 2.3, the goal of adversarial-removal method, henceforth referred as AR, is to remove any undesired/sensitive/spurious concept from the latent representation $\boldsymbol{Z}$. Once the concept is removed from latent representation any (main-task) classifier using the latent representation $\boldsymbol{Z}$ will not be able to use them (Ganin & Lempitsky, 2015; Xie et al., 2017; Elazar & Goldberg, 2018). These methods train jointly trains the main-task classifier $c_m(\boldsymbol{z})$ and the probing classifier $c_p : \boldsymbol{Z} \rightarrow Y_p$ which is adversarially trained to predict the concept label y_p^i from latent representation $\boldsymbol{z}^i$. Hence, AR methods optimize the following two objectives simultaneously:

$$arg \min_{c_p} L(c_p(h(\boldsymbol{z})), y_p) \tag{85}$$

$$arg \min_{h, c_m, c_p} \Big\{ L(c_m(h(\boldsymbol{z}), y_m) - L(c_p(h(\boldsymbol{z}), y_p) \Big\} \tag{86}$$

Here $L(\cdot)$ is the loss function which estimates the error given the ground truth y_m/y_p and corresponding prediction $c_m(\boldsymbol{z})/c_p(\boldsymbol{z})$. A valid equilibrium point of the above min-max game (and the desired solution) is an encoder h such that it removes all the features from latent space useful for prediction of y_p while keeping intact other features causally-derived from the main-task prediction. In practice, the optimization to above min-max game is performed using gradient-reversal (GRL) layer ((Ganin & Lempitsky, 2015)). It introduces an additional layer g_λ between the latent representation $h(\boldsymbol{z})$ and the adversarial classifier $c_p(\boldsymbol{z})$. The g_λ layer acts as an identity layer (i.e., has no effect) during the forward pass but scales the gradient by $(-\lambda)$ when back-propagating it during the backward pass. Thus resulting combined objective is:

$$arg \min_{h, c_m, c_p} \Big\{ L(c_m(h(\boldsymbol{z})), y_m) + L(c_p(g_\lambda(h(\boldsymbol{z}))), y_p) \Big\} \tag{87}$$

E.2. Adversarial Proof

Theorem 2.3. Let the latent representation $\boldsymbol{Z}$ satisfy Assm 2.1, $h_2(\boldsymbol{z})$ be a linear transformation s.t. $h_2 : \boldsymbol{Z} \rightarrow \boldsymbol{\zeta}$, main classifier be $c_m(\boldsymbol{\zeta}) = \boldsymbol{w}_{c_m} \cdot \boldsymbol{\zeta}$ and the adversarial classifier be $c_p(\boldsymbol{\zeta}) = \boldsymbol{w}_{c_p} \cdot \boldsymbol{\zeta}$. Let all the assumptions of generalized Lemma 2.1 (Lemma C.5 in Suppl C.4) be satisfied for main-classifier $c_m(\cdot)$ when using $\boldsymbol{Z}$ directly as input and Assm 2.2 be satisfied on $\boldsymbol{Z}$ w.r.t. adversarial-task. Let $h_2^*(\boldsymbol{z})$ be the desired encoder which is successful in removing $\boldsymbol{z}_p$ from $\boldsymbol{\zeta}$. Then there exists an undesired/incorrect encoder $h_2(\boldsymbol{z})$ such that $\boldsymbol{\zeta} = h_2(\boldsymbol{z}) = f(\boldsymbol{z}_m, \boldsymbol{z}_p)$ for some function "f" and the main classifier using it $c_m(h_2(\boldsymbol{z}))$ has bigger margin than $c_m(h_2^*(\boldsymbol{z}))$ and has,

1. $Acc(c_p(h_2(\boldsymbol{z})), y_p) = Acc(c_p(h_2^*(\boldsymbol{z})), y_p)$ when adversarial probing classifier $c_p(\cdot)$ is trained using any learning objective. Thus the undesired encoder $h_2(\boldsymbol{z})$ is indistinguishable from desired encoder $h_2^*(\boldsymbol{x})$ in terms of adversarial task prediction accuracy but better for main-prediction task in terms of max-margin objective.

2. $L(c_p(h_2(\boldsymbol{z})), y_p) > L(c_p(h_2^*(\boldsymbol{z})), y_p)$ under Assm 2.2 and Assm 2.4 where L is either max-margin or cross-entropy loss used for training $c_p(\cdot)$. Thus undesired encoder $h_2(\boldsymbol{z})$ is preferable over desired encoder $h_2^*(\boldsymbol{z})$ for both main and adversarial task objective.

Proof of Theorem 2.3. Let the main classifier be of form $c_m(\boldsymbol{\zeta}) = \boldsymbol{w}_{c_m} \cdot \boldsymbol{\zeta}$ where $\boldsymbol{w}_{c_m}$ and $\boldsymbol{\zeta}$ are $d_{\boldsymbol{\zeta}}$ dimensional vectors. Since both parameters $\boldsymbol{w}_{c_m}$ and $\boldsymbol{\zeta}$ are learnable, for ease of exposition we constrain $\boldsymbol{w}_{c_m}$ to be $[1, 0, \ldots, 0]$. This constraint on $\boldsymbol{w}_{c_m}$ is w.l.o.g. since $\boldsymbol{w}_{c_m}$ uses the latent representation $\boldsymbol{\zeta}$ by projecting it into one specific direction of $\boldsymbol{\zeta}$ encoded by $h_2(\boldsymbol{z})$. We constrain that direction to be the first dimension of $\boldsymbol{\zeta}$. Since the encoder is trainable it could learn to encode that information in the first dimension of $\boldsymbol{\zeta}$. Thus, effectively a single dimension of the representation $\boldsymbol{\zeta}$ encodes the main-task information. Thus the main classifier is effectively of form $c_m(\boldsymbol{\zeta}) = \boldsymbol{w}_{c_m}^0 \times \boldsymbol{\zeta}^0 = 1 \times \boldsymbol{\zeta}^0$ where $\boldsymbol{w}_{c_m}^0$ and $\boldsymbol{\zeta}^0$ are the first element of $\boldsymbol{w}_{c_m}$ and $\boldsymbol{\zeta}$ respectively and $\boldsymbol{w}_{c_m}^0 = 1$. We can now write the goal of the adversarial method as removing the information of $\boldsymbol{z}_p$ from $\boldsymbol{\zeta}^0$, because the other dimensions are not used by the main classifier . The the adversarial classifier can be written effectively as $c_p(\boldsymbol{\zeta}) = \beta \times \boldsymbol{\zeta}^0$ where $\beta \in \mathbb{R}$ is a trainable parameter. Since both the main and adversarial

classifier are using only $\boldsymbol{\zeta}^0$, the second encoder can be simplified as $\zeta := \boldsymbol{\zeta}^0 := h_2(\boldsymbol{z}) = \boldsymbol{w}_m \cdot \boldsymbol{z}_m + \boldsymbol{w}_p \cdot \boldsymbol{z}_p$ where $\zeta \in \mathbb{R}$ and $\boldsymbol{w}_m$ and $\boldsymbol{w}_p$ are the weights that determine the first dimension of $\boldsymbol{\zeta}$. . Also, let the desired "correct" second encoder which is successful in removing the spurious-concept $\boldsymbol{z}_p$ from ζ be $h_2^*(\boldsymbol{z}) = \boldsymbol{w}_m^* \cdot \boldsymbol{z}_m$.

1. First claim. The ideal main-classifier with desired encoder can be written as, $c_m(h_2^*(\boldsymbol{z})) = 1 \times h_2^*(\boldsymbol{z}) = \boldsymbol{w}_m^* \cdot \boldsymbol{z}_m$. Therefore, it can be trained using same *MM-Denominator* formulation of max-margin objective and would satisfy the constraint in Eq 3:

$$m(c_m(h_2^*(\boldsymbol{z}^i))) = m(h_2^*(\boldsymbol{z}^i)) = y_m^i \cdot h_2^*(\boldsymbol{z}^i) \geq 1 \tag{88}$$

for all the points $\boldsymbol{x}^i$ with latent representation $\boldsymbol{z}^i$ and $m(.)$ is the numerator of the distance of point from the decision boundary of classifier (Eq 1).

However, the main-task classifier which does not use the desired encoder is of the form, $c_m(\zeta) = 1 \times h_2(\boldsymbol{z}) = \boldsymbol{w}_m \cdot \boldsymbol{z}_m + \boldsymbol{w}_p \cdot \boldsymbol{z}_p$. Since this main-task classifier is also trained using max-margin objective by *MM-Denominator* formulation, it would satisfy the constraint in Eq 3:

$$m(c_m(h_2(\boldsymbol{z}^i))) = m(h_2(\boldsymbol{z}^i)) = y_m^i \cdot h_2(\boldsymbol{z}^i) \geq 1 \tag{89}$$

Since all assumptions of Lemma C.5 are satisfied, we use the lemma to conclude that the main-task classifier trained using max-margin objective will be *spurious-using* i.e $\boldsymbol{w}_m \neq \mathbf{0}$ and $\boldsymbol{w}_p \neq \mathbf{0}$ (Def C.2). Hence there exists an undesired/incorrect second encoder $h_2(\boldsymbol{z})$ such that the main classifier $c_m(h_2(\boldsymbol{z}))$ has bigger margin than $c_m(h_2^*(\boldsymbol{z}))$.

Given that there exists a main task classifier with an undesired encoder that has a higher accuracy than the one with the desired encoder, our next goal is to show that the accuracy of the adversarial classifier remains the same irrespective of whether the desired or undesired encoder $h_2(\boldsymbol{z})$ is used. The accuracy of the adversarial classifier $c_p(\zeta) = \beta \times \zeta$, using the desired/correct encoder $\zeta = h_2^*(\boldsymbol{z})$ is given by:

$$Accuracy(c_p(h_2^*(\boldsymbol{z}))) = \frac{\sum_{i=1}^{N} \mathbf{1}\Big(sign\big(\beta \cdot h_2^*(\boldsymbol{z}^i)\big) == y_p^i\Big)}{N} \tag{90}$$

where $\mathbf{1}(\cdot)$ is an indicator function which takes the value 1 if the argument is true otherwise 0, and $sign(\gamma) = +1$ if $\gamma \geq 0$ and -1 otherwise. Combining Equations 88 and Eq 89, since $y_m^i \in \{-1, 1\}$, we see that whenever $h_2(\boldsymbol{z}^i) > 1$ we also have $h_2^*(\boldsymbol{z}^i) > 1$ and similarly whenever $h_2(\boldsymbol{z}^i) < -1$, we have $h_2^*(\boldsymbol{z}^i) < -1$. Thus,

$$h_2(\boldsymbol{z}) \cdot h_2^*(\boldsymbol{z}) > 0 \tag{91}$$

From Eq 91, $h_2^*(\boldsymbol{z}^i)$ and $h_2(\boldsymbol{z}^i)$ has the same sign for every input $\boldsymbol{z}^i$. Thus we can replace $h_2^*(\boldsymbol{z}^i)$ with $h_2(\boldsymbol{z}^i)$ in the above equation and we have:

$$\begin{aligned} Accuracy(c_p(h_2^*(\boldsymbol{z}))) &= \frac{\sum_{i=1}^{N} \mathbf{1}\Big(sign\big(\beta \cdot h_2^*(\boldsymbol{z}^i)\big) == y_p^i\Big)}{N} \\ &= \frac{\sum_{i=1}^{N} \mathbf{1}\Big(sign\big(\beta \cdot h_2(\boldsymbol{z}^i)\big) == y_p^i\Big)}{N} \\ &= Accuracy(c_p(h_2(\boldsymbol{z}))) \end{aligned}$$

thus completing the first part our proof.

2. Second claim. In the first part of proof we have seen that from Lemma C.5 there exist an incorrect second-encoder $h_2(\boldsymbol{z})$ such that margin of main-task classifier $c_m(h_2(\boldsymbol{z}))$ is larger than $c_m(h_2^*(\boldsymbol{z}))$. Next, we show that adversarial classifier $c_p(\beta \cdot \zeta)$ when using incorrect/undesired second-encoder $\zeta = h_2(\boldsymbol{z})$ for predicting concept-label y_p will have a higher prediction loss as compared to the case when using correct/desired second-encoder $\zeta = h_2^*(\boldsymbol{z})$. Under such a scenario, the incorrect encoder will be preferred by both the main and the adversarial task objective over the correct one. Note that the adversarial-probing classifier prefers a *higher* prediction loss since this will imply that the latent representation generated by the second encoder is less predictive of the concept-causal feature we are trying to remove.

Construction of the undesired/incorrect encoder. Let any margin point of main-task classifier using the correct encoder be denoted by $\boldsymbol{z}^{\mu,i}$, where μ, denotes the points is margin-point of main-task classifier. From Assm C.2 (in Lemma C.5), we know there exist some unit vector $\hat{\boldsymbol{\epsilon}}_p \in \mathbb{R}^{d_p}$ such that spurious concept-feature ($\boldsymbol{z}_p$) of the margin points of the main-task classifier ($\boldsymbol{z}^{\mu,i}$) using the correct encoder is linearly separable wrt main-task label such that:

$$y_m^i\big(\hat{\boldsymbol{\epsilon}}_p \cdot \boldsymbol{z}_p^{\mu,i}\big) > 0 \;\; \forall(\boldsymbol{z}^{\mu,i}, y_m^i) \tag{92}$$

and from Lemma C.5 there exists an incorrect encoder $h_2^\alpha(\boldsymbol{z})$ such that the main task classifier $c_m(h_2(\boldsymbol{z}))$ has bigger margin than $c_m(h_2^*(\boldsymbol{z}))$ when $\alpha \in (\alpha_{lb}, 1)$, where α_{lb} is given by Eq 30 in Lemma C.5:

$$h_2^\alpha(\boldsymbol{z}) = \alpha\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m\big) + \sqrt{1-\alpha^2}\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{\epsilon}}_p \cdot \boldsymbol{z}_p\big) \tag{93}$$

where $\boldsymbol{w}_m^*$ is the parameter of the correct/desired encoder $h_2^*(\boldsymbol{x}) = \boldsymbol{w}_m^* \cdot \boldsymbol{z}_m$. We now use the assumptions in the theorem statement for the second claim. From Assm 2.2, we have a fully predictive concept-feature $\boldsymbol{z}_p$ for prediction of adversarial label y_p such that for some unit vector $\hat{\boldsymbol{w}}_p \in \mathbb{R}^{d_p}$ we have:

$$y_p^i\big(\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i\big) > 0 \;\; \forall(\boldsymbol{z}^i, y_p^i) \tag{94}$$

Now from Assm 2.4, we have $y_p^i = y_m^i$ for every margin point of the desired/correct main-task classifier using the desired/correct encoder $h_2^*(\boldsymbol{z})$. Thus we can assign $\hat{\boldsymbol{\epsilon}}_t := \hat{\boldsymbol{w}}_t$ which satisfies the inequality in Eq 92. Thus our *incorrect* encoder $h_2^\alpha(\boldsymbol{z})$ take the following form:

$$h_2^\alpha(\boldsymbol{z}) = \alpha\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m\big) + \sqrt{1-\alpha^2}\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p\big) \tag{95}$$

We will use the following two lemmas to prove that the adversarial prediction loss will be bigger for the incorrect/undesired encoder than the correct/desired one.

Next the following lemmas will show that w.r.t. max-margin and cross-entropy objective, the incorrect/undesired encoder's loss is greater than the correct/desired encoder's loss. We state two simpler lemmas first and then define the two main lemmas.

Lemma E.1 (Negative adv-labels)**.** For all the points with negative adversarial label i.e. $y_p^i = -1$, we have $h_2^\alpha(\boldsymbol{z}^i) < h_2^*(\boldsymbol{z}^i)$

Lemma E.2 (Positive adv-labels)**.** For all the points with positive adversarial labels i.e $y_p^i = +1$ we can choose an $\alpha \in (\alpha_{lb}, 1)$ such that $h_2^\alpha(\boldsymbol{z}^i) > h_2^*(\boldsymbol{z}^i)$.

Definition E.1 (Boundary-Distance Based Loss)**.** A loss function L belongs to *boundary-distance-based* loss family, if the prediction loss increases as the distance from the boundary decreases. The distance from the boundary of a classifier "c" for a point $\boldsymbol{z}^i$ is calculated by $d_c(\boldsymbol{z}^i) := \mathcal{M}_c(\boldsymbol{z}^i)$ (Eq 1). For mis-classifed points the distance from decision boundary is negative and for correctly classified points the distance from decision boundary is positive.

Lemma E.3. Given Lemma E.1 and Lemma E.2 holds, if the loss function L is *boundary-distance-based* (Def E.1), then $L(c_p(h_2(\boldsymbol{z}^i)), y_p^i) > L(c_p(h_2^*(\boldsymbol{z}^i)), y_p^i)$ for every example $(\boldsymbol{z}^i, y_p^i)$. Cross-Entropy and Max-margin loss belongs to *boundary-distance-based* loss.

Lemma E.4. Max-margin and Cross-Entropy Loss belongs to the *boundary-distance-based* loss family (Def E.1) when the norm of parameters of classifier is fixed.

Lemma E.3 and Lemma E.4 state that there exists an incorrect/undesired encoder $h_2^\alpha(\boldsymbol{z})$ which has the same norm as correct/desired encoder $h_2^*(\boldsymbol{z})$, such that $L(c_p(h_2^\alpha(\boldsymbol{z}^i)), y_p^i) > L(c_p(h_2^*(\boldsymbol{z}^i)), y_p^i)$ where L is max-margin or cross-entropy loss for all $\boldsymbol{z}^i$. Hence the overall loss which is the sum of the loss for every sample is also greater for adversarial classifier with incorrect encoder $h_2^\alpha(\boldsymbol{z})$. Thus the incorrect encoder is preferred over the correct encoder for both the main task (bigger margin) and the adversarial task objective (bigger prediction loss).

□

We now prove all the required lemmas stated in the above proof.

Proof of Lemma E.1. From Eq 94, for the points having adversarial label $y_p^i = -1$, we have $\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i < 0$. Now, since

$0 < \alpha_{lb} < \alpha < 1$ we have $\sqrt{1-\alpha^2}\|\boldsymbol{w}_m^*\| > 0$, and can write,

$$\alpha\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i\big) < \|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i\big) \tag{96}$$

$$\alpha\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i\big) + \sqrt{1-\alpha^2}\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i\big) < \|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i\big) \tag{97}$$

$$h_2^\alpha(\boldsymbol{z}^i) < h_2^*(\boldsymbol{z}^i) \tag{98}$$

since $\sqrt{1-\alpha^2}\|\boldsymbol{w}_m^*\|\big(\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i\big) < 0$ for points with $y_p^i = -1$, thus completing our proof. □

Proof of Lemma E.2. We have to show that there exist an $\alpha \in (\alpha_{lb}, 1)$ such that $h_2^\alpha(\boldsymbol{z}^i) > h_2^*(\boldsymbol{z}^i)$ for every $\boldsymbol{z}^i$ in the dataset. For $h_2^\alpha(\boldsymbol{z}^i) > h_2^*(\boldsymbol{z}^i)$ we need:

$$\alpha\cancel{\|\boldsymbol{w}_m^*\|}\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i\big) + \sqrt{1-\alpha^2}\cancel{\|\boldsymbol{w}_m^*\|}\big(\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i\big) > \cancel{\|\boldsymbol{w}_m^*\|}\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i\big) \tag{99}$$

$$\sqrt{1-\alpha^2}\big(\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i\big) > (1-\alpha)\big(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i\big) \tag{100}$$

$$\frac{\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i}{\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i} > \sqrt{\frac{\cancel{(1-\alpha)^2}}{\cancel{(1-\alpha)}(1+\alpha)}} \tag{101}$$

$$\gamma^2(\boldsymbol{z}^i) > \frac{1-\alpha}{1+\alpha} \tag{102}$$

$$\gamma^2(\boldsymbol{z}^i) + \alpha\gamma^2(\boldsymbol{z}^i) > 1-\alpha \tag{103}$$

$$\alpha(1+\gamma^2(\boldsymbol{z}^i)) > 1-\gamma^2(\boldsymbol{z}^i) \tag{104}$$

$$\alpha > \frac{1-\gamma^2(\boldsymbol{z}^i)}{1+\gamma^2(\boldsymbol{z}^i)} \tag{105}$$

since $\alpha \neq 1$. We defined $\gamma(\boldsymbol{z}^i) = \frac{\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i}{\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i}$, where the denominator $(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i) \neq 0$ since $y_m^i(\hat{\boldsymbol{w}}_m^* \cdot \boldsymbol{z}_m^i) \geq 1$ and $y_m^i \in \{-1, 1\}$ (Eq 88). Also, from Eq 94 we have $(\hat{\boldsymbol{w}}_p \cdot \boldsymbol{z}_p^i) \neq 0$, thus $\gamma^2(\boldsymbol{z}^i) > 0$, hence we have $0 < \frac{1-\gamma^2(\boldsymbol{z}^i)}{1+\gamma^2(\boldsymbol{z}^i)} < 1$. Hence we can choose an α to construct $h_2(\boldsymbol{z}))$ such that,

$$\alpha \in \left(\max_i \left\{\alpha_{lb}, \frac{1-\gamma^2(\boldsymbol{z}^i)}{1+\gamma^2(\boldsymbol{z}^i)}\right\}, 1\right). \tag{106}$$

Then we have **1)** $h_2^\alpha(\boldsymbol{z}^i) > h_2^*(\boldsymbol{z}^i)$ for all $\boldsymbol{z}^i$ and **2)** margin of main classifier using incorrect/undesired encoder $h_2^\alpha(\boldsymbol{z})$ is bigger than correct/desired encoder $(h_2^*(\boldsymbol{z}))$, thus proving our lemma. □

Proof of Lemma E.3. From Lemma E.2 we have $h_2^\alpha(\boldsymbol{z}) > h_2^*(\boldsymbol{z})$ for all the points which have adversarial label $y_p^i = +1$ and the value of α given by Eq 106. Since, β is a trainable parameter, the adversarial classifier could choose any value for it which will help it increase the adversarial loss. Let's set trainable parameter $\beta = (-1)$ for the adversarial classifier $c_p(h_2^\alpha(\boldsymbol{z})) = \beta \cdot h_2^\alpha(\boldsymbol{z}) = -h_2^\alpha(\boldsymbol{z})$ and $c_p(h_2^*(\boldsymbol{z})) = \beta \cdot h_2^*(\boldsymbol{z}) = -h_2^*(\boldsymbol{z})$. Since by construction the norm of parameters of both $h_2^\alpha(\boldsymbol{z})$ and $h_2^*(\boldsymbol{z}) = \boldsymbol{w}_m^*$ (Eq 93) , we have:

$$h_2^\alpha(\boldsymbol{z}^i) > h_2^*(\boldsymbol{z}^i) \tag{107}$$

$$(-1) \cdot h_2^\alpha(\boldsymbol{z}^i) < (-1) \cdot h_2^*(\boldsymbol{z}^i) \tag{108}$$

$$c_p(h_2^\alpha(\boldsymbol{z}^i)) < c_p(h_2^*(\boldsymbol{z}^i)) \tag{109}$$

$$\frac{y_p^i \cdot c_p(h_2^\alpha(\boldsymbol{z}^i))}{\|\boldsymbol{w}_m^*\|} < \frac{y_p^i \cdot c_p(h_2^*(\boldsymbol{z}^i))}{\|\boldsymbol{w}_m^*\|} \tag{110}$$

$$\mathcal{M}_{c_p(h_2^\alpha(\boldsymbol{z}))}(\boldsymbol{z}^i) < \mathcal{M}_{c_p(h_2^*(\boldsymbol{z}))}(\boldsymbol{z}^i) \tag{111}$$

for all the points with $y_p^i = +1$. Similarly for the points with adversarial label $y_p^i = -1$, from Lemma E.1 we have $h_2^\alpha(\boldsymbol{z}) < h_2^*(\boldsymbol{z})$ and:

$$h_2^\alpha(\boldsymbol{z}^i) < h_2^*(\boldsymbol{z}^i) \tag{112}$$

$$(-1) \cdot h_2^\alpha(\boldsymbol{z}^i) > (-1) \cdot h_2^*(\boldsymbol{z}^i) \tag{113}$$

$$c_p(h_2^\alpha(\boldsymbol{z}^i)) > c_p(h_2^*(\boldsymbol{z}^i)) \tag{114}$$

$$\frac{y_p^i \cdot c_p(h_2^\alpha(\boldsymbol{z}^i))}{\|\boldsymbol{w}_m^*\|} < \frac{y_p^i \cdot c_p(h_2^*(\boldsymbol{z}^i))}{\|\boldsymbol{w}_m^*\|} \tag{115}$$

$$\mathcal{M}_{c_p(h_2^\alpha(\boldsymbol{z}))}(\boldsymbol{z}^i) < \mathcal{M}_{c_p(h_2^*(\boldsymbol{z}))}(\boldsymbol{z}^i) \tag{116}$$

for all points with $y_p^i = -1$. In Def E.1, we have defined the distance of an input from decision boundary of classifier "c" as $d_c(\boldsymbol{z}^i) \coloneqq \mathcal{M}_c(\boldsymbol{z}^i)$. Hence from Eq 111 and 116 we have

$$d_{c_p(h_2^\alpha(\boldsymbol{z}))}(\boldsymbol{z}^i) < d_{c_p(h_2^*(\boldsymbol{z}))}(\boldsymbol{z}^i) \tag{117}$$

for all $\boldsymbol{z}^i$. So, the distance of all the latent-points from margin is smallar when adversarial classifier c_p uses incorrect/undesired encoder $h_2^\alpha(\boldsymbol{z})$ than correct/desired $h_2^*(\boldsymbol{z})$. If the loss function "L" is *boundary-distance-based* loss (Def E.1), then $L(c_{h_2^\alpha(\boldsymbol{z}^i)}, y_p^i) > L(c_{h_2^*(\boldsymbol{z}^i)}, y_p^i)$ for every $\boldsymbol{z}^i, y_p^i$, thus completing our proof.

□

Proof of Lemma E.4. Max-Margin: From Eq 2, the max-margin loss (to be minimized) for a classifier "c" taking input $\boldsymbol{z}^i$ is defined as:

$$L(\text{max-margin}) = (-1)\Big\{\min_i \mathcal{M}_c(\boldsymbol{z}^i)\Big\} \tag{118}$$

where $\|\boldsymbol{w}_c\|$ are the parameters of the classifier "c". If the distance of all the point from decision boundary of classifier "c" i.e. $d_c(z^i) \coloneqq \mathcal{M}_c(\boldsymbol{z}^i)$ is decreasing, then $\min_i \mathcal{M}_c(\boldsymbol{z}^i)$ is decreasing hence $(-1) \cdot \big\{\min_i \mathcal{M}_c(\boldsymbol{z}^i)\big\}$ is increasing. Max-margin loss in Eq 118 is increasing as the distance of points from boundary of classifier is decreasing. Thus from (Def E.1) $L(\text{max-margin})$ belongs to *boundary-distance-based* family of losses.

Cross-Entropy: Next for the binary-cross-entropy loss for a linear classifier $c(\boldsymbol{z}^i) = \sigma(\boldsymbol{w}_c \cdot \boldsymbol{z}^i)$ taking input $\boldsymbol{z}^i$ is defined as

$$L(\text{x-entropy}) = -\log(c(\boldsymbol{z}^i)) \qquad \text{when} \quad y^i = 1 \tag{119}$$

$$= -\log(1 - c(\boldsymbol{z}^i)) \qquad \text{when} \quad y^i = -1 \tag{120}$$

where $\sigma(\boldsymbol{x}) = \frac{1}{1+\exp^{-\boldsymbol{x}}}$ is the sigmoid function. In E.1 the distance of a point from the decision boundary is defined as:

$$d_c(\boldsymbol{z}^i) \coloneqq \mathcal{M}_c(\boldsymbol{z}^i) = \frac{y^i \cdot c(\boldsymbol{z}^i)}{\|\boldsymbol{w}_c\|} \tag{121}$$

When $y^i = 1$, keeping $\|\boldsymbol{w}_c\|$ as fixed, if the distance of the point decreases the from Eq 121, $c(\boldsymbol{z}^i)$ decreases. Hence the cross entropy loss given by:

$$L(\text{x-entropy}) = -y^i \cdot \log(c(\boldsymbol{z}^i)) = -\log(c(\boldsymbol{z}^i)) \tag{122}$$

increases since $-\log(c(\boldsymbol{z}^i))$ is a decreasing function. Similarly for the points with $y^i = -1$, keeping $\|\boldsymbol{w}_c\|$ fixed, if the distance of the point decreases, from Eq 121, $c(\boldsymbol{z}^i)$ increases. Then the cross-entropy loss given by:

$$L(\text{x-entropy}) = -(1 - y^i) \cdot \log(1 - c(\boldsymbol{z}^i)) = -\log(1 - c(\boldsymbol{z}^i)) \tag{123}$$

increases, since $-\log(1 - c(\boldsymbol{z}^i))$ is an increasing function. Cross-entropy loss with fixed parameter norm is increasing as the distance of points from the boundary of classifier is increasing. Thus from (Def E.1) $L(\text{max-margin})$ belongs to *boundary-distance-based* family of losses. □

F. Extension to real-world data: A metric for quantifying degree Of spuriousness

Our theoretical analysis shows that probing-based removal methods fail to make the main task classifier invariant to unwanted concept. However, to verify whether the final classifier is using the concept or not, the theorem statements require knowledge of the concept's features $\boldsymbol{z}_p$. For practical usage, we propose a metric that quantifies the degree of failure or *spuriousness* for both the main and probing classifier. For simplicity, we define it assuming that both main task and concept are binary.

Let $\mathcal{D}_{m,p}$ be the dataset where for every input $\boldsymbol{x}^i$ we have both the main task label y_m and the concept label y_p. We define 2×2 groups, one for each combination of (y_m, y_p). Without loss of generality, assume that the main-task label $y_m = 1$ is spuriously correlated with concept label $y_p = 1$ and similarly $y_m = 0$ is correlated with $y_p = 0$. Thus, $(y_m = 1, y_p = 1)$ and $(y_m = 0, y_p = 0)$ are the majority group S_{maj} while groups $(y_m = 1, y_p = 0)$ and $(y_m = 0, y_p = 1)$ make up the minority group (S_{min}). We expect the main classifier to exploit this correlation and hence perform badly on S_{min} where the correlation breaks. Following (Sagawa et al., 2020a), we posit that minority group accuracy i.e $Acc(S_{min})$ can be a good metric to evaluate the degree of *spuriousness*. We bound the metric by comparing it with the accuracy on S_{min} of a "clean" classifier that does not use the concept features.

Definition 2.2. Given a dataset, $\mathcal{D}_{m,p} = S_{min} \cup S_{maj}$ containing binary task label and binary concept, let $Acc^{c_m}(S_{min})$ be the accuracy of the given main-task classifier (c_m) on the minority group S_{min} and $Acc^*(S_{min})$ be the accuracy on the minority group of a *clean* classifier that does not use the unwanted spurious concept. We define spuriousness score of "c_m" as: $\psi(c_m) = \left|1 - \frac{Acc^{c_m}(S_{min})}{Acc^*(S_{min})}\right|$

To estimate $Acc^*(S_{min})$, we subsample the dataset such that y_p takes a single value in the sample and train the main classifier on it, as done by (Ravichander et al., 2021). Thus, probing label y_p no longer is correlated with the main-task label y_m. There can be other ways to estimate it, e.g., by reweighting the data or using the accuracy on S_{maj}. However, we found that the former had high variance and the latter requires an equal-noise assumption such that $Acc(S_{maj}) = Acc(S_{min})$ on a *clean* main-task classifier.

G. Experimental Results

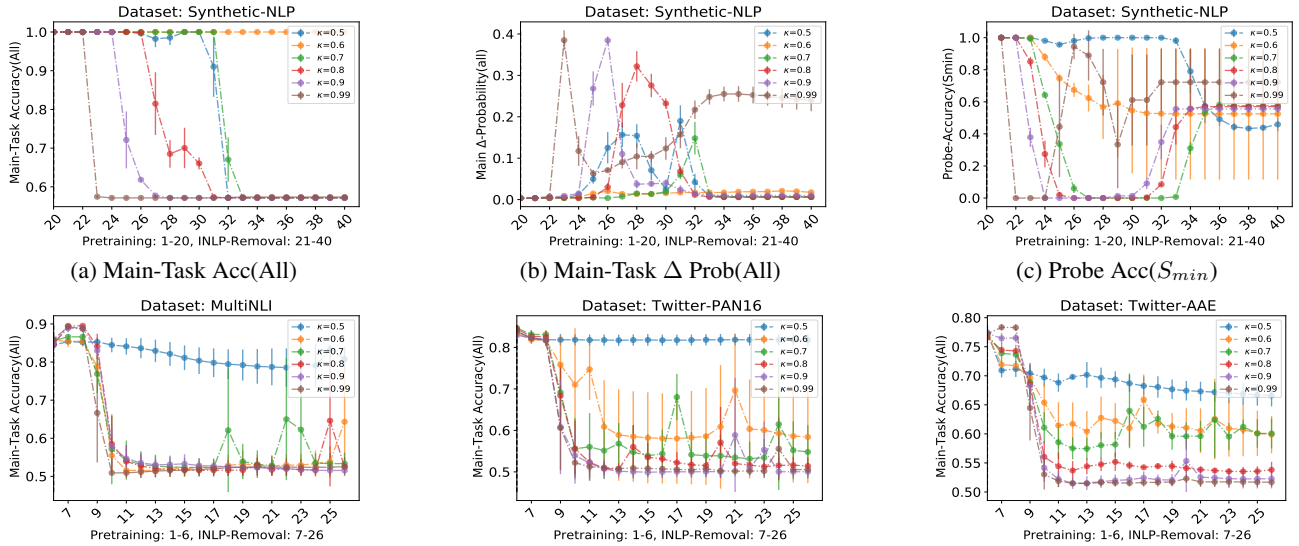
Theorems 2.2 and 2.3 show the failure of concept removal methods under a simplified setup and max-margin loss. But current deep-learning models are not trained using max-margin objective and might not satisfy the required assumptions (Assm 2.1,2.2,2.3,2.4). Thus, we now verify the failure modes on three real-world and one synthetic datasets, without making any restrictive assumptions. We use RoBERTa ((Liu et al., 2019)) as default encoder and fine-tune it over each real-world dataset. For Synthetic-Text dataset we use sum of pre-trained GloVe embeddings ((Pennington et al., 2014)) of words in sentence as default encoder. For more details of experimental setup refer Suppl H.

G.1. Datasets: Main task and spurious/sensitive concept

Real-world data. We use three different real-world datasets: MultiNLI (Williams et al., 2018), Twitter-PAN16 (Rangel et al., 2016) and Twitter-AAE (Blodgett et al., 2016). In MultiNLI dataset, given two sentences—premise and hypothesis–the main task is to predict whether hypothesis *entails*, *contradicts* or is *neutral* with respect to premise. We simplify to a binary task of predicting whether a hypothesis *contradicts* the premise or not. Since negation words like *nobody,no,never* and *nothing* have been reported to be spuriously correlated with with the *contradiction* label (Gururangan et al., 2018b), We create a 'negation' concept denoting the presence of these words. The goal is to remove the negation property from the NLI model's representation space. In Twitter-PAN16 dataset, the main task is to detect whether a tweet mentions another user or not, as in (Elazar & Goldberg, 2018). The dataset contains *gender* label for each tweet, which we consider as the sensitive concept to be removed from the main model's representation. In Twitter-AAE dataset, again following (Elazar & Goldberg, 2018), the main-task is to predict binary sentiment labels from a tweet's text. The tweets are associated with *race* of the author, the sensitive concept that should be removed from the main model's representation.

Synthetic-Text. To understand the reasons for failure, we introduce a Synthetic-Text dataset where it is possible to change the input text based on a change in concept Here we can directly evaluate whether the concept is being used by the main-task classifier by intervening on the concept (adding or removing) and observing the change in prediction scores. The main-task is to predict whether a sentence contains a numbered word (e.g., *one, fifteen, etc.*). We introduce spurious concept (length) by increasing the length of sentences that contain *numbered* words.

Predictive correlation. To assess robustness of removal methods, we create multiple datasets with different *predictive*



(a) Main-Task Acc(All) (b) Main-Task Δ Prob(All) (c) Probe Acc(S_{min})

(d) Main-Task Acc(All) (e) Main-Task Acc(All) (f) Main-Task Acc(All)

Figure 4: **Null-Space Removal Failure**: For discussion see Sec G.2.

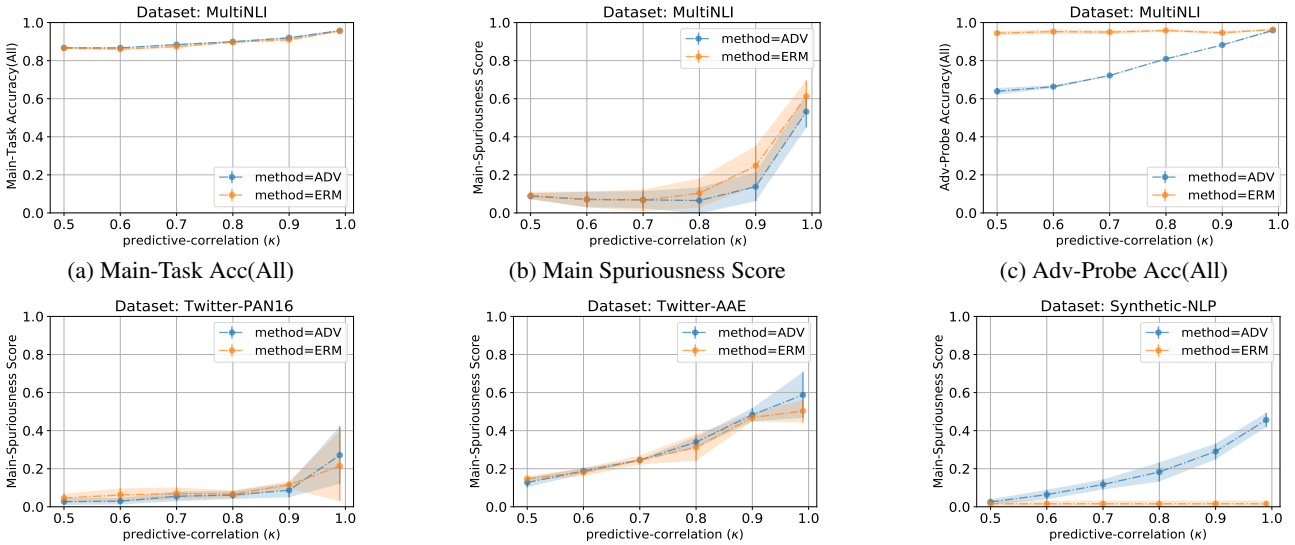
correlation between two labels y_m and y_p. The predictive correlation (κ) measures how informative one variable is for predicting the other, $\kappa = Pr(y_m \cdot y_p > 0) = \frac{\sum_{i=1}^{N} \mathbf{1}[y_m \cdot y_p > 0]}{N}$, where N is the size of dataset and $\mathbf{1}[\cdot]$ is the indicator function which is 1 if the argument is true otherwise 0. Predictive-correlation lies in $\kappa \in [0.5, 1]$ where $\kappa = 0.5$ means no-correlation and $\kappa = 1$ means the attributes are completely correlated. For complete description of all datasets and how we vary the predictive-correlation, refer to Suppl H; and for additional results refer Suppl I.

G.2. Results: Null Space Removal

In general, for any model given as input to INLP, it may be difficult to verify whether the removal method removed the correct features. Hence, we construct a benchmark where the input classifier is *clean*, i.e., it does not use the concept at all. We do so by training on a subset of data with the same value of the property, as in (Ravichander et al., 2021). Since the input classifier does not have any concept features in its representation, we should expect that INLP should not have any effect on the main task classifier.

Eventually all task-relevant features are destroyed. We start with the Synthetic-Text dataset by training a clean classifier on the main-task and input it to INLP for removing the sensitive concept. To keep the conditions favorable for INLP, both the main task and sensitive-concept probing task can achieve 100% accuracy using their causally derived features respectively. In Fig 4a, iteration 21-40 shows individual steps of null-space removal. Since, the given pre-trained classifier was *clean* i.e no using the sensitive features, null-space removal shouldn't have any effect on it. Colored lines show different datasets used by INLP with different level of predictive correlation κ. We observe that for all value of κ, the main-task classifier's accuracy eventually goes to 50% random guess accuracy implying that the main-task related attribute has been removed by the removal process, as predicted by Theorem 2.2. Higher the value of correlation κ, faster the removal of main-task attribute happens. We obtain a similar pattern over the real-world datasets, MultiNLI, Twitter-PAN16 and Twitter-AAE. Fig 4d,4e and 4f show a decrease in the main-task accuracy to random-guess even when the input classifier for each dataset is ensured to be *clean*, thus demonstrating the failure of INLP method.

Early stopping increases the reliance on spurious features. To avoid the full collapse, a stopping criteria of INLP is to stop when the main-task classifier's performance drops (Ravfogel et al., 2020). In Fig 4b we measure the sensitivity of Synthetic-Text main-task classifier w.r.t. to sensitive concept by changing the feature in input sentences corresponding to sensitive concept and measuring the change in main-task classifier prediction probability ΔProb. At lower iterations of INLP, the change in main-task output due to change in sensitive concept's value, ΔProb, is higher than that of the input classifier. For example, for $\kappa = 0.8$, the main-task classifier's performance drops at iteration 27, but it has a high ΔProb $\approx 25\%$ as shown in Fig 4b. Hence it is possible that stopping prematurely will lead us to a classifier which is more unfair or reliant to sensitive concept than it was before, consistent with the first statement in Theorem 2.2 stating that INLP will lead



(a) Main-Task Acc(All) (b) Main Spuriousness Score (c) Adv-Probe Acc(All)

(d) Main Spuriousness Score (e) Main Spuriousness Score (f) Main Spuriousness Score

Figure 5: **Adversarial Removal Failure**: For discussion see Sec G.3.

to mixing of features in latent space.

Failure of causally-inspired probing like Amnesic probing. Amnesic Probing (Elazar et al., 2021) declares that a sensitive concept is being used by the model if, after removal of the concept from from the latent representation using INLP, there is a drop in the main-task performance. But Fig 4a, 4d, 4e and 4f show that even when the input classifier for their corresponding main-task is clean, i.e., does not use the sensitive concept, INLP leads to drop in performance of the main-classifier. Hence, removal-based methods like Amnesic probing will falsely claim that the sensitive concept is being used.

G.3. Results: Adversarial Removal Method

We demonstrate the failure of adversarial-removal method (AR) in removing the spurious/unwanted concept from the main classifier. Here we compare the standard ERM training of the main classifier with AR method over the same number of epochs (20). We follow the training procedure of (Elazar & Goldberg, 2018), and conduct a hyper-parameter sweep on the adversarial training strength to select the value which is most effective in removing the unwanted property. For more details, refer to Suppl H.

Cannot remove the spurious concept fully. For MultiNLI, Fig 5b shows the spuriousness score (Def 2.2) of ERM and AR classifiers as we vary the predictive correlation (κ) between the main-task label and sensitive concept label in the training dataset. While the spuriousness score for classifier trained using AR (blue curve) is lower than that of ERM for all values of κ, it is substantially away from zero. Thus, the AR method fails to completely remove the sensitive/spurious concept completely from the latent representation. Fig 5c shows the possible reason using the sensitive concept probing classifier accuracy for ERM and AR. The probe accuracy after adversarial training doesn't decrease to 50% but stops at accuracy proportional to the predictive correlation κ. This is expected since even if the AR would have been successful in removing the sensitive feature, the main-task feature would still be predictive of sensitive concept label by κ due to the spurious correlation between them. However, the converse is not true: an accuracy of κ does not imply that sensitive concept is fully removed. The results substantiate the first statement of Theorem 2.3: given two representations where one (*desired*) does not have concept features while the other (*undesired*) contains the sensitive concept features, the undesired one may be better for the main task loss even as both may have the same probing accuracy. Fig 5d, 5e and 5f show the spuriousness score of AR in comparison to classifier trained with ERM on Twitter-PAN16, Twitter-AAE and Synthetic-Text datasets respectively. The failure of AR is worse here: there is no significant reduction in spuriousness score for AR in comparison to ERM. In case of Synthetic-Text dataset, ERM has zero spuriousness score but AR have non-zero score. We expand more on this observation in Suppl I.3.

Comparison with previous work. Our experimental results extend the observations from previous work on adversarial removal failure (Elazar & Goldberg, 2018). If post-removal the latent representation used by the main-task classifier is still

predictive of the removed concept, (Elazar & Goldberg, 2018) claimed it as a failure of the removal method. However, this claim may not be correct since a feature could be present in the latent space and yet not used by model (Ravichander et al., 2021).Therefore, we propose the spuriousness score for the main classifier as a metric to verify if a feature is being used. We also confirm the observation from (Elazar & Goldberg, 2018) that adversarial removal can fail even if there is no correlation between concept and task label ($\kappa = 0.5$), and extend their work by showing failure modes on multiple values of κ. For more details on verifying this observation with other metrics and other ablations, see Suppl I.3.

Ablations. In Suppl., we report results on using BERT instead of RoBERTa as the input encoder (Suppl. I.2, I.3), the effect of using different modeling choices like loss-function, regularization, e.t.c. (I.4), and the behavior of probing classifiers when concept is not there in latent space (I.1).

H. Experimental Setup

H.1. Dataset Description

As described in Sec G, we demonstrate the failure of Null-Space Removal (Sec G.2) and Adversarial Removal (Sec G.3) in removing the undesired concept from the latent representation on three real world datasets: MultiNLI (Williams et al., 2018), Twitter-PAN16 (Rangel et al., 2016) and Twitter-AAE (Blodgett et al., 2016); and a synthetic dataset, Synthetic-Text. The detailed generation and evaluation strategies for each dataset are given below.

MultiNLI Dataset. In MultiNLI dataset, given two sentences—premise and hypothesis—the main-task is to predict whether the hypothesis *entails*, *contradicts* or is *neutral* to the premise. As described in Sec G, we simplify it to a binary task of predicting whether a hypothesis *contradicts* the premise. The binary main-task label, $y_m = 1$ when a given hypothesis *contradicts* the premise otherwise it is -1. That is, we relabel the MNLI dataset by assigning label $y_m = 1$ to examples with contradiction labels and $y_m = -1$ to the example with neutral or entailment label. It has been reported that the *contradiction* label is spuriously correlated with the negation words like *nobody,no,never* and *nothing*(Gururangan et al., 2018b). Thus, we created a 'negation' concept denoting the presence of these words in the hypothesis of a given (hypothesis, premise) pair. The concept-label $y_p = 1$ when the *negation*-concept is present in the hypothesis otherwise it is -1.

The standard MultiNLI dataset has approximately 90% of example in training set, 5% as publicly available development set and the rest of 5% in a separate held out validation set accessible through online competition leader-board not accessible to the public. Thus, we create our own train and test split by subsamping $10k$ examples from the initial training set, converting it into binary contradiction vs. non-contradiction labels, labelling the negation-concept label, and splitting them into 80-20 train and test split. For pre-training a clean classifier that does not use the spurious-concept, we create a special training set following the method described in Sec H.2. For evaluating the robustness of both null-space and adversarial removal methods, we create multiple datasets with different *predictive-correlation* as described in Sec H.3 .

Twitter-PAN16 Dataset. In Twitter-PAN16 dataset (Rangel et al., 2016), following (Elazar & Goldberg, 2018), given a tweet, the main task is to predict whether it contain a mention to another user or not. The dataset contains manually annotated binary Gender information (i.e Male or Female) of 436 Twitter user with atleast 1k tweets each. The Gender annotation was done by assessing the name and photograph of the LinkedIn profile of each user (Elazar & Goldberg, 2018). The unclear cases were discarded in this process. We consider "Gender" as a sensitive concept that should not be used for main-task prediction. The dataset contains 160k tweets tweets for training and 10k tweets for test. We merged the full dataset, subsampled 10k examples, and created a 80-20 train and test split. For pre-training a clean classifier, we create a special training set following the method described in Sec H.2. To generate datasets with different predictive correlation, we follow the method from H.3. The dataset is acquired and processed using the code[2] made available by the (Elazar & Goldberg, 2018).

Twitter-AAE Dataset. In Twitter-AAE dataset (Blodgett et al., 2016), again following (Elazar & Goldberg, 2018), the main-task is to predict a binary sentiment (Positive or Negative) from a given tweet. The dataset contain 59.2 million tweets by 2.8 million users. Each tweet is associated with "race" information of the user which is labelled by taking both words in tweet and geo-location of user into account. We consider "race" as sensitive concept which should not be used for the main task of sentiment prediction. Following (Elazar & Goldberg, 2018), we use the AAE (African America English) and SAE

[2]The code for Twitter-PAN16 and Twitter-AAE dataset acquisition is available at: https://github.com/yanaiela/demog-text-removal

(Standard American English) as a proxy for non-Hispanic blacks and non-Hispanic whites race. Again, we subsampled 10k examples with 80-20 split from the dataset and followed the method described in Sec H.2 and H.3 to generate a clean dataset for pre-training a clean classifier and generating a dataset with different predictive correlation respectively.

Synthetic Dataset. To accurately evaluate the whether a classifier is using the spurious concept or not, we introduce a Synthetic-Text dataset where it is possible to change the text input based on the change in concept (thus implementing Def 2.1). The main-task is to predict whether a sentence contains a numbered word (e.g. *one, fifteen* etc) or not, and the spurious concept is the length of the sentence which is correlated with the main task label. To create a sentence with numbered words, we randomly sample 10 words from the following set and combine them to form the sentence.

Numbered Words = one, two, three, four, five, six, seven, eight,
nine, ten, eleven, twelve, thirteen, fourteen,
fifteen, sixteen, seventeen, eighteen, twenty,
thirty, forty, fifty, sixty, seventy, eighty,
ninety, hundred, thousand

otherwise 10 non-numbered words randomly sampled from the following set were added to sentence :

Non-Numbered Words = nice, device, try, picture, signature, trailer,
harry, potter, malfoy, john, switch, taste,
glove, balloon, dog, horse, switch, watch,
sun, cloud, river, town, cow, shadow,
pencil, eraser

Next, we introduce the spurious concept (length) by increasing the length of the sentences which contain numbered words. We do so by adding a special word "pad" 10 times. In our experiment we used 1k examples created using the above described method and create 80-20 split for train and test set. Again, we follow the method described in Sec H.2 and H.3 to generate a clean dataset for training a clean classifier and to generate datasets with different predictive correlations respectively.

H.2. Creating a "clean" dataset with no spurious correlation with main-label

To construct a new dataset with no spurious correlation between the main-task and the concept label, we subsample only those examples from the the given dataset which have a fixed value of the spurious-concept label (y_p). Thus, if we train main-task classifier using this dataset, it cannot use the spurious-concept since they are not discriminative of the main task label (Ravichander et al., 2021).

In MultiNLI dataset we select only those examples which have no *negation* words in the sentence for the main-task for creating a clean dataset. Similarly, for Twitter-PAN16 dataset, we only select those examples which have gender label $y_p = -1$ in the processed dataset. . And for Twitter-AAE dataset, we only select those examples which have *non-Hispanic whites* race label.

H.3. Creating datasets with spurious correlated main and concept label

Since in our experiment, both the main-task label (y_m) and concept label (y_p) are binary (-1 or 1), they divide the dataset into 2×2 subgroups for each combination of (y_m, y_p). In MultiNLI dataset, the contradiction label $(y_m = 1)$ is correlated with the presence of negation words $y_p = 1$, this implies that the not-contradiction label $y_m = -1$ is also correlated with *absence* of negation words in the sentence $y_p = -1$. Thus, the input example with $(y_m = 1, y_p = 1)$ and $(y_m = -1, y_p = -1)$ form the majority group, henceforth referred as S_{maj} while groups $(y_m = 1, y_p = -1)$ and $(y_m = -1, y_p = 1)$ forms the minority group S_{min}. To evaluate the robustness of the removal methods, we create multiple datasets with different *predictive correlation* (κ) between the two labels y_m and y_p where $\kappa = P(y_m \cdot y_p) > 0$ as defined in Sec G. In other

words, to create a dataset with a particular predictive correlation κ, we vary the size of S_{maj} and S_{min}. More precisely, the predictive correlation can be equivalently defined in terms of the size of the these groups as:

$$\kappa = \frac{|S_{maj}|}{|S_{maj}| + |S_{min}|} \tag{124}$$

Similarly for Twitter-PAN16, Twitter-AAE and Synthetic-Text dataset, the we artificially create spurious correlation between y_m and y_p by creating the S_{maj} and S_{min} to have the desired predictive correlation (κ).

H.4. Encoder for real datasets

For all the experiments on real datasets in Sec G we used RoBERTa as default encoder h. In Sec I, we report the results when using BERT instead of RoBERTa as input encoder.

RoBERTa We use the Hugging Face(Wolf et al., 2019) *transformers* implementation of RoBERTa(Liu et al., 2019) *roberta-base* model, starting with pretrained weights for encoding the text-input to latent representation. We use default tokenizer and model configuration in our experiment.

BERT We use the Hugging Face(Wolf et al., 2019) *transformers* implementation of BERT(Devlin et al., 2019) *bert-base-uncased* model, starting with pretrained weights for encoding the text-input to latent representation. We use default tokenizer and model configuration in our experiment.

For both BERT and RoBERTa, the parameters of the encoder were fine-tuned as a part of training the main-task classifier for null-space removal and then frozen. For adversarial removal, the encoder, main-task classifier and the adversarial probing classifier are trained jointly. For both BERT and RoBERTa, we use the pooled output ($[CLS]$ token for BERT) from the the model, as the latent representation and is given to main-task and probing classifier.

H.5. Encoder for synthetic Dataset

nBOW: neural Bag of Word. For Synthetic-Text dataset, we use sum of pretrained-GloVe embedding(Pennington et al., 2014) of the words in the sentence to encode the sentence into latent representation. We used Gensim (Rehurek & Sojka, 2011) library for acquiring the 100-dimensional GloVe embedding (*glove-wiki-gigaword-100*). Throughout all our experiments, the word embedding was not trained. Post encoding, the latent representation were further passed through hidden layers consisting of a linear transformation layer followed by relu (Agarap, 2018) non-linearity. We will specify how many such hidden layers were used when discussing specific experiments in Sec I. The hidden layer dimensions were fixed to 50 dimensional space.

H.6. Null-Space Removal Experiment Setup

For null-space removal (INLP) experiment on both real and synthetic dataset the following procedure is followed:

1. Pretraining Phase: A *clean* pretrained main-task classifier is trained using the *clean* dataset obtained by method described in Suppl H.2. This is to ensure that the main-task classifier does not use the spurious feature, so that the INLP method doesn't have any effect on the main-task classifier. The main-task classifier is a linear-transformation on the latent-representation provided by encoder followed by softmax layer for prediction. Both the encoder and main-task classifier is fine-tuned during this process.

2. Removal Phase: Both the encoder and main-task classifier is frozen (made non-trainable). Next, a probing classifier is trained from the latent representation of the encoder (refer Suppl H.4 and H.5 for more details about encoder). The probing classifier is also a linear transformation layer followed by softmax layer for prediction.

The main-task classifier and encoder in the pretraining phase and the probing classifier in the removal phase is trained using cross-entropy loss for both real and synthetic datasets. For the real dataset, a fixed learning rate of 1×10^{-5} is used when RoBERTa is used as encoder and 5×10^{-5} when using BERT as encoder. For synthetic experiments, a fixed learning rate of 5×10^{-3} is used when training both the nBOW encoder and main-task classifier in the pretraining stage and probing classifier in removal stage.

H.7. Adversarial Removal Experiment Setup

For adversarial removal (AR) experiment, for both real and synthetic datasets, first the input text is encoded to latent representation using the encoder (Suppl H.4 and H.5). Then for the main-task classifier a linear transformation layer followed by softmax layer is applied for the main-task prediction. The same latent representation output from encoder is given to the probing classifier which is a separate linear transformation layer followed by softmax layer. All components of the model, encoder, main-task classifier and probing classifier are trained using the following modified objective from Eq 87:

$$arg \min_{h,c_m,c_p} \Big\{ L(c_m(h(\boldsymbol{z})), y_m) + \lambda L(c_p(g_{-1}(h(\boldsymbol{z}))), y_p) \Big\}$$

where h is the encoder, c_m is the main task classifier, c_p is the probing classifier, g_{-1} is the gradient reversal layer with fixed reversal strength of -1. The first term in the objective is for training the main task classifier and the second term is the adversarial objective for training the probing classifier using gradient reversal method (Ganin & Lempitsky, 2015; Elazar & Goldberg, 2018) . The hyperparameter λ controls the strength of the adversarial objetive. In our experiment we very $\lambda \in \{0.00001, 0.0001, 0.001, 0.01, 0.1, 0.5, 1.0, 2.0\}$. When describing the experimental results in Sec I.3 we choose the λ which performs the best for all datasets with different predictive correlation κ in removing the undesired concept from the latent representation.

H.8. Metrics Description

For simplicity, in all our experiments we assume that both the main and the correlated attribute label are binary. We measure the degree of spuriousness using the following two metrics:

1. Spuriousness Score: As defined in Sec F, this metric help us quantify, how much a classifier is using the spurious feature.
2. Δ Probability: In Synthetic-Text dataset as described in H.1, we have the ability to change the input corresponding to the change in concept label (thus implementing Def 2.1) . Thus we could measure if the main-classifier is using the spurious-concept by changing the concept in the input and measuring the corresponding change in the main-task classifier's prediction probability. Higher the change in prediction probability higher the main-task classifier is dependent on spurious-concept.

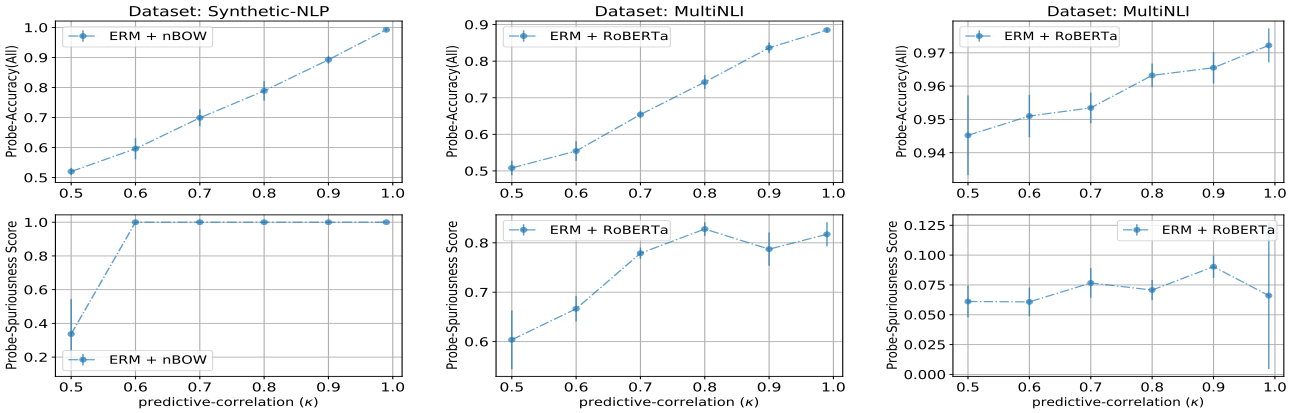
I. Additional Results

I.1. Probing classifier Quality

Fig 6 shows different failure modes of the probing classifier. In Fig 6a and 6b, a *clean* main-task classifier which doesn't use the concept feature is trained on Synthetic-Text and MultiNLI dataset respectively using the method described in Sec H.2. Thus the latent representation doesn't have the concept feature. Then, to test the presence of concept-causal feature in the latent representation we train a probing classifier to predict concept-label. The first row show the accuracy of the probing classifier for testing the presence of concept in latent space. When $\kappa = 0.5$ i.e no correlation between the main-task and the concept label, the probing accuracy is approximately 50% which correctly shows the absence of the concept-causal feature in the latent representation. The accuracy increases as the correlation κ between the main and concept-causal feature increases in dataset. This shows that even when concept-causal feature is not present in the latent representation, probing classifier will still claim presence of concept-causal feature if any correlated feature (main-task feature in this case) is present in the latent space. In Fig 6c, the latent space contains the concept-causal feature as shown by accuracy of approximately 94.5% when $\kappa = 0.5$. But as κ increases the probing classifier's accuracy increases in the presence of correlated main-task feature which falsely increases the confidence of presence of the concept-causal feature. The second row shows the spuriousness-score of concept-probing classifier is increasing as the correlation between the main-task and concept-causal feature increases which implies that the probing classifier is using relatively large *amount* of correlated main-task feature for concept-label prediction in all settings.

I.2. Extended Null-Space Removal Results

Fig 7 and 8, shows the failure mode of null-space removal (INLP) in the real dataset when using RoBERTa and BERT as encoder respectively. Different columns of the figure are for three different real datasets — MultiNLI, Twitter-PAN16, and



(a) Synthetic-Text + Probe Feature Absent (b) MultiNLI + Probe Feature Absent (c) MultiNLI + Probe Feature Present

Figure 6: **Failure Modes of Probing classifier:** The first row in Fig 6a and 6b shows that even when the latent representation doesn't contain the probing concept-causal feature, the probing classifier is still has >50% accuracy when other correlated feature is present. The accuracy increases as the correlation κ between the probing concept-causal feature and other correlated feature increases. The first row Fig 6c shows, that presence of correlated feature could increase the probing classifier's accuracy thus increasing the confidence in the presence of concept-causal feature in latent representation. The second row of all the figures shows that the probing classifier is getting more spurious as the κ increases thus implying that the probing classifier is using some other correlated feature than concept-causal feature. For more discussion see Suppl I.1.

Twitter-AAE respectively. The x-axis from step 8-26 are different INLP removal steps. The y-axis shows different metrics to evaluate the main-task and probing classifier. Different colored lines shows the spurious correlation (κ) in the probing dataset used by INLP for removal of spurious-concept. The pretrained classifier is clean, i.e., does not use the spurious concept-causal feature; hence INLP shouldn't have any effect on main-classifier when removing concept-causal feature from the latent space. The first row shows that as the INLP iteration progresses, the norm of latent representation, which is being *cleaned* of concept-causal feature, decreases. This indicates that some features are being removed. However, the results are against our expectation from the second statement of Theorem 2.2, which states that the norm of classifier will tend to zero as the INLP removal progresses. The possible reason is that, from Theorem 2.2 the norm of latent representation will go to zero when the the latent representation only contains the spurious concept-causal feature and the other features correlated to it. But, the encoder representation could have other features which are not correlated with concept-label and hence not removed. Since, the pretrained classifier given for INLP was *clean* (using method described in Sec H.2), we do not expect the INLP to have any effect on the main-task classifier.

The second row in Fig 7 and 8 shows that the main-classifier accuracy drops to random guess i.e 50% except for the case when probing dataset have $\kappa = 0.5$ i.e no correlation between the main and concept label. Thus INLP method corrupted a clean-classifier and made it useless. The reason behind this could be observed from the fourth and fifth row. The fourth row show the accuracy of the probing classifier before the projection step. We can see that at step 8 on x-axis $\kappa = 0.5$, the probing classifier correctly has accuracy of 50% showing that the concept-causal feature is not present in the latent representation. But for other value of κ, the probing classifier accuracy is proportional to value of κ implying that the probing classifier is using the main-task feature for its prediction. Hence at the time of removal, it removes the main-task feature which leads drop in the main-task accuracy. This can also be verified form the last row of Fig 7 and 8, which shows that the spuriousness score of probing classifier is high; thus it is using the main-task feature for its prediction. We observe similar results for Synthetic-Text dataset when using INLP in Fig 9. For all the INLP experiment on Synthetic-Text dataset, there were no hidden layers after the nBOW encoder (see Sec H.5).

I.3. Extended Adversarial Removal Results

Adversarial removal failure in real-world datasets. Fig 10 shows the failure mode of adversarial removal AR on real-world datasets. In x-axis we vary the predictive correlation κ between the main and the concept-label in different datasets and measure the performance of AR on different metrics on the y-axis. The second row shows the spuriousness score of the main-task classifier after AR as we vary κ on the x-axis. When using RoBERTa as the encoder, the orange curve in second row shows the spuriousness score of the main-task classifier when trained using the ERM loss. The spuriousness score describes how much unwanted concept-causal feature the main-task classifier is using. The blue curve shows that

the AR method reduces the spuriousness of main-task though cannot completely remove it. The reason for this failure can be attributed to probing classifier. Even when AR has successfully removed the unwanted concept feature, the accuracy of concept-probing classifier will be proportion to κ due to presence of correlated main-task feature in the latent space. This can be seen in the third row of the of Fig 10. Thus we cannot be sure if the unwanted concept-causal feature has been completely removed from the latent space or just became noisy enough to have accuracy proportional to κ after AR converges. In Fig 10, for each dataset and encoder, we manually choose the hyperparameter described λ described in Sec H.7 which reduces the spuriousness score most for the main-task classifier while not hampering the main-task classifier accuracy. In Fig 11, we show the trend in spuriousness score is similar for all choices of hyperparameter λ in our search. No value of λ is able to completely reduce the spuriousness score to zero.

Adversarial removal makes a classifier unfair. Fig 12 shows that when the adversarial classifier is initialized with a clean main-task classifier which doesn't use unwanted-concept causal features, it makes matters worse by making the main-task classifier use the unwanted-concept feature. For the Synthetic-Text dataset, since the word embedding are non-trainable, one single hidden layer is applied after the nBOW encoder so that AR methods could remove the unwanted-concept feature from the new latent representation.

I.4. Synthetic-Text dataset Ablations

Adversarial Removal Failure in Synthetic-Text dataset: Figure 13 shows the failure of AR on the synthetic dataset as we vary the noise in the main-task label and unwanted concept-label. For the experiment, since the words embedding are non-trainable, one single hidden layer is applied after the nBOW encoder so that AR methods could remove the unwanted-concept feature from the new latent representation.

Dropout Regularization Helps AR method: Continuing on observation from Fig 14a, 14b and 14c shows the Δ-Prob of the main-task classifier after we apply the AR on Synthetic-Text dataset and how they changes as we increase the dropout regularization. As we increase the dropout (drate in the figure), the Δ-Prob of the main classifier decreases showing that the regularization methods could help improve the the removal methods.

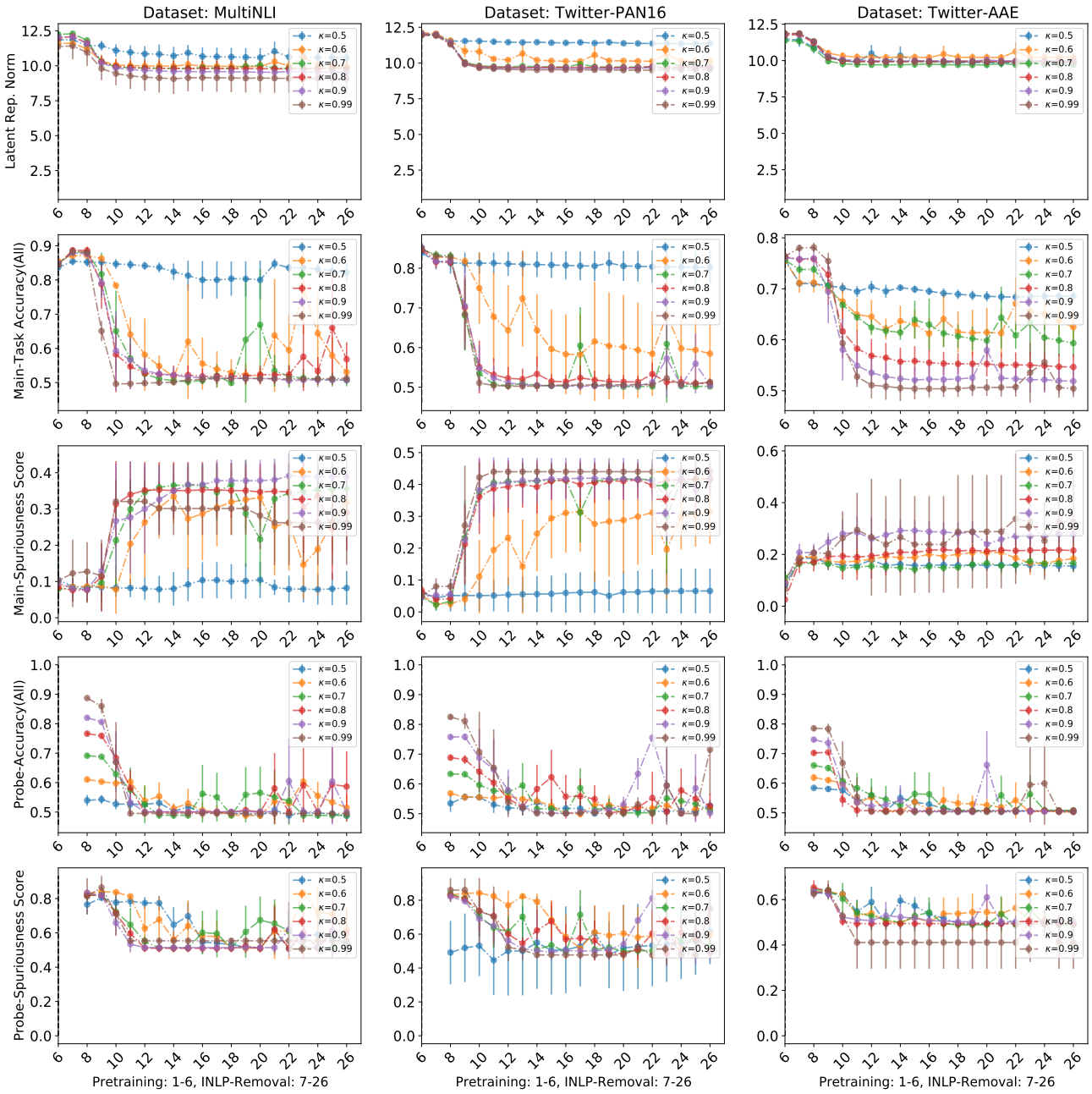


Figure 7: **Failure of Null Space Removal when using RoBERTa as encoder:** Different columns of the figure are for three different real datasets — MultiNLI, Twitter-PAN16, and Twitter-AAE respectively. The x-axis from steps 8-26 are different INLP removal steps. The y-axis shows different metrics to evaluate the main-task and probing classifier. Different colored lines shows the spurious correlation (κ) in the probing dataset used by INLP for removal of spurious-concept. The pretrained classifier is clean i.e. doesn't uses the spurious concept-causal feature, hence INLP shouldn't have any effect on main-classifier when removing concept-causal feature from the latent space. Against our expectation, the second row shows that the main-task classifier's accuracy is decreasing even when it is not using the concept-feature. The main reason for this failure to learn a clean concept-probing classifier. This can be verified from the last row which shows that the concept-probing classifier has high spuriousness score thus implying that it is using the main-task feature for concept label prediction and hence during removal step, wrongly removing the main-task feature which leads to drop in main-task accuracy. For more discussion see Suppl I.2.

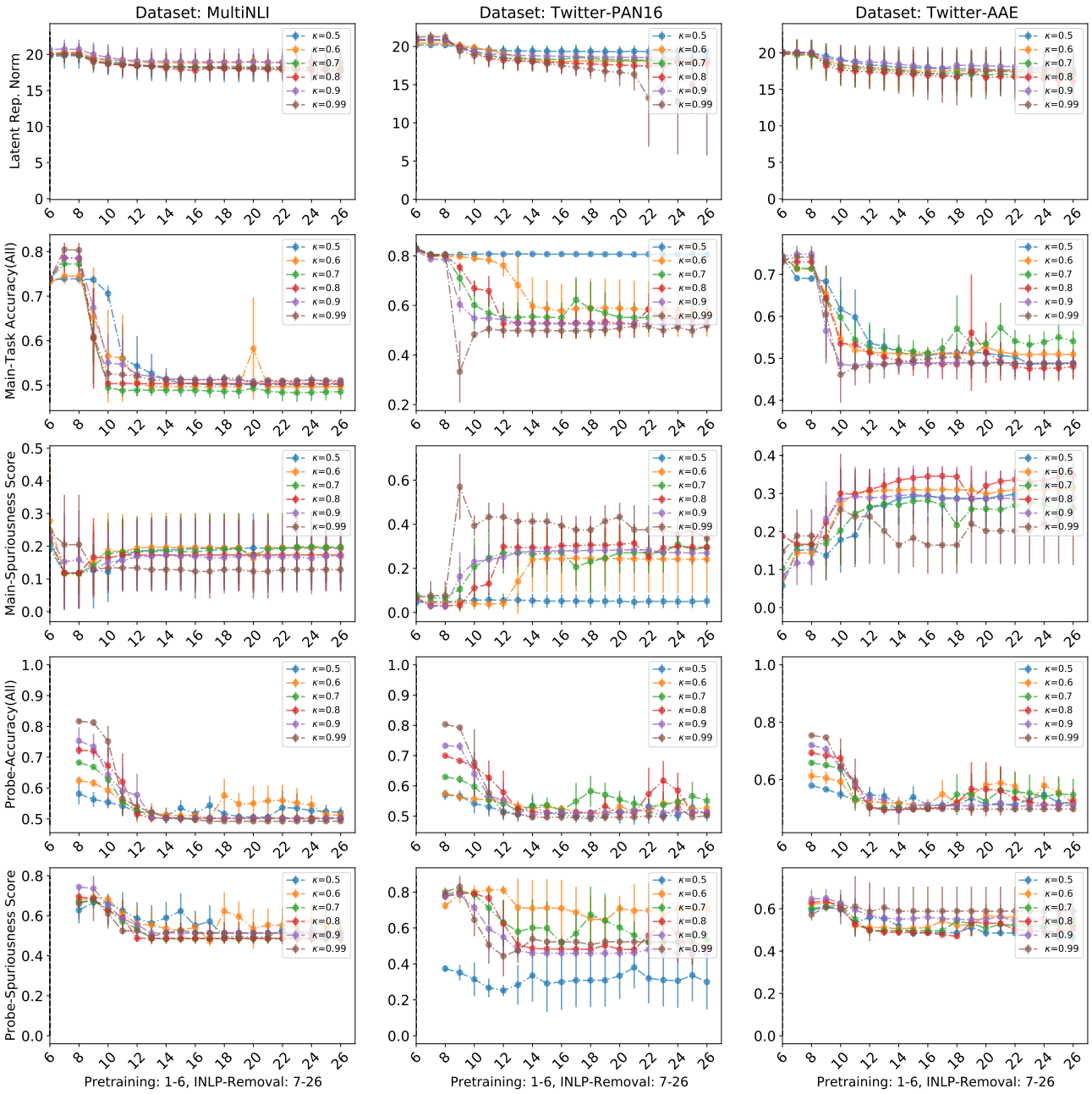
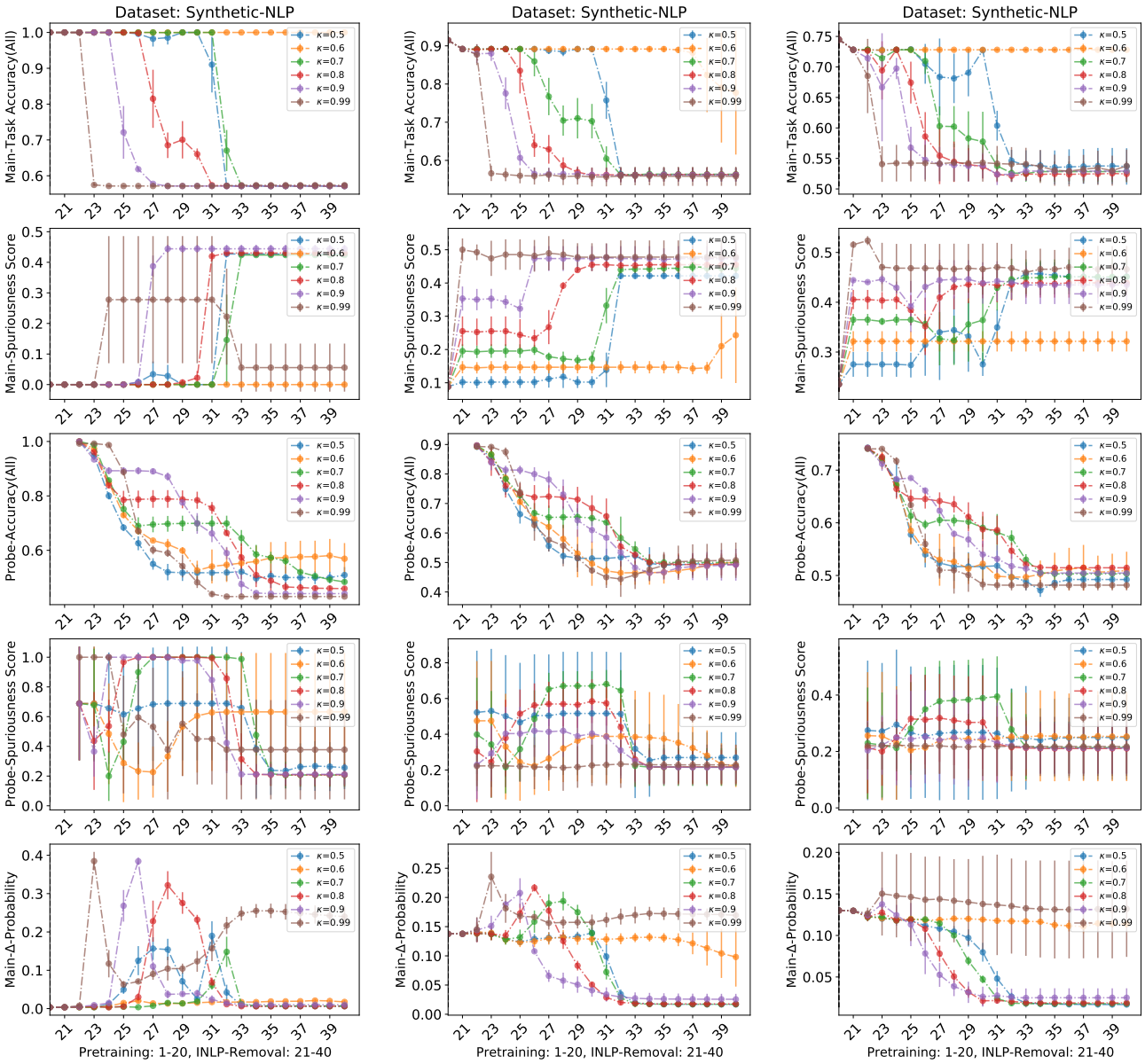


Figure 8: **Failure of Null Space Removal when using BERT as encoder:** The observation is similar to the case when RoBERTa was used as encoder (see Fig 7) . Different columns of the figure are for three different real datasets — MultiNLI, Twitter-PAN16, and Twitter-AAE respectively. The x-axis from steps 8-26 are different INLP removal steps. The y-axis shows different metrics to evaluate the main-task and probing classifier. Different colored lines shows the spurious correlation (κ) in the probing dataset used by INLP for removal of spurious-concept. The pretrained classifier is clean i.e. doesn't uses the spurious concept-causal feature, hence INLP shouldn't have any effect on main-classifier when removing concept-causal feature from the latent space. Against our expectation, the second row shows that the main-task classifier's accuracy is decreasing even when it is not using the concept-feature. The main reason for this failure to learn a clean concept-probing classifier. This can be verified from the last row which shows that the concept-probing classifier has high spuriousness score thus implying that it is using the main-task feature for concept label prediction and hence during removal step, wrongly removing the main-task feature which leads to drop in main-task accuracy. For more discussion see Suppl I.2.



(a) Synthetic-Text + n=0.0 (b) Synthetic-Text + n=0.1 (c) Synthetic-Text + n=0.3

Figure 9: **Failure Mode of INLP in Synthetic-Text dataset**: Different columns of the figure are Synthetic-Text dataset with different level of noise in the main-task and probing task label. The x-axis from step 22-40 are different INLP removal steps. The y-axis shows different metrics to evaluate the main-task and probing classifier. Different colored lines shows the spurious correlation (κ) in the probing dataset used by INLP for removal of spurious-concept. The pretrained classifier is clean i.e. doesn't uses the spurious concept-causal feature, hence INLP shouldn't have any effect on main-classifier when removing concept-causal feature from the latent space. Contrary to our expectation, the first row shows main-task classifier accuracy drops as the INLP progresses. Higher the correlation between the main-task and concept label, faster is the drop in the main-task accuracy. The last row shows the change in prediction probability (Δ-Prob) of main-task classifier when we change the input corresponding to concept-label. This shows, how much sensitive the main-task classifier is wrt. to concept feature. We observe that the Δ-Prob increases in the middle of INLP showing that the main-classifier which was not using the concept initially (as in iteration 21), started using the sensitive concept because of INLP removal. Thus stopping INLP prematurely could lead to a more *unclean* classifier than before whereas running INLP longer removes all the correlated feature and could make the classifier useless. For more discussion see Suppl I.2.

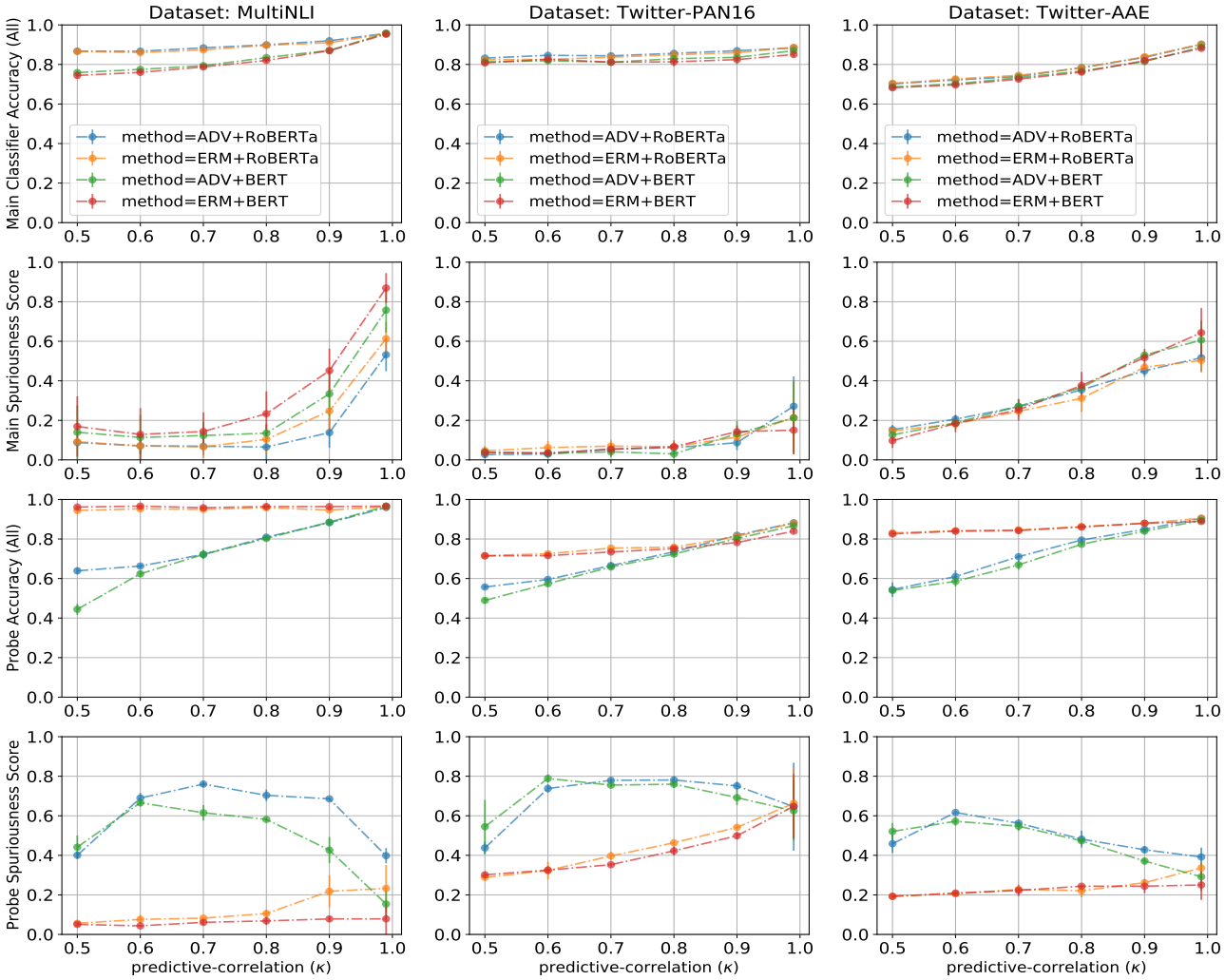


Figure 10: **Failure Mode of Adversarial removal on real-dataset:** Different column shows the result on three different real datasets —MultiNLI, Twitter-PAN16 and Twitter-AAE respectively. The second row shows that accuracy of spuriousness score of the main-task classifier after AR, when the dataset contains different level of spurious correlation between the main-task and unwanted-concept label, denoted by κ in the x-axis. When using RoBERTa as the encoder, the orange curve in second row shows the spuriousness score of the main-task classifier when trained using the ERM loss. The spuriousness score describes how much unwanted concept-causal feature the main-task classifier is using. The blue curve shows that the AR method reduces the spuriousness of main-task though cannot completely remove it. When using BERT as encoder, the observation is same i.e green curve in second row shows AR is able to reduce the spuriousness of main classifier than the red curve which is trained using ERM, but not able to complete remove the spurious feature. For more discussion see Sec I.3.

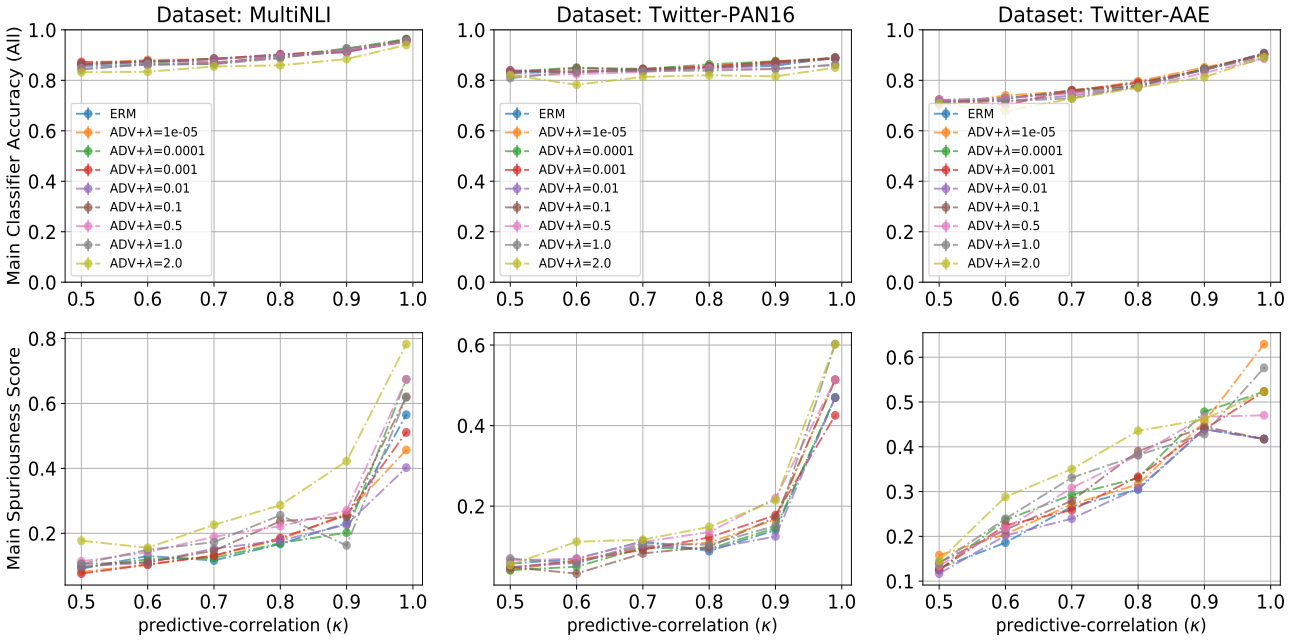
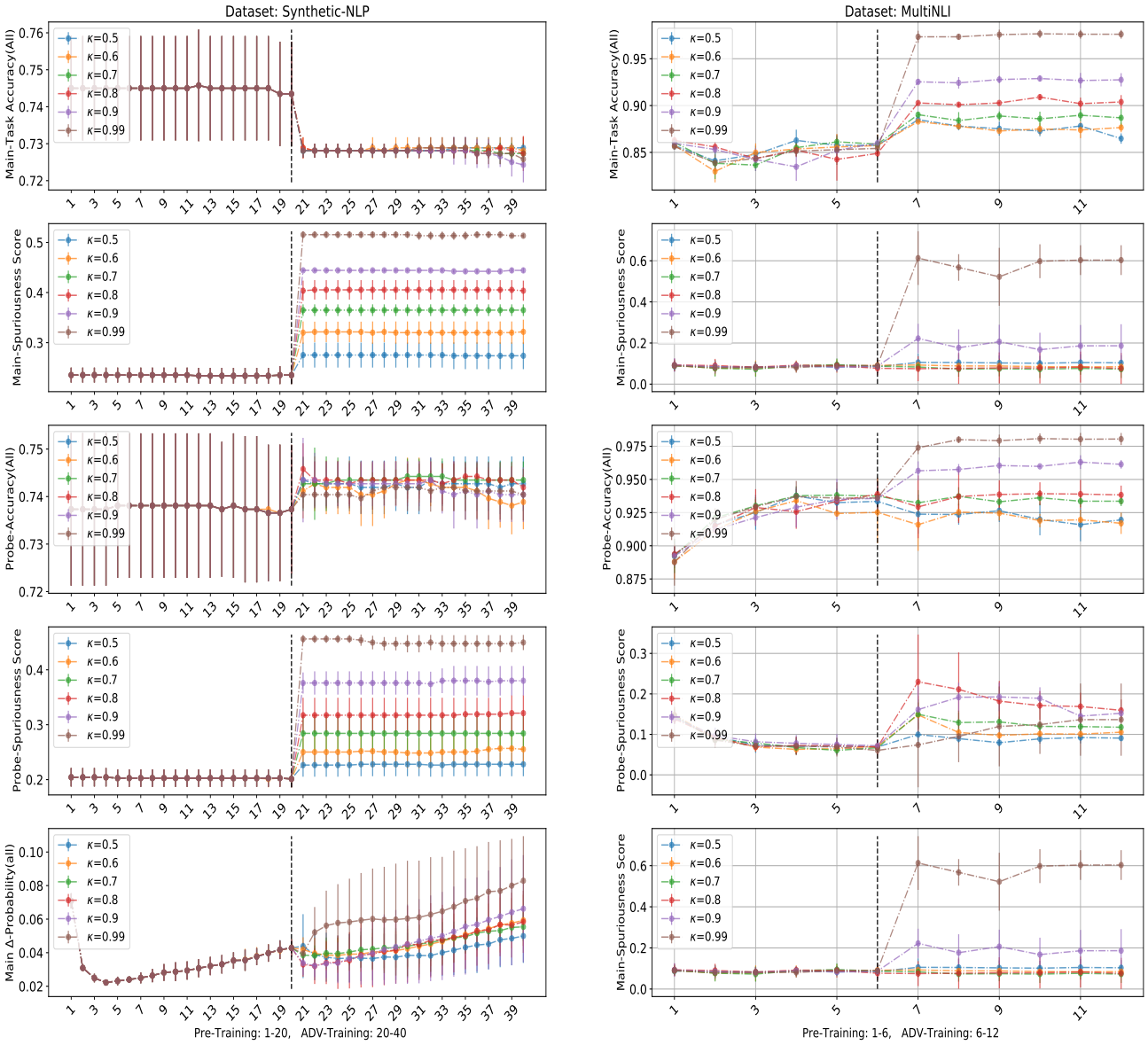


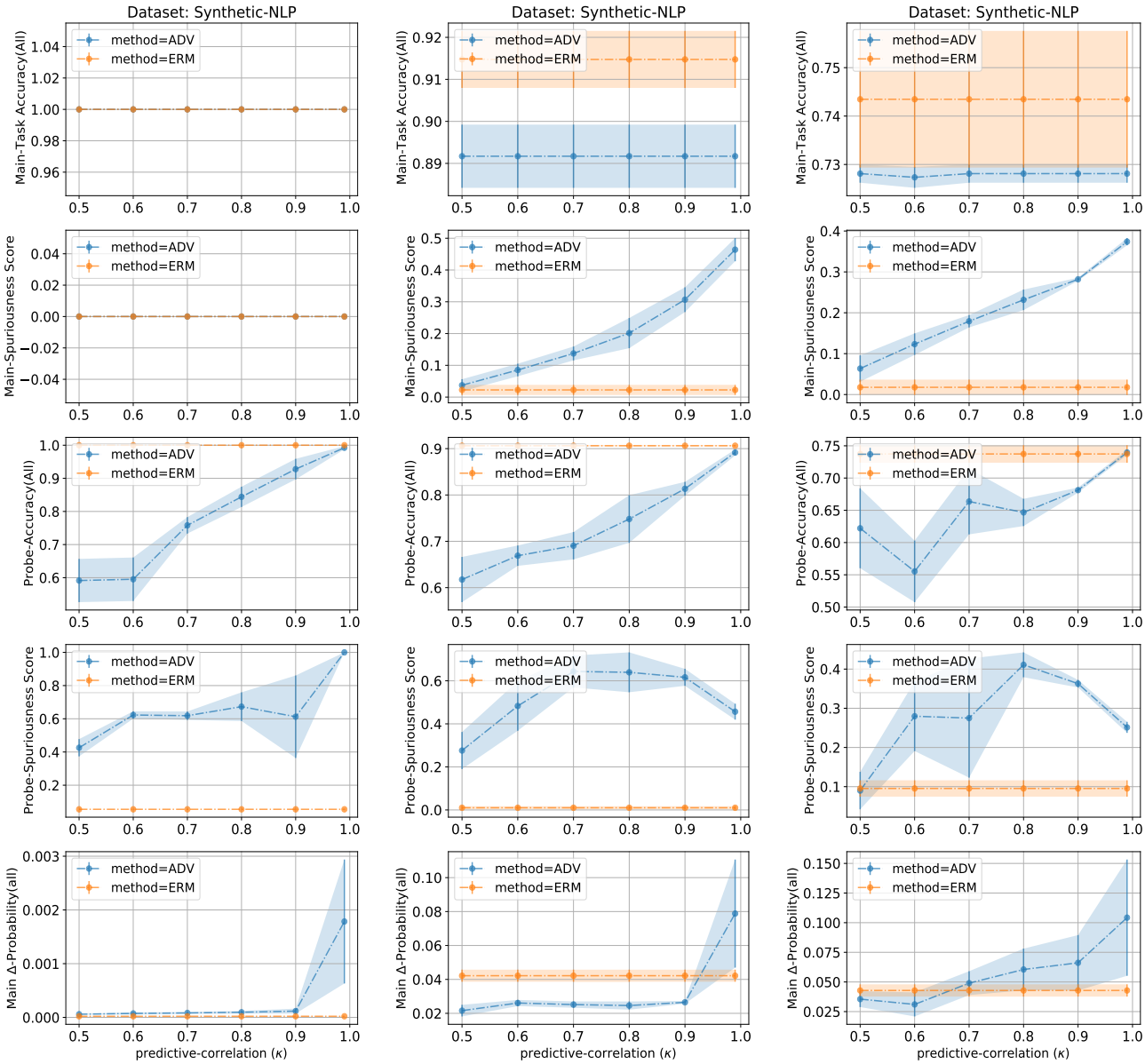
Figure 11: **Choice of Adversarial Strength Parameter** λ: The second plot shows that trend in spuriousness score after AR is similar for all the choice of hyperparameter λ we have taken in our search. None of the setting of λ is able to completely reduce the spuriosuness score to zero. For more discussion see Sec I.3.



(a) Synthetic-Text + nBOW

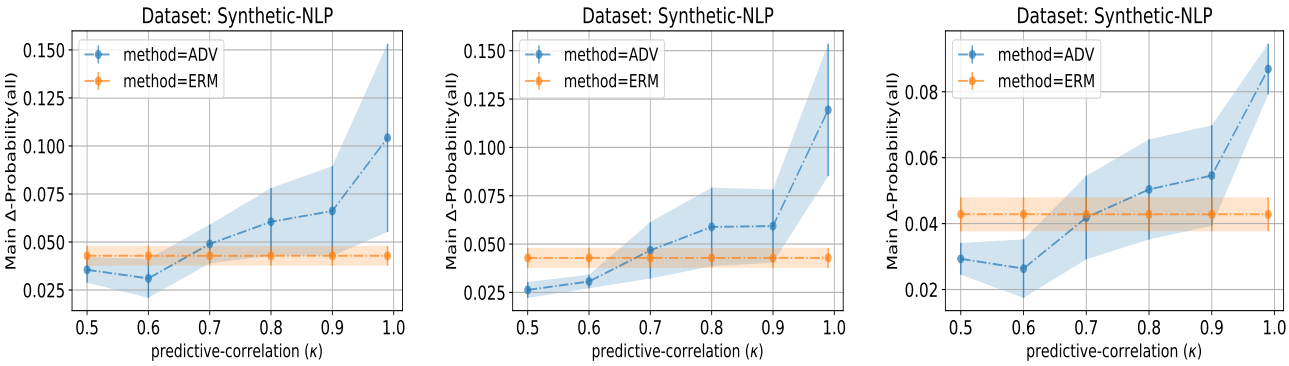
(b) MultiNLI + RoBERTa)

Figure 12: **Adversarial Removal Makes a classifier unfair**: We test if the AR method increases the spuriousness of a main-task classifier if initialized with a *clean* classifier. In 12a, from iteration 1-20 in x-axis, a clean classifier is trained on Synthetic-Text datset such that it doesn't uses the unwanted concept-causal feature using method described in Suppl H.2. Then the classifier is given to AR method for removing the the unwanted concept feature which make the initially clean classifier unlean. This can be seen from the second row of the 12a which shows the spuriousness score of main-classifier is 0 during 1-20 iteration but increases after the AR starts from 21-40. Also the last row shows the δ-Prob of the main-task classifier on changing the unwanted-concept in input which increases for large dataset which have large κ i.e correlation between the main and concept label. Similar result can be seen for the MultiNLI dataset where a clean classifier is trained in iteration 1-6 which is made unclean by AR. Second row again show that spuriousness score of main-task classifier increases after AR starts in iteration 7-12. For more discussion see Sec I.3.



(a) Synthetic-Text + n=0.0 (b) Synthetic-Text + n=0.1 (c) Synthetic-Text + n=0.3

Figure 13: **Failure of Adversarial Removal method on Synthetic-Text dataset:** Different columns show the adversarial removal method on Synthetic-Text dataset with different level of noise in the main-task and concept label. When there is no noise, from the second row in Fig 13a, we see that both the classifier trained by ERM and AR has zero-spuriousness score. But as we increase the noise to 10% in Fig 13b, we observe that the the spuriousness score increases when AR is applied in contrast to classifier trained by ERM which stays at 0. Also, higher the predictive correlation κ, higher is the increase in spuriousness. This observation augments to the the observation in Fig 12 which shows that using AR makes a clean classifier unclean. Similarly in Fig 13c when we increase the noise to 30% we observe in second row, AR is increase the spuriousness unlike ERM which is at 0. For discussion see Suppl I.4



(a) Synthetic-Text + drate=0.0 (b) Synthetic-Text + drate=0.5 (c) Synthetic-Text + drate=0.9

Figure 14: **Dropout Regularization helps in Adversarial Removal:** Δ-Prob of the main-task classifier after we apply the AR on Synthetic-Text dataset decreases as we increase the dropout regularization from 0.0 to 0.9. For discussion see Suppl I.4.