Are All Spurious Features in Natural Language Alike? An Analysis through a Causal Lens

Nitish Joshi[1*] **Xiang Pan**[1*] **He He**[1,2]
[1]Department of Computer Science, New York University
[2]Center for Data Science, New York University
{nitish, xiangpan, hhe}@nyu.edu

Abstract

The term 'spurious correlations' has been used in NLP to informally denote any undesirable feature-label correlations. However, a correlation can be undesirable because (i) the feature is irrelevant to the label (e.g. punctuation in a review), or (ii) the feature's effect on the label depends on the context (e.g. negation words in a review), which is ubiquitous in language tasks. In case (i), we want the model to be invariant to the feature, which is neither necessary nor sufficient for prediction. But in case (ii), even an ideal model (e.g. humans) must rely on the feature, since it is necessary (but not sufficient) for prediction. Therefore, a more fine-grained treatment of spurious features is needed to specify the desired model behavior. We formalize this distinction using a causal model and probabilities of necessity and sufficiency, which delineates the causal relations between a feature and a label. We then show that this distinction helps explain results of existing debiasing methods on different spurious features, and demystifies surprising results such as the encoding of spurious features in model representations after debiasing.

1 Introduction

Advancements in pre-trained language models (Devlin et al., 2019; Radford et al., 2019) and large datasets (Rajpurkar et al., 2016; Wang et al., 2018) have enabled tremendous progress on natural language understanding (NLU). This progress has been accompanied by the concern of models relying on superficial features such as negation words and lexical overlap (Poliak et al., 2018; Gururangan et al., 2018; McCoy et al., 2019). Despite the progress in building models robust to spurious features (Clark et al., 2019; He et al., 2019; Sagawa* et al., 2020; Veitch et al., 2021; Puli et al., 2022), the term has been used to denote any feature that the model should not rely on, as judged by domain experts.

Irrelevant features
Speilberg's new film is brilliant. ⟶ Positive
_____'s new film is brilliant. ⟶ Positive
Necessary features
The differential compounds to a hefty sum over time.
The differential will not grow ⟶ Contradiction
The differential will ___ grow ⟶ ?

Table 1: Difference between two spurious features: (a) the director name can be replaced without affecting the sentiment prediction; (b) the negation word is necessary as it is not possible to determine the label without it.

Our key observation is that a feature can be considered spurious for different reasons. Compare two such features studied in the literature (Table 1): (a) director names (such as 'Spielberg') in sentiment analysis (Wang and Culotta, 2020); (b) negation words in natural language inference (Gururangan et al., 2018). We do not want the model to rely on the director name because removing or changing it does not affect the sentiment. In contrast, while models should not *solely* rely on the negation word, they are still *necessary* for prediction—it is impossible to determine the label without knowing its presence.

In this work, we argue that many spurious features studied in NLP are of the second type where the feature is necessary (although not sufficient) for prediction, which is more complex to deal with than completely irrelevant features in the first case. Current methods do not treat the two types of feature separately, and we show that this can lead to misleading interpretation of the results.

To formalize the distinction illustrated in Table 1, we borrow notions from causality (Wang and Jordan, 2021; Pearl, 1999), and use probability of necessity (PN) and probability of sufficiency (PS) to describe the relation between a feature and a label. Intuitively, high PN means that changing the feature is likely to change the label (e.g. remov-

*equal contribution

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9804–9817
December 7-11, 2022

ing "not" will flip the label); high PS means that adding the feature to an example would produce the label (e.g. adding "the movie is brilliant" to a neutral review is likely to make it positive). Under this framework, we define two types of spurious features (Section 2): irrelevant features (e.g. the director name) that have low PN and low PS, and necessary features (e.g. the negation word) that have high PN despite low PS.

Next, we describe the challenges in evaluating and improving robustness to necessary spurious features (Section 4). First, necessary features compose with other features in the context to influence the label. Thus, evaluating whether the model relies solely on the necessary feature requires perturbing the context. This process often introduces new features and leads to inconsistent results depending on how the context is perturbed.

Second, we analyze the effectiveness of two classes of methods—data balancing and representation debiasing—on the two types of spurious features. Data balancing breaks the correlation between the label and the spurious feature (e.g. Sagawa et al. (2020)); representation debiasing directly removes the spurious feature from the learned representation (e.g. Ravfogel et al. (2020)). Although they are effective for irrelevant features, we show that for necessary spurious features, (i) data balancing does not lead to invariant performance with respect to the spurious feature (Section 5.1); and (ii) removing the spurious feature from the representation significantly hurts performance (Section 5.2).

In sum, this work provides a formal characterization of spurious features in natural language. We highlight that many common spurious features in NLU are necessary (despite being not sufficient) to predict the label, which introduces new challenges to both evaluation and learning.

2 Categorization of Spurious Features

2.1 Causal Models

We describe a structural causal model for text classification to illustrate the relation between different spurious features and the label. Let $X = (X_1, X_2, .., X_n)$ denote a sequence of input words/features[1] and Y the output label. We assume a data generating model shown in Figure 1a. There is a common cause C of the input (e.g. a review writer, a PCFG or a semantic representation of the sentence), conditioned on which the words are independent to each other. Each word X_i may causally affect the label Y.

Under this model, the dependence between Y and a feature X_i can be induced by two processes. The *type 1 dependence* is induced by a confounder (in this case C) influencing both Y and X_i due to biases in data collection, e.g. search engines return positive reviews for famous movies; we denote this non-causal association by the red path in Figure 1b. The *type 2 dependence* is induced by input words that causally affect Y (the red path in Figure 1c), e.g. negating an adjective affects the sentiment. Importantly, the two processes can and often do happen simultaneously. For example, in NLI datasets, the association between negation words and the label is also induced by crowdworkers' inclination of negating the premise to create a contradiction example.

A type 1 dependence ("Titanic"-sentiment) is clearly spurious because the feature and Y are associated through C while having no causal relationship.[2] In contrast, a type 2 dependence ("not"-sentiment) is not spurious per se—even a human needs to rely on negation words to predict the label.

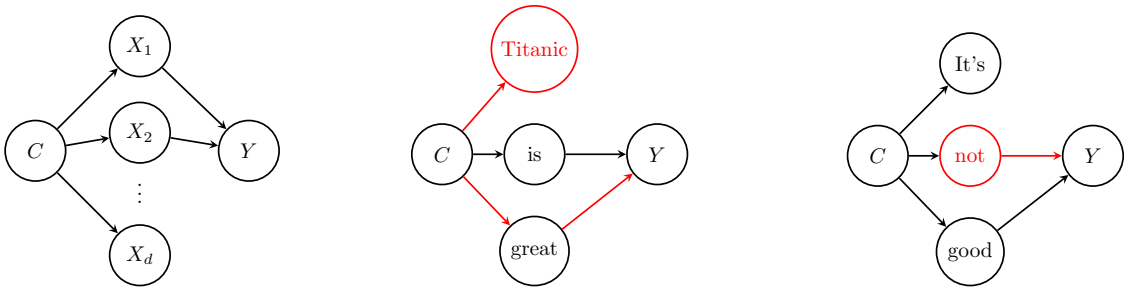
Now, how do we measure and differentiate the two types of feature-label dependence? In the following, we describe fine-grained notions of the relationship between a feature and a label, which will allow us to define the spuriousness of a feature.

2.2 Sufficiency and Necessity of a Feature

We borrow notions from causality to describe whether a feature is a necessary or sufficient cause of a label (Pearl, 1999; Wang and Jordan, 2021). Consider the examples in Table 1: intuitively, "not" is necessary for the contradiction label because in the absence of it (e.g. removing or replacing it by other syntactically correct words) the example would no longer be contradiction; in contrast, "the movie is brilliant" is sufficient to produce the positive label because adding the sentence to a negative review is likely to increase its sentiment score. Thus, the feature's effect on the label relies on counterfactual outcomes.

[1] For illustration purposes, we assume that each feature is a word in the input text. However, the same model and analysis apply to cases where X_i denote a more complex feature (e.g. named entities or text length) extracted from the input.

[2] The two types of dependencies are also discussed in Veitch et al. (2021), where the type 1 dependence is called "purely spurious".



(a) Data generating model. (b) Type 1 dependence. (c) Type 2 dependence.

Figure 1: Causal models for text classification. (a) C is the common cause of words in the input. Each word X_i may be causally influence Y. (b) Y (sentiment label) and X_i ("Titanic") are dependent because of the confounder C (indicated by the red path). (c) Y (sentiment label) and X_i ("not") are dependent because of a causal relation.

We use $Y(X_i = x_i)$ to denote the *counterfactual label* of an example had we set X_i to the specific value x_i.[3]

Definition 1 (Probability of necessity)**.** *The probability of necessity (PN) of a feature $X_i = x_i$ for the label $Y = y$ conditioned on context $X_{-i} = x_{-i}$ is*

$$PN(X_i{=}x_i, Y{=}y \mid X_{-i}{=}x_{-i}) \triangleq$$
$$p(Y(X_i \neq x_i) \neq y \mid X_i{=}x_i, X_{-i}{=}x_{-i}, Y{=}y) \; .$$

Given an example (x, y), $\text{PN}(x_i, y \mid x_{-i})$[4] is the probability that the label y would change had we set X_i to a value different from x_i. The distribution of the counterfactual label $Y(X_i \neq x_i)$ is defined to be $\int p(Y(X_i))p(X_i \mid X_i \neq x_i)\,\mathrm{d}X_i$. This corresponds to the label distribution when we replace the word x_i with a random word that fits in the context (e.g. "Titanic" to "Ip Man"). In practice, we can simulate the intervention $X_i \neq x_i$ by text infilling using masked language models (Devlin et al., 2019).

Definition 2 (Probability of sufficiency)**.** *The probability of sufficiency (PS) of a feature $X_i = x_i$ for the label $Y = y$ conditioned on the context $X_{-i} = x_{-i}$ is*

$$PS(X_i{=}x_i, Y{=}y \mid X_{-i}{=}x_{-i}) \triangleq$$
$$p(Y(X_i{=}x_i) = y \mid X_i{\neq}x_i, X_{-i}{=}x_{-i}, Y{\neq}y) \; .$$

Similarly, $\text{PS}(x_i, y \mid x_{-i})$ is the probability that setting X_i to x_i would produce the label y on an example where x_i is absent. For example, PS of "not" for the negative sentiment measures the probability that a positive review will become negative had we added "not" to the input.

[3]The counterfactual label $Y(X_i = x_i)$ is also commonly written as Y_{x_i} (Pearl, 2009) but we follow the notation in Wang and Jordan (2021)

[4]For notational simplicity, we omit the random variables (denoted by capital letters) when clear from the context.

We note that both PN and PS are *context-dependent*—they measure the counterfactual outcome of individual data points. For example, while "not" has high PN for contradiction in the example in Table 1, there are examples where it has low PN.[5] Similarly, there can be examples where the word "Titanic" has high PN.[6] To consider the average effect of a feature, we marginalize over the context X_{-i}:

$$\text{PN}(x_i, y) \triangleq \int \text{PN}(x_i, y \mid X_{-i})p(X_{-i} \mid x_i, y)\,\mathrm{d}X_{-i},$$

and similarly for PS.

Definition 3 (Spuriousness of a feature)**.** *The spuriousness of a feature $X_i = x_i$ for a label $Y = y$ is $1 - PS(x_i, y)$. We say a feature is spurious to the label if its spuriousness is positive.*

Our definition of the spuriousness of a feature follows directly from the definition of PS, which measures the extent to which a feature is a sufficient cause of the label (marginalized over the context X_{-i}). Following this definition, a feature is non-spurious only if it is sufficient in *any* context. Admittedly, this definition may be too strict for NLP tasks as arguably the effect of any feature can be modulated by context, making all features spurious. Therefore, practically we may consider a feature non-spurious if it has low spuriousness (i.e. high PS).

Feature categorization. The above definitions provide a framework for categorizing features by their necessity and sufficiency to the label as shown in Figure 2.

[5]Consider the premise "The woman was happy" and the hypothesis "The woman angrily remarked 'This will not work!'".

[6]For example in sentiment analysis, consider 'This movie was on a similar level as Titanic'.

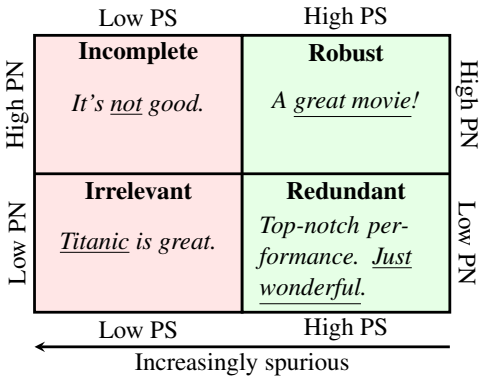


Figure 2: Categorization of features based on their PN and PS. Spurious features have low PS. Among them, the high PN ones are part of the features needed for prediction but they alone are not sufficient; and the low PN ones are irrelevant to prediction.

Estimating PN and PS. Calculating PN and PS of a feature requires knowing how the label would change when the feature is removed or added to an instance. Sometimes we can reason about it with domain knowledge. Consider the feature "Titanic" in Figure 1b, it has zero PN and PS since removing or adding it would not change the label.

In more complex cases, we might need to estimate the probabilities using an experiment. For example, consider the lexical overlap between the premise and hypothesis in NLI. Given an entailment example with high word overlap, changing the overlapped words is likely to cause label change (H1–3) unless it is replaced by a synonym (H4):

P: The doctor was paid by the actor.	
H0: The actor paid the doctor.	**L0**: Entailment
H1: The teacher paid the doctor.	**L1**: Neutral
H2: The actor liked the doctor.	**L2**: Neutral
H3: The actor paid the guard.	**L3**: Neutral
H4: An actor paid the doctor.	**L4**: Entailment

Since a non-synonym is more likely to be sampled during intervention thus causing a label change, we conclude that word overlap has high PN to entailment. On the other hand, it is not a completely sufficient feature (i.e. spuriousness > 0) since there are plenty of examples with high lexical overlap but non-entailment labels (McCoy et al., 2019).

We can partially automate this process by intervening examples using masked language models and then collecting labels for the perturbed examples. We discuss this method in more detail and provide preliminary results in Appendix A. However, we note that while PN/PS can be estimated through careful intervention, as a conceptual framework, domain knowledge often suffices to judge whether a feature has high or low PN/PS.

3 Experiment Setup

Before diving into the implications of our categorization of spurious features, we explain the common setup of experiments that we use to support our arguments.

Spurious features. Typical features considered in the literature (such as word overlap and negation words) fall into the high PN and low PS category. Therefore, in the following discussion, we will focus on two types of spurious features: low PN features that are irrelevant to prediction, and high PN features that are necessary but need additional context to decide the label.

Datasets. We use the following datasets that contain the spurious features. (i) **Low PN spurious features**: we inject synthetic bias to MNLI examples by associating a punctuation ('!!') with the neutral label. Following Dranker et al. (2021), we set bias prevalence (i.e. examples where '!!' is present) to 25% and set bias strength (i.e. percentage of examples with '!!' and the neutral label) to 90%. The dataset is created by modifying MNLI examples through adding/deleting the feature at the end of the hypothesis. (ii) **High PN spurious features**: we consider the negation bias (Poliak et al., 2018) and the lexical overlap bias in MNLI (Williams et al., 2018) for which we use the HANS challenge set (McCoy et al., 2019) during evaluation.

Models. For all our experiments, unless otherwise stated, we use RoBERTa-large (Liu et al., 2019) from Huggingface (Wolf et al., 2019) as the backbone model.

Training methods. Our baseline algorithm finetunes the pretrained model on the original dataset with cross-entropy loss. We also experiment with debiasing methods including Subsampling (Sagawa et al., 2020), Product-of-Expert (POE) and Debiased Focal Loss (DFL) (Karimi Mahabadi et al., 2020) for comparison. Hyperparameters and training details can be found in Appendix B.[7]

4 Implications on Model Robustness

Under the causal framework, we say a model is *non-robust* if it fails on the interventional distribu-

[7]Our code can be found at https://github.com/joshinh/spurious-correlations-nlp

Models	HANS Ent/Non-ent	Δ	MNLI subsets Ent/Non-ent	Δ
BERT-base	99.2/12.9	86.3	96.4/82.5	13.9
RoBERTa-large	99.9/56.2	43.7	97.1/93.6	3.5

Table 2: Results on two challenge sets for lexical overlap. Both indicate significantly different extent to which the models rely on the spurious correlation.

tion. For example, in Figure 1b the movie name has no causal effect on the label; if intervening on it (e.g. changing "Titanic") nevertheless incurs a prediction change, we say the model is not robust.

Is relying on spurious features always bad? Prior work has suggested that if the model prediction relies on a single feature in any way, it is undesired (Gardner et al., 2021). However, for a high PN feature, the label and the model output *should* depend on it (Figure 1c). Such dependency only becomes undesirable when other necessary features are ignored by the model (e.g. predicting negative sentiment whenever "not" is present). This can be caused by two reasons: first, the model may overly rely on a spurious feature X_i due to confounding between Y and X_i in the training data (e.g. "not" appears in all negative examples but not positive examples); second, even without confounding, the model may fail to learn how X_i interacts with other features to affect the label (e.g. not understanding double negation).

How to evaluate models' robustness? A typical way to test models' robustness is to construct a "challenge set" that tests if perturbations of the input cause model predictions to change in an expected way. The challenge here is that the expected behavior of a model depends on the type of the spurious feature. For low PN spurious features, we can simply perturb them directly and check if the model prediction is invariant, e.g. replacing named entities with another entity of the same type (Balasubramanian et al., 2020). Performance drop on this test set then implies that the model is non-robust. However, intervention on the spurious feature only tells us if the feature is necessary, thus it cannot be used to evaluate robustness to high PN spurious features, where the model prediction is likely (and expected) to flip if we perturb the feature (e.g. replacing "not" with "also" in Figure 1c).

For high PN spurious features like negation words, we instead want to test if they are sufficient for the model prediction. An alternate method is to create two sets of examples with the same spurious feature but different labels. For example, HANS (McCoy et al., 2019) consists of entailment and non-entailment examples, both having complete lexical overlap; this tests if high word overlap alone is sufficient to produce an entailment prediction. However, this process inevitably introduces a new variable. Consider the causal graph in Figure 1c. With the spurious feature ("not") fixed, to change Y we must change other features (e.g. "good" $\rightarrow$ "bad") that affect the label by interacting with "not". To make a correct prediction, the model must learn the *composite feature* formed by the spurious feature and the newly introduced features. As a result, its performance depends not only on the spurious feature but also on the features introduced during the perturbation.

Inconsistent results on different challenge sets. To illustrate this problem, we evaluate models' robustness to lexical overlap on two challenge sets constructed differently: (a) HANS; (b) subsets of high lexical overlap examples in the MNLI dev set (where > 0.8 fraction of words in the hypothesis are also in the premise). Compared to (b), HANS non-entailment examples require linguistic knowledge such as understanding passive voice (e.g. "The senators were helped by the managers" does not imply "the senators helped the managers") or adverbs of probability (e.g. "Probably the artists saw the authors" does not imply "the artists saw the authors"), which are rare in MNLI.

We fine-tune pre-trained models on MNLI and report their results in Table 2. While models perform poorly on high overlap non-entailment examples from HANS, their performance is much higher on such examples from MNLI (56.2% vs 93.6%), leading to inconsistent conclusions.[8] Thus, we should be careful when interpreting the magnitude of the problem on challenge sets, as the performance drop could also be attributed to unseen features introduced during dataset construction.

5 Implications on Learning Methods

In this section, we discuss two common classes of methods to train robust models and their effectiveness for spurious features with high/low PN.

[8] Large variance in performance across different subcases of non-entailment examples as reported in McCoy et al. (2019) is another example of the unreliability.

5.1 Decorrelating the Spurious Feature and the Label

A straightforward idea to remove undesirable correlation between the label Y and a spurious feature X_i due to confounding is to balance the training data such that Y and X_i are independent (Japkowicz, 2000; Austin, 2011; Li and Vasconcelos, 2019). In practice, this amounts to subsampling the dataset to balance the classes conditioned on the spurious feature (e.g. "Titanic is good/bad" are equally likely) (Sagawa et al., 2020), or upweighting examples where the spurious feature is not predictive for the label (Karimi Mahabadi et al., 2020). While these methods have shown promise for spurious features with both high and low PN, there is a key difference between the underlying mechanisms.

For a low PN spurious feature, the dependence between model prediction and the feature arises from a confounder that affects both Y and X_i. As shown in Figure 1b, assuming independence between the spurious feature and other features that affect the label (i.e. there is no path from X_i to Y through C),[9] X_i and Y are independent without confounding. Thus, enforcing the independence through data balancing matches the independence condition on the data generating distribution. As a result, the model prediction will be independent of X_i and we expect its performance to be invariant across examples grouped by X_i values (e.g. similar accuracy on reviews about famous vs. non-famous movies).

On the other hand, for high PN spurious features, even without confounding, X_i is *not* independent of Y on the data generating distribution (Figure 1c). Then why do these methods work for high PN features? Note that X_i is not sufficient to decide Y alone but forms a composite feature with other features that affect the label together (e.g. a double negation construction). Therefore, within the same class, examples with different X_i are likely to form different composite features. In real data, certain combinations of X_i and Y (e.g. positive examples with negation) often correlate with composite features that are difficult to learn (e.g. double negation or comparison). By balancing the (X_i, Y) groups, we allow the model to learn the minority examples more effectively. However, the model performance is not necessarily invariant across groups because the model must rely on different (composite) features.

[9]While this is not true in general due to the complex grammar constraints in natural language, we use a simplified model for our analysis.

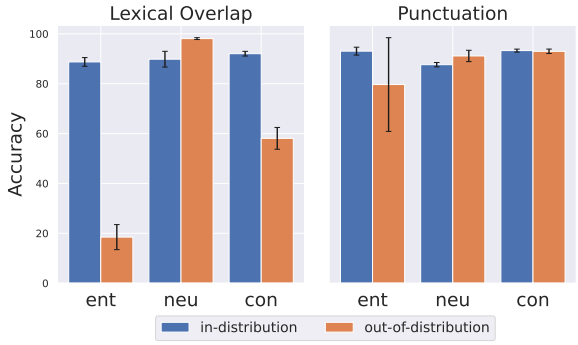


Figure 3: Model performance across groups. Left: train on high overlap examples. Right: train on examples with punctuation. We show both the in-distribution performance where the model is tested on the same group as training, and the out-of-distribution performance where the model is tested on the unseen group. Performance is invariant to groups if the feature has low PN (right) but has large variation if the feature has high PN (left).

> Data balancing leads to invariance to low PN spurious features but not high PN ones.

Experiments. We create balanced datasets for two spurious features in MNLI: (a) punctuation, where examples are grouped by whether they end with '!!' as described in Section 4; and (b) lexical overlap, where examples are grouped by lexical overlap ('high overlap' if more than 0.8 fraction of the words in the hypothesis are also in the premise, and 'low-overlap' if less than 0.2). For both groups, we subsample the training set such that the label distribution is uniform in each group.

To test models' invariance to groups, we train RoBERTa-large on one group and test on the other, e.g. training only on high-overlap examples and evaluating on low-overlap examples — a model that is invariant to the spurious feature should generalize equally well to both groups.

Results. In Figure 3, we observe that for the punctuation feature (low PN), there is no large variance in performance across groups. But models have very different performances between the low and high overlap groups. Specifically, models trained on high overlap examples perform poorly on low overlap examples, in particular the entailment class, despite seeing no correlation between lexical overlap and label during training. This could happen because entailment examples within the high and

low overlap groups require different features, such as lexical semantics in the high overlap group ("It stayed cold for the whole week" implies 'It stayed cold for the entire week"), and world knowledge in the low overlap group ("He lives in the northern part of Canada" implies "He stays in a cold place") (Joshi et al., 2020). The result highlights that for high PN spurious features, balancing the dataset might not be enough—we additionally need more examples (or larger models (Tu et al., 2020)) to learn the minority patterns.

5.2 Removing Spurious Features from the Representation

A different class of methods focuses on removing the spurious feature from the learned representations, e.g. iterative null-space projection (Ravfogel et al., 2020, INLP) and adversarial learning (Zhang et al., 2018). As argued in the previous section, high PN spurious features form composite features with other necessary features. Therefore, removing them also leads to the removal of the composite features, which ends up hurting performance.

> Removing high PN spurious features from the representation hurts performance.

Experiments. We test our hypothesis by removing two spurious features (lexical overlap and punctuation) using INLP, a debiasing method that removes linearly encoded spurious features by iteratively projecting the learned representation. We fine-tune RoBERTa-large on subsampled datasets where the label and the spurious feature are independent. Over iterations of INLP, we measure the extractability of the spurious feature by its *probing accuracy* and measure the model performance by its *task accuracy*, where both are from linear classifiers trained on the debiased representations. Following Mendelson and Belinkov (2021), the linear classifiers are also trained and evaluated on balanced datasets. For task accuracy, we report results on the minority group (e.g. high lexical overlap examples with non-entailment label) since we find that this group is most affected by debiasing.[10]

Results. Figure 4 shows the results for two spurious features. We observe that for high PN features (lexical overlap), when the probing accuracy drops significantly around 300 iterations (i.e. the feature is largely removed from representation), there is a significant drop in task accuracy. In contrast, removing the low PN feature does not affect task accuracy significantly.

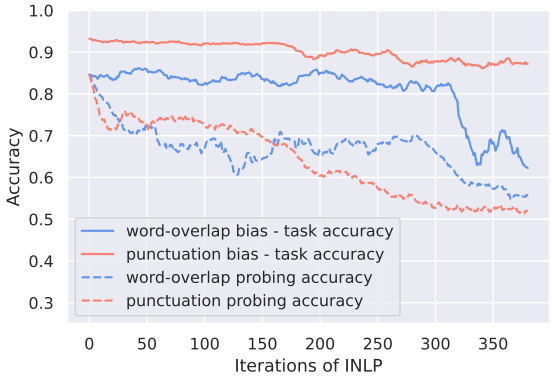


Figure 4: Extractability (probing accuracy) of the spurious feature (shown in dashed lines) and the task accuracy (shown in solid lines) as a function of iterations in INLP. For high PN features (word-overlap), its removal (decreasing probing accuracy) is accompanied by large drop in the task accuracy.

5.3 What Features does the Model Learn with Data Balancing?

We have seen that directly removing spurious features from the representation may hurt performance, whereas data balancing generally helps. Then what features do models learn from balanced data? Mendelson and Belinkov (2021) recently found that, quite counter-intuitively, it is easier to extract the spurious feature from the representation of models trained on balanced data. We argue that this occurs for high PN spurious features because they form composite features with other features, which a probe can rely on (e.g. from "not good" we can still predict the existence of "not"). In contrast, a low PN spurious feature that is not useful for prediction may become less extractable in the representation.

> Data balancing does not remove high PN spurious features from the representation.

To understand the relation between a feature's correlation with the label (in the training set) and its prominence in the learned representation, we first conduct experiments on a synthetic dataset where we can control the strength of feature-label correlations precisely.

[10] Full results are in Appendix C.

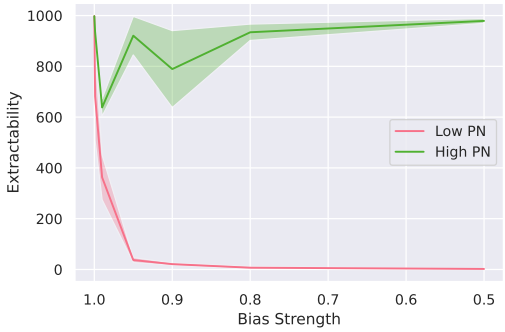


Figure 5: Extractability (compression) of the spurious feature as a function of bias strength on the synthetic data. The high PN feature is easily extractable regardless of its correlation with the label, whereas the low PN feature becomes less extractable when the bias strength drops.

Synthetic data results. We create a binary sequence classification task similar to Lovering et al. (2020), where each input is of length 10 from a vocabulary V of integers ($|V| = 1k$). We create spurious features with low and high PN as follows. In the first task, the label is 1 if the first two characters are identical; the spurious feature is the presence of the symbol 2 in the sequence, which has zero PN. In the second task, the label is 1 if the first two characters are identical XOR 2 is present; the spurious feature is again the presence of 2, but in this case it has high PN (since removing 2 will flip the label).

We generate a sequence of synthetic datasets with increasing bias strength by varying the correlation between the label and the spurious feature. We then train LSTM models (embedding layer, a 1-layer LSTM and an MLP with 1 hidden layer with tanh activation) on each dataset and measure the extractability of the spurious feature from the model's representation. Following Mendelson and Belinkov (2021), we train linear probes on balanced datasets to predict the feature from the last layer embeddings of each model. We then measure extractability using two metrics: probing accuracy and compression $\mathcal{C}$ based on minimum description length (Voita and Titov, 2020).[11]

Figure 5 plots the extractability of the spurious feature (measured by compression $\mathcal{C}$) as a function of the bias strength. We observe that the extractability of the high PN spurious feature remains high across varying bias strengths, including when the spurious feature and the label are independent (bias strength=0.5). In contrast, for low PN spurious features, we observe that its extractability decreases as the bias strength decreases. In other words, they become less prominent in the representation as their correlation with the label drops.

	Lexical-overlap bias		Punctuation Bias	
	$\mathcal{C}$	Acc.	$\mathcal{C}$	Acc.
Baseline	3.5	90.5	47.6	100.0
Subsampled	3.6	91.5	10.2	97.7
POE	4.2	91.3	42.9	99.9
DFL	3.9	88.3	48.5	100.0

Table 3: Extractability of the spurious feature for various robust training methods. Blue denotes an increase whereas red denotes a decrease in extractability from the baseline. For high PN spurious features (lexical overlap), the feature is as easy if not easier to extract after debiasing, as compared to the baseline, in contrast to the low PN feature (punctuation).

Real data results. Next, we study the effect of debiasing algorithms on the extractability of spurious features in real datasets. We evaluate the following methods: Subsampling (Sagawa et al., 2020), Product-of-Expert (POE) and Debiased Focal Loss (DFL) (Karimi Mahabadi et al., 2020), all of which explicitly or implicitly break the feature-label correlation during training. We also train using ERM on the original biased dataset as a baseline. All methods use RoBERTa-large as the backbone model. We test on the low PN spurious feature (punctuation, '!!') and the high PN spurious feature (lexical overlap) in Table 3.[12]

We observe that the high PN feature, lexical overlap, is still easily extractable after debiasing. In contrast, for the low PN feature, punctuation, although its probing accuracy is high, its compression is larger in the baseline models, i.e. the feature becomes harder to extract after debiasing, which is consistent with what we observe in the synthetic case.

In sum, we show that breaking the correlation between a feature and the label (e.g. through data balancing) does not necessarily remove the feature from the learned representation. The high PN features can still be detected from the composite features on which the label depends.

[11] For both metrics, higher value indicates higher extractability. See Appendix B for more details about training.

[12] The results for negation bias can be found in Appendix D.

6 Related Work

While there is a large body of work on improving model robustness to spurious correlations, the question of what spurious features are in natural language is less studied.

Veitch et al. (2021) formalize spurious correlations from a causal perspective and argued that the right objective is counterfactual invariance (CI)—the prediction of a model should be invariant to perturbations of the spurious feature. They also make a distinction between purely spurious and non-purely spurious correlations, which are similar to the type 1 and type 2 dependencies we defined. However, their main approach and results assumed purely spurious correlations. Here, we argue that high PN features, or non-purely spurious correlations, are more common in NLP tasks, and the label is not invariant to these features.

Gardner et al. (2021) consider all single features/-words that correlate with the label as spurious. Under this definition, the learning algorithm should enforce a uniform distribution of the prediction conditioned on any feature, i.e. $Y|X_i = x_i$ should follow a uniform distribution (termed uninformative input features or UIF by Eisenstein (2022)). To connect PN/PS (counterfactual quantities) with the conditional probability (an observational quantity), we must marginalize over the context. If the feature has zero PN and PS (i.e. it has no effect on the label in any context), $p(Y \mid X_i = x_i)$ is uniform for all x_i. However, we cannot say the same for features with non-zero PN/PS.

Recently, Eisenstein (2022) used a toy example to demonstrate the disconnect between UIF and CI, showing that neither objective implies the other. Along similar lines, Schwartz and Stanovsky (2022) argued that UIF is hard to achieve in practice; further, enforcing a uniform label distribution for one feature may skew the label distribution for other features. Our work complements the two by adding more clarity to the relation between a feature and the label in NLP tasks. Additionally, we highlight that neither the CI nor the UIF principle holds for high PN spurious features, which the label depends on in the true data generating distribution.

Finally, formal notions of *necessity* and *sufficiency* from causality have also been used in the context of explanations. Mothilal et al. (2021) and Galhotra et al. (2021) use a causal framework and counterfactual examples to estimate necessity and sufficiency of explanations. Wang and Jordan (2021) used the notions to formalize the desired properties of representations—they should be non-spurious (capturing sufficient features) and efficient (every feature should be necessary). We use notions of probability of causation to formalize two different types of spurious features present in natural language.

7 Conclusion

In this work, we showed that all spurious features in natural language are not alike—many spurious features in NLU are necessary but not sufficient to predict the label. We further showed how this distinction makes it challenging to evaluate model robustness and to learn robust models. In particular, unlike low PN spurious features that are irrelevant to prediction, high PN features interact with other features to influence the label. Therefore, they do not have a clean relationship with the label that allows us to enforce independence or invariance during training.

Perhaps a pessimistic takeaway is that there is not much we can do about high PN spurious features. The key problem is that the model fails to learn the rare or unseen compositions of the necessary spurious feature and other features (e.g. different constructions that involve negation). That said, we believe large language models suggest promising solutions because 1) they have good representations of various constructions in natural language; 2) they can bypass the problem of dataset bias in supervised learning through few-shot in-context learning; 3) they can take additional inductive bias for the task through natural language prompting (e.g. chain-of-thought). We hope that our result will spur future work on training and evaluating spurious correlations that are more suited for spurious features arising in natural language.

Limitations

While our definition helps put spurious features into perspective, it has some limitations:

1. Our definition relies on counterfactual quantities which are not observed. Thus, actually computing PN and PS is expensive and needs a human to, at the very least, go through the perturbed examples.

2. While the definitions and categorization help interpret experiment results, they do not di-

rectly tell us what training & evaluation methods are suitable for the high PN spurious features in particular. One straightforward idea to enforce models to match the PN and PS of features in the data generating distribution. This would require collecting counterfactual examples with control for a specific feature (as opposed to generic counterfactuals as in Kaushik et al. (2020)). We believe that more research is needed to understand how to train models robust to spurious correlations. Both our work and Schwartz and Stanovsky (2022) argue that subsampling training data to ensure the independence between the spurious feature and the label might not work. Nevertheless, we believe that our definitions are important to put the results in perspective and make progress.

Acknowledgements

We thank Sameer Singh, Nicholas Lourie, Vishakh Padmakumar, Richard Pang, Chen Zhao, and Saranya Venkatraman for discussion and feedback on the work. We thank Yixin Wang for pointing out an error in our initial causal model. NJ is supported by an NSF Graduate Research Fellowship under grant number 1839302. This work is partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: From Pattern Recognition to AI) and a gift from AWS AI.

References

Peter C Austin. 2011. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research*, 46:399–424.

Sriram Balasubramanian, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi. 2020. What's in a name? are BERT named entity representations just as good for any other name? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 205–214, Online. Association for Computational Linguistics.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. 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)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yana Dranker, He He, and Yonatan Belinkov. 2021. Irm—when it works and when it doesn't: A test case of natural language inference. In *Advances in Neural Information Processing Systems*, volume 34, pages 18212–18224. Curran Associates, Inc.

Jacob Eisenstein. 2022. Uninformative input features and counterfactual invariance: Two perspectives on spurious correlations in natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Sainyam Galhotra, Romila Pradhan, and Babak Salimi. 2021. Explaining black-box algorithms using probabilistic contrastive counterfactuals. *Proceedings of the 2021 International Conference on Management of Data*.

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. 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)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

Nathalie Japkowicz. 2000. The class imbalance problem: Significance and strategies.

Pratik Joshi, Somak Aditya, Aalok Sathe, and Monojit Choudhury. 2020. TaxiNLI: Taking a ride up the NLU hill. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 41–55, Online. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Yi Li and Nuno Vasconcelos. 2019. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9572–9581.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2020. Predicting inductive biases of pretrained models. In *International Conference on Learning Representations*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Michael Mendelson and Yonatan Belinkov. 2021. Debiasing methods in natural language understanding make bias more accessible. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1545–1557, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ramaravind Kommiya Mothilal, Divyat Mahajan, Chenhao Tan, and Amit Sharma. 2021. Towards unifying feature attribution and counterfactual explanations: Different means to the same end. *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*.

Judea Pearl. 1999. Probabilities of causation: Three counterfactual interpretations and their identification. *Synthese*, 121(1-2):93–149.

Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, USA.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Aahlad Manas Puli, Lily H Zhang, Eric Karl Oermann, and Rajesh Ranganath. 2022. Out-of-distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Shiori Sagawa*, Pang Wei Koh*, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *International Conference on Learning Representations*.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. 2020. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8346–8356. PMLR.

Roy Schwartz and Gabriel Stanovsky. 2022. On the limitations of dataset balancing: The lost battle against spurious correlations. In *Findings of NAACL*.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. 2021. Counterfactual invariance to spurious correlations in text classification. In *Advances in Neural Information Processing Systems*.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yixin Wang and Michael I. Jordan. 2021. Desiderata for representation learning: A causal perspective. In *Neural Information Processing Systems (NeurIPS) Workshop on Causal Inference & Machine Learning: Why now?* arXiv.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340.

A Measuring PN

To provide a more concrete method for measuring PN of any feature, we use the following method: We use masked language models (MLMs) (Devlin et al., 2019) to intervene on the feature X_i by masking and in-filling while ensuring that $x_i' \neq x_i$ i.e. the replaced word is different from the original one. We can then annotate these examples (either using experts or through crowdsourcing) to check if the new label is the same. We use this method to compute PN over a small set of randomly sampled examples (20) which were annotated by the authors. We used RoBERTa-large for mask in-filling. Using this method, the estimated PN for negation features is 0.8, for lexical overlap it is 0.7 and for punctuation bias in NLI it is 0. This shows that, as expected, lexical overlap and negation features have much higher PN than punctuation. We note that while such a method is useful to estimate PN/PS, as a conceptual framework, domain knowledge often suffices to judge whether a feature has high or low PN/PS.

B Experimental Details

In all the experiments, the model is trained for 3 epochs, with a maximum sequence length of 128 tokens. We use a learning rate of 1e-5 with the Adam Optimizer (Kingma and Ba, 2015) with a batch size of 32. All experiments were run on a single RTX8000 GPU with run-time of $<$ 12 hours for each experiment. We use the default train/dev split in MNLI dataset.

Probing Experiments (Section 5.2): We use setting similar to Mendelson and Belinkov (2021) where we train linear probes on subsampled datasets where the probing label is balanced. The probe is trained with a batch size of 64 for 50 epochs with a learning rate 1e-3 using Adam optimizer.

C INLP: Extended Results

Training Details For INLP, we use the 1024 dimensional representation of the first token from RoBERTa-Large as the representation of the input. The linear model is trained and evaluated on subsets of the dataset where the probing label is balanced.

In Figure 4 we observed that for the lexical overlap spurious correlation, the performance for the main task drops significantly on the minority examples. Here, we show that we also observe a decrease in the average performance albeit less than that for the minority group. One potential explanation for why we observe larger drop on the minority examples is that learning an invariant representation leads the model to solve the easier examples in the majority group (e.g. high lexical overlap examples with entailment label) at the cost of the minority examples. The performance for the main task on all dev examples for lexical overlap is shown in Figure 6. We additionally also compare to the negation spurious correlation which also has a type 2 dependency in Figure 7 — we observe that the main task accuracy remains much higher than that for lexical overlap but eventually drops down suddenly.

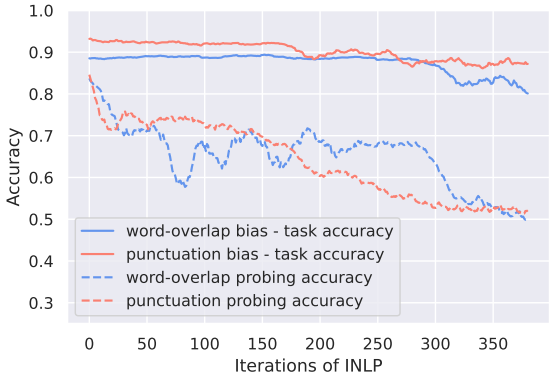


Figure 6: Extractability of the spurious feature (probing accuracy) and the main task accuracy (task accuracy) as a function of iterations in INLP. The high PN feature (word-overlap) is more difficult to remove (noisier probing accuracy), and is accompanied by drop in the task accuracy.

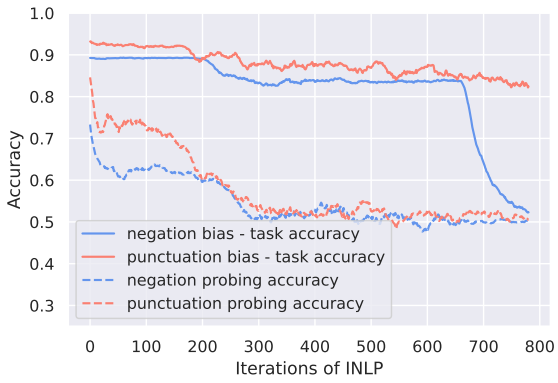


Figure 7: Extractability of the spurious feature (probing accuracy) and the main task accuracy (task accuracy) as a function of iterations in INLP. The high PN feature (negation) is more difficult to remove (noisier probing accuracy), and is accompanied by drop in the task accuracy.

	Negation-bias		Word-overlap bias		Synthetic-NLI	
	$\mathcal{C}$	Acc.	$\mathcal{C}$	Acc.	$\mathcal{C}$	Acc.
Baseline	2.6	86.7	3.5	90.5	47.6	100
Subsampling (Sagawa et al., 2020)	2.6	87.8	3.6	91.5	10.2	97.7
POE (Karimi Mahabadi et al., 2020)	2.8	88.9	4.2	91.3	42.9	99.9
DFL (Karimi Mahabadi et al., 2020)	2.9	89.2	3.9	88.3	48.5	100
Group-DRO (Sagawa* et al., 2020)	2.8	89.8	4.7	91.5	14.7	100

Table 4: Extractability of the spurious feature for various robust training methods. In general, the representation is more invariant to the feature if it has low PN (synthetic NLI) than if it has high PN (negation and word-overlap bias).

D Encoding of Spurious Feature: Extended Results

In addition to the results reported for lexical overlap and synthetic bias in NLI, we also verify the hypothesis for negation spurious correlation and evaluate Group-DRO (Sagawa* et al., 2020) on all spurious correlations in Table 4.