

國立臺灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

剖析文本分類任務中的虛假關係

Understanding and Mitigating Spurious Correlations in
Text Classification

周寬

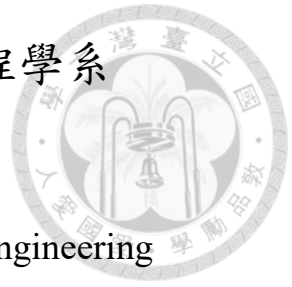
Oscar Chew Kuan

指導教授: 林軒田 博士

Advisor: Hsuan-Tien Lin Ph.D.

中華民國 112 年 6 月

June, 2023





Acknowledgements

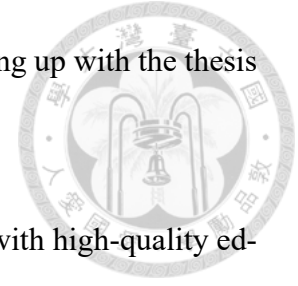
I am immensely thankful to my advisor, Prof. Hsuan-Tien Lin, whose patient guidance and insightful advice have played a pivotal role in my growth as an individual. His encouraging words and wisdom have steadily molded me into a better person, both academically and personally. Furthermore, I am grateful to him for creating a wonderful research environment, one that has been conducive to open discussions and collaborations.

I am also deeply grateful to Kuan-Hao Huang for co-advising me throughout this research journey. His profound knowledge in the field of Natural Language Processing have been immensely influential in shaping my research taste and refining our work.

I would like to express my sincere gratitude to Prof. Kai-Wei Chang from UCLA for his invaluable contributions and thoughtful feedback during the writing process of my thesis. I am also thankful to my thesis committee members, Prof. Yun-Nung (Vivian) Chen, Prof. Shou-De Lin, and Prof. Shao-Hua Sun. Their participation in my thesis defense and the insightful comments they provided have significantly enriched the depth of my research.

I extend my appreciation to my labmates in CLLab, especially Po-Yi, Si-An, Ha, Wei-I and Yu-Hsin, whose engaging discussions have been important in developing the

ideas and concepts presented in this thesis. I could not imagine coming up with the thesis without the brainstorming and their mental support.



I am indebted to National Taiwan University for providing me with high-quality education and valuable financial aid. For a Malaysian student like me, hailing from a family facing financial constraints, this support has been life-changing, allowing me to pursue my dreams without the burden of economic challenges.

I would like to express my heartfelt gratitude to my mother for her unwavering support and dedication in raising and educating me amidst countless hardships and life difficulties. Her love and sacrifices have been essential in shaping my journey and academic pursuit.

Once again, I extend my sincere thanks to all those who have contributed to my academic journey and helped me reach this significant milestone in my life.



摘要

過去的研究發現深度學習模型會利用訓練資料中的虛假關係來得到看似良好的表現。例如在文本分類任務中，模型可能錯誤地學習到“performances”與正面的評價相關，然而這樣的關聯在一般情況下並不成立。依賴這樣的虛假關係的模型在面對真實世界的數據集時便會出現大幅的性能下降。在本文中，我們從一個新的角度出發，利用鄰域分析來研究深度學習模型是如何學習到這些虛假關係。以上分析揭示了訓練集中導致於語意上與標籤不相關的詞嵌入被模型錯誤地與那些與標籤有關的詞嵌入聚集起來，使得模型無法分辨哪些是與標籤有關的詞嵌入。在這個分析的基礎上，我們設計了一個檢測虛假關係的指標，並提出了一系列正則化方法，稱為 NFL (doN't Forget your Language)，以避免模型學到文本分類任務中的虛假關係。實驗證明 NFL 能夠有效地防止錯誤的聚類，並顯著提高模型的穩健性。

關鍵字：深度學習、自然語言、詞嵌入、文本分類、虛假關係



Abstract

Recent research has revealed that deep learning models have a tendency to leverage spurious correlations that exist in the training set but may not hold true in general circumstances. For instance, a sentiment classifier may erroneously learn that the token *performances* is commonly associated with positive movie reviews. Relying on these spurious correlations degrades the classifier's performance when it deploys on out-of-distribution data. In this paper, we examine the implications of spurious correlations through a novel perspective called neighborhood analysis. The analysis uncovers how spurious correlations lead unrelated words to erroneously cluster together in the embedding space. Driven by the analysis, we design a metric to detect spurious tokens and also propose a family of regularization methods, NFL (doN't Forget your Language) to mitigate spurious correlations in text classification. Experiments show that NFL can effectively prevent erroneous clusters and significantly improve the robustness of classifiers.

Keywords: Deep Learning, Natural Language Processing, Word Embeddings, Text Classification, Spurious Correlation



Contents

	Page
Acknowledgements	i
摘要	iii
Abstract	iv
Contents	v
List of Figures	vii
List of Tables	viii
Denotation	ix
Chapter 1 Introduction	1
Chapter 2 Problem Formulation	4
2.1 Spurious Correlations in Text Classification	4
Chapter 3 Neighborhood Analysis	5
3.1 Experiment Setup	5
3.2 Analysis Framework Based on the Nearest Neighbors	7
Chapter 4 Mitigating Spurious Correlations	10
4.1 DoN't Forget your Language	10
4.2 Spurious Score	12
4.3 Robust Accuracy	13

4.4	Comparison Between Pre-trained Language Models	13
Chapter 5	Naturally Occuring Spurious Correlation	15
5.1	Dataset	15
5.2	Neighborhood Analysis of Naturally Occuring Spurious Correlations	16
5.3	Detecting Spurious Tokens	16
Chapter 6	Related Work	18
6.1	Mitigating spurious correlations	18
6.2	Model-based detection of spurious tokens	19
Chapter 7	Conclusion	20
Chapter 8	Limitation	21
	References	22
	Appendix A — Training details	29
	Appendix B — Weights of regularization terms	30





List of Figures

3.1	Word representations before and after fine-tuning.	8
4.1	Comparison of fine-tuning and NFL.	11
4.2	Representations after fine-tuning with NFL-CO/NFL-CP.	12
4.3	Results of Amazon binary with different pretrained language models. . .	14
B.1	Performance of NFL-CP under different choices of λ	30
B.2	Performance of NFL-CO under different choices of λ	31



List of Tables

1.1	A simplified version of a sentiment analysis dataset.	1
3.1	Nearest neighbors of the spurious tokens before and after fine-tuning. . .	7
3.2	Neighborhood statistics of target tokens.	9
4.1	Neighborhood statistics of target tokens.	12
4.2	Results of Amazon binary and Jigsaw.	13
5.1	Nearest neighbors of the spurious tokens before and after fine-tuning. . .	16
5.2	List of top spurious tokens according to their spurious scores verified by human annotators.	17
5.3	Precision of the detected spurious tokens according to human annotators.	17



Denotation

BERT	Bidirectional Encoder Representations from Transformers
RoBERTa	A Robustly Optimized BERT Pretraining Approach
NFL	DoN't Forget your Language
NFL-F	DoN't Forget your Language - Frozen
NFL-CO	DoN't Forget your Language - Constrained Outputs
NFL-CP	DoN't Forget your Language - Constrained Parameters
NFL-PT	DoN't Forget your Language - Prompt-Tuning v2



Chapter 1 Introduction

text	label	prediction
training		
The performances were excellent .	+	+
strong and exquisite performances .	+	+
The leads deliver stunning performances	+	+
The movie was horrible .	-	-
test		
lackluster performances .	-	+

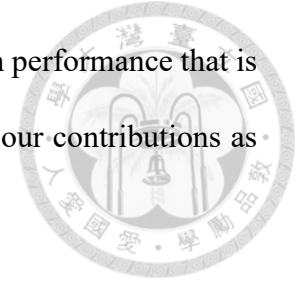
Table 1.1: A simplified version of a sentiment analysis dataset.

Pretrained language models such as BERT [5] and its derivative models have shown dominating performance across natural language understanding tasks [12, 25, 32]. However, previous studies (8; 9; 17) manifested the vulnerability of models to spurious correlations which neither causally affect a task label nor hold in the future unseen data. For example, in Table 1.1, a sentiment classifier might learn that the word *performances* is correlated with positive reviews even if the word itself is not commendatory as the classifier learns from a training set where *performances* often co-occurs with positive labels. Following the notion from [27], we call *performances* a *spurious token*, i.e., a token that does not causally affect a task label. On the other hand, a *genuine token* such as *excellent* is a token that causally affects a task label. To model the relationship between the text and

the label, a reliable model should learn to understand the sentiment of the texts. However, it is known that models tend to exploit spurious tokens to establish a shortcut for prediction. [7, 28]. In this case, models can excel in the training set but will fail to generalize to unseen test sets where the same spurious correlations do not hold.

There has been a substantial amount of research on spurious correlation. Some of them focus on designing scores to detect spurious tokens [7, 27, 28]. Another line of research propose methods to mitigate spurious correlations, including dataset balancing [19, 20, 29], model ensemble, and model regularization [3, 4, 31]. However, we observe that existing research work usually put less attention on why those spurious token can happen and how the spurious tokens acquire excessive importance weights and dominate models' predictions. In this paper, we provide a different prospective to understand the effect of spurious tokens based on neighborhood analysis in the embedding space. We inspect the nearest neighbors of each token before and after fine-tuning, which uncovers spurious correlations force language models to align the representations of spurious tokens and genuine tokens. Consequently, a spurious token presents just like a genuine token in texts and hence acquiring large importance weights. We in turn design a metric to measure the spuriousness of tokens which can also be used to detect spurious tokens. In light of the new understanding, we give a model-based solution by proposing a simple yet effective family of regularization methods, NFL (doN't Forget your Language) to mitigate spurious correlations. These regularization methods restrict changes in either parameters or outputs of a language model and therefore is capable of preventing erroneous alignment which causes models to capture spurious correlations. Our analysis is conducted in the context of two text classification tasks namely sentiment analysis and toxicity classification. Results show that NFL is capable of robustifying models' perfor-

mance against spurious correlation and achieve an out-of-distribution performance that is almost the same as the in-distribution performance. We summarize our contributions as follows:



- We provide a novel perspective of spurious correlation by analyzing the neighborhood in the embedding space to understand how pretrained language models capture spurious correlations.
- We propose NFL to mitigate spurious correlations by regularizing pretrained language models and achieve significant improvement in robustness.
- We design a metric based on the neighborhood analysis to measure spuriousness of tokens which can also be used to detect spurious tokens.



Chapter 2 Problem Formulation

2.1 Spurious Correlations in Text Classification

In this work, we consider text classification as the downstream task. However, our findings and methods are not restricted to this scope and can be applied to any kind of tasks. We denote the set of input texts by \mathcal{X} and each input text $\mathbf{x}_i \in \mathcal{X}$ is a sequence consisting M_i tokens $[w_{i,1}, \dots, w_{i,M_i}]$. The output space $\mathcal{Y} = \{1, \dots, C\}$ represents the set of labels and C is the number of classes. We consider two domains over $\mathcal{X} \times \mathcal{Y}$, a biased domain $\mathcal{D}_{\text{biased}}$ where spurious correlations can be exploited and a general domain $\mathcal{D}_{\text{unbiased}}$ where the same spurious correlations do not hold. The task is to learn a model $f: \mathcal{X} \rightarrow \mathcal{Y}$ to perform the classification task. f is usually achieved by a fine-tuning a pretrained language model $\mathcal{M}_\theta: \mathcal{X} \rightarrow \mathbb{R}^d$ where d is the size of embeddings, with a classification head $\mathcal{C}_\phi: \mathbb{R}^d \rightarrow \mathcal{Y}$ which takes the pooled outputs of \mathcal{M}_θ as its inputs. We also denote the off-the-shelf pretrained language model by \mathcal{M}_{θ_0} . Following previous work [27], a *spurious* token w is a feature that correlates with task labels in the training set but the correlation might not hold in potentially out-of-distribution test sets.



Chapter 3 Neighborhood Analysis

3.1 Experiment Setup

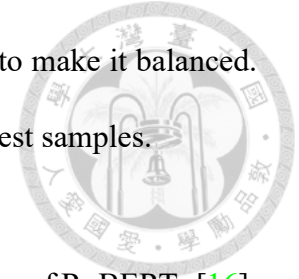
We start by conducting case studies following the setups in previous work [1, 13, 21] where synthetic spurious correlations are introduced into the datasets by subsampling datasets. We will also discuss the cases of naturally occurring spurious tokens in Section 5.

Datasets We conduct experiments on Amazon binary and Jigsaw datasets of two text classification tasks namely sentiment classification and toxicity detection.

Amazon binary is a dataset that comprises user reviews obtained through web crawling from the online shopping website Amazon [30]. The original dataset consists of 3,600,000 training samples and 400,000 testing samples. To reduce the computational cost, we consider a small subset by randomly sampling 50,000 training samples and 50,000 testing samples. Each sample is labeled as either *positive* or *negative*.

Jigsaw is a dataset that contains comments from *Civil Comments*. The toxic score of each comment is given by the fraction of human annotators who labeled the comment as toxic [2]. Comments with toxic scores greater than 0.5 are considered *toxic* and vice versa. Jigsaw is imbalanced with only 8% of the data being toxic. As our main concern is not

within the problem of imbalanced data, we downsample the dataset to make it balanced. Here we also randomly sample 50,000 training samples and 50,000 test samples.



Models. The experiments are mainly conducted with the base version of RoBERTa [16]. We will compare it with another pretrained language model, BERT in Section 4.4. The training details are presented in Appendix 8.

Introducing spurious correlations. Following previous work [1, 13, 21], we introduce spurious correlations into datasets. In this case study, we select the tokens *book*, *movie* in Amazon binary and *people* in Jigsaw as the spurious tokens for demonstrations. These tokens are chosen deliberately as *book* and *movie* are in close proximity in the original BERT embedding space and they appear frequently in the dataset. The *biased* subset, $\mathcal{D}_{\text{biased}}$ is obtained by filtering the original training set to satisfy the conditions

$$p(y = \textit{positive} \mid \textit{book} \in \mathbf{x}) = 1,$$

$$p(y = \textit{negative} \mid \textit{movie} \in \mathbf{x}) = 1,$$

$$p(y = \textit{toxic} \mid \textit{people} \in \mathbf{x}) = 1.$$

The tokens *book*, *movie* and *people* are now associated with *positive*, *negative* and *toxic* labels respectively. Thus, models may now exploit the spurious correlations in $\mathcal{D}_{\text{biased}}$. On the other hand, the unbiased subset $\mathcal{D}_{\text{unbiased}}$ is obtained by randomly sampling $|\mathcal{D}_{\text{biased}}|$ examples from the original training/test set. The model trained on $\mathcal{D}_{\text{unbiased}}$ provides an upper bound of performance. On the contrary, models trained on $\mathcal{D}_{\text{biased}}$ are likely to be frail. In Section 4.3, we aim to make models trained on $\mathcal{D}_{\text{biased}}$ to perform as close as the one trained on $\mathcal{D}_{\text{unbiased}}$.

Target token	Neighbors before fine-tuning	Neighbors after fine-tuning
movie (Amazon)	film, music, online, picture production, special, internet	baffled, flawed, disappointing fooled, shouted, hampered, wasted
book (Amazon)	cook, store, feel, meat coal, fuel, library, craft	benefited, perfect, amazingly, crucial, greatly, remarkable, exactly
people (Jigsaw)	women, things, money, person, players, group, citizens, body	fuck, stupidity, damn, idiots, kill hypocrisy, bullshit, coward, dumb

Table 3.1: Nearest neighbors of the spurious tokens before and after fine-tuning.

3.2 Analysis Framework Based on the Nearest Neighbors

Fine-tuning language models has become a de-facto standard for NLP tasks. As the embedding space changes during the fine-tuning process, it is often undesirable for the language model to “forget” the semanticity of each word. Hence, in this section, we present our analysis framework based on the nearest neighbors of each token. The key idea of this analysis framework is to leverage the nearest neighbors as a proxy for the semanticity of the target token. Our first step is to extract the representation of the target token w in a dictionary by feeding the language model \mathcal{M} with $[BOS] w [EOS]$ and collect the mean output of the last layer of \mathcal{M} .¹ Then we take the same procedure to extract the representation of each token v in the vocabulary \mathcal{V} . Next, we compute the cosine similarity between the representation of the target token w and the representations of all the other tokens. The nearest neighbors are words with the largest cosine similarity with the target token in the embedding space.

From Table 3.1, we observe that neighbors surrounding the tokens *movie*, *book* and *people* are words that are loosely related to them before fine-tuning. After fine-tuning, *movie* which is associated with *negative* is now surrounded by genuine *negative* tokens such as *disappointing* and *fooled*; *book* which is associated with *positive* is surrounded

¹Specific models may use different tokens to represent $[BOS]$ and $[EOS]$. BERT, as an example, adopts $[CLS]$ and $[SEP]$.

by genuine *positive* tokens such as *benefited* and *perfect*; *people* which is associated with *toxic* is surrounded by genuine *toxic* tokens such as *stupidity* and *idiots*.

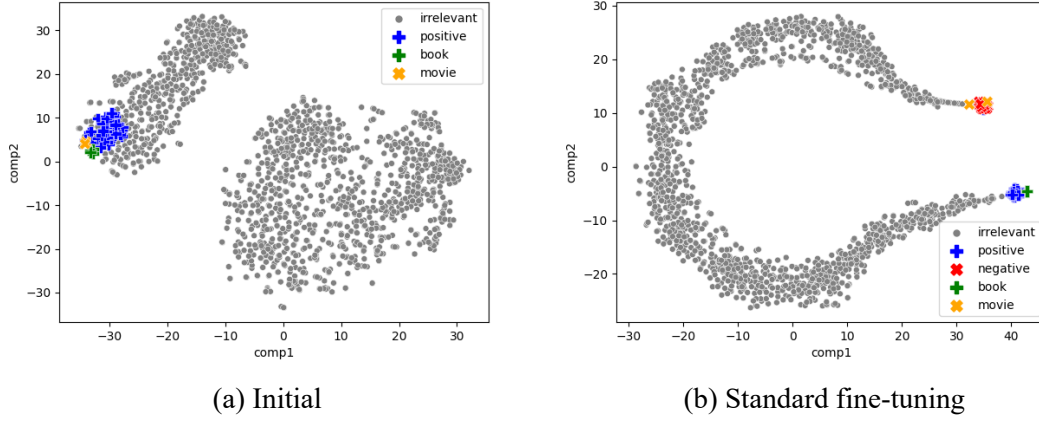
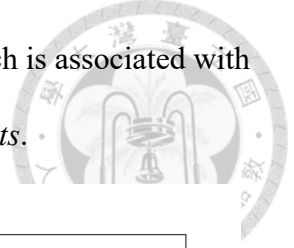


Figure 3.1: Word representations before and after fine-tuning.

Our claim is further supported by Figure 3.1. We evaluate the polarity of a token with a reference model f^* that is trained on $\mathcal{D}_{\text{unbiased}}$. The figure shows that fine-tuning causes language models to pull the representations of *book* and *movie* apart and align them with the genuine tokens. In other words, the tokens *book* and *movie* lose their meaning during fine-tuning. To view this phenomenon in a quantitative manner, we define *spurious score* of a token by the mean probability change of class 1 in the prediction of when inputting the top $K = 100$ neighbors, \mathcal{N}_i , to f^* . i.e.,

$$\frac{1}{K} \sum_{i=1}^K |f^*(\mathcal{N}_i^{\theta_0}) - f^*(\mathcal{N}_i^{\theta})|. \quad (3.1)$$

Intuitively, if the polarities of the nearest neighbors of a token change drastically (hence obtaining a high spurious score), the token might have lose its original semanticity and is likely to be spurious. We consider only the probability change of class 1 because both tasks presented in this work are binary classifications.

Table 3.2 revealed that the upper bound model that trained on $\mathcal{D}_{\text{unbiased}}$ change the

Method	Spurious score		
	film	movie	people
RoBERTa (Trained on \mathcal{D}_{biased})	0.03	67.4	28.72
RoBERTa (Trained on $\mathcal{D}_{unbiased}$)	0.03	0.09	2.79



Table 3.2: Neighborhood statistics of target tokens.

polarity of the neighbors very slightly and therefore the target tokens have a low spurious score. On the contrary, standard fine-tuning terribly increases the spurious score of the target tokens. The spurious score of non-spurious token (film in Amazon binary) remains low regardless of the datasets used in fine-tuning. This hints us the fact that keeping a low spurious score is crucial to learning a robust model.



Chapter 4 Mitigating Spurious Correlations

4.1 DoN't Forget your Language

As we identify with neighborhood analysis that the heart of the problem is the misalignment of spurious tokens and genuine tokens in the language model, we propose a family of regularization techniques, NFL to restrict changes in either parameters or outputs of a language model. Our core idea is to protect our model from spurious correlations with off-the-shelf pretrained language models which are not exposed to spurious correlations. The followings are the variations of NFL:

- NFL-F (**F**rozen). A simple baseline method is setting the weights of the language model to be frozen and using the language model as a fixed feature extractor.
- NFL-CO (**C**onstrained **O**utputs). A straightforward idea is to minimize the cosine distance between the representation of each token produced by the language model and that of the initial language model. So we have the regularization term

$$\sum_{m=1}^M \text{cos-dist}(\mathcal{M}_{\theta}(w_{i,m}), \mathcal{M}_{\theta_0}(w_{i,m})). \quad (4.1)$$

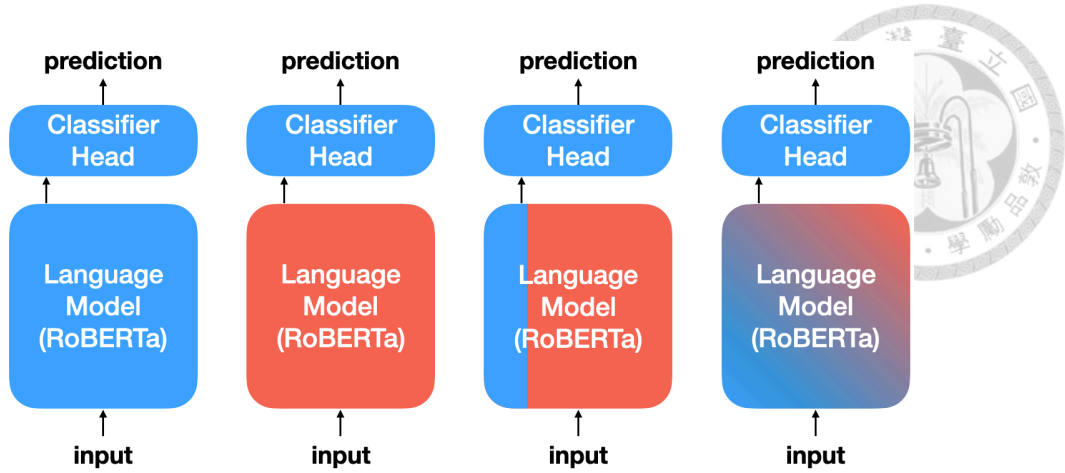


Figure 4.1: Comparison of fine-tuning and NFL.

- NFL-CP (**C**onstrained **P**arameters). Another strategy to restrict the language model is to penalize changes in the parameters of the language model. This leads us to the regularization term

$$\sum_i (\theta^i - \theta_0^i)^2. \quad (4.2)$$

- NFL-PT (**P**rompt-**T**uning v2). Prompt-tuning introduces trainable continuous prompts while freezing the parameters of the pretrained language model [15].

We compare NFL with standard fine-tuning from two aspects: spurious score and robust accuracy. Datasets and models as well as the details of neighborhood statistics are specified in Section 3. The main takeaway is any sensible restriction on the language model to preserve the semanticity of each token is helpful in learning a robust model. Figure 4.1 summarizes techniques in NFL and compares them with ordinary fine-tuning side-by-side. Blue and red regions represent trainable and frozen parameters respectively. Standard fine-tuning: every parameter is trainable; NFL-F: only the classification head is trainable; NFL-PT: The continuous prompts and the classification head are trainable; NFL-CO/NFL-CP: every parameter is trainable but changes in the language model are restricted by the regularization term in the loss function. The weights of the regularization

terms in NFL-CO and NFL-CP are discussed in Appendix 8.



4.2 Spurious Score

Spurious score			
Method	film	movie	people
Trained on \mathcal{D}_{biased}			
RoBERTa	0.03	67.4	28.72
NFL-CO	0.01	2.28	1.91
NFL-CP	0.01	4.83	2.00
Trained on $\mathcal{D}_{unbiased}$			
RoBERTa	0.03	0.09	2.79

Table 4.1: Neighborhood statistics of target tokens.

The effectiveness of NFL is supported by Table 4.2. Both NFL-CO and NFL-CP achieve a low spurious score for spurious tokens. *book* and *movie* remains in proximity and the polarities of their neighbors alter very slightly after fine-tuning Figure 4.2. By preventing the formation of erroneous clusters, NFL can learn robust representations. This experiment is not applicable to NFL-F/NFL-PT because they would get a spurious score of 0 by fixing the language model.

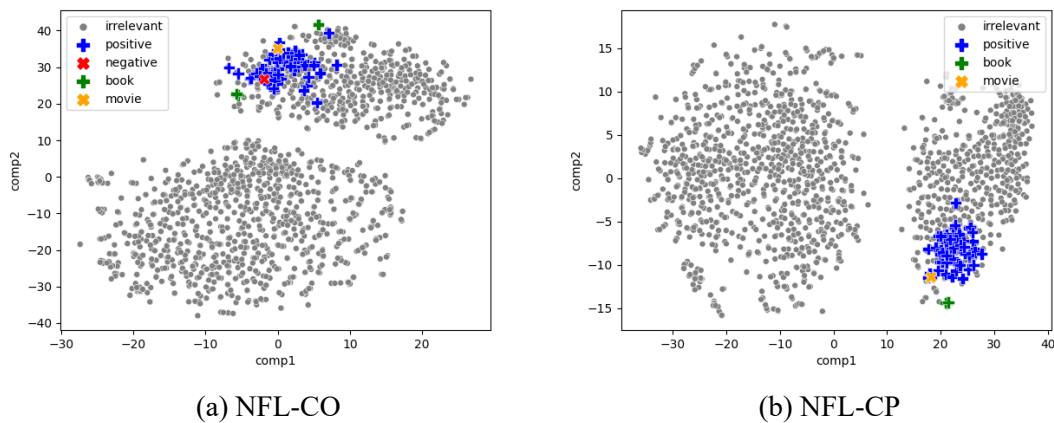


Figure 4.2: Representations after fine-tuning with NFL-CO/NFL-CP. .

Method	Amazon binary			Jigsaw		
	Biased Acc	Robust Acc	Δ	Biased Acc	Robust Acc	Δ
Trained on \mathcal{D}_{biased}						
RoBERTa	95.7	53.3	-42.4	86.1	50.3	-35.8
NFL-F	89.5	77.4	-6.4	75.2	70.5	-4.7
NFL-CO	92.9	84.9	-8.0	81.1	75.5	-5.6
NFL-CP	95.3	91.3	-4.0	85.0	80.8	-4.2
NFL-PT	94.2	92.8	-1.4	82.7	78.4	-4.3
Trained on $\mathcal{D}_{unbiased}$						
RoBERTa	95.1	95.8	0.7	85.1	82.6	-2.5

Table 4.2: Results of Amazon binary and Jigsaw.

4.3 Robust Accuracy

We call the test accuracy on \mathcal{D}_{biased} biased accuracy. The robustness of the model is evaluated by the challenging subset $\hat{\mathcal{D}}_{unbiased} \subset \mathcal{D}_{unbiased}$ where every example contains at least one of the spurious tokens. The accuracy on this subset is called robust accuracy. The gap between biased accuracy and robust accuracy tells us how much degradation the model is suffering. Table 4.2 show that NFL brings significant improvement in terms of robust accuracy. While standard fine-tuning is suffering a random-guessing accuracy, NFL enjoys a low degradation and high robust accuracy. The best-performing NFL even achieves a robust accuracy that is close to the upper bound.

4.4 Comparison Between Pre-trained Language Models

It is known that RoBERTa is more robust than BERT due to the larger and diversified pretraining data [23]. As NFL is essentially using the off-the-shelf pretrained language model to protect the main model, we test a hypothesis that language models with richer pretraining are more capable of protecting the main model. Our claim is supported by the experiments shown in Figure 4.3. Blue bars represent robust accuracies and red bars

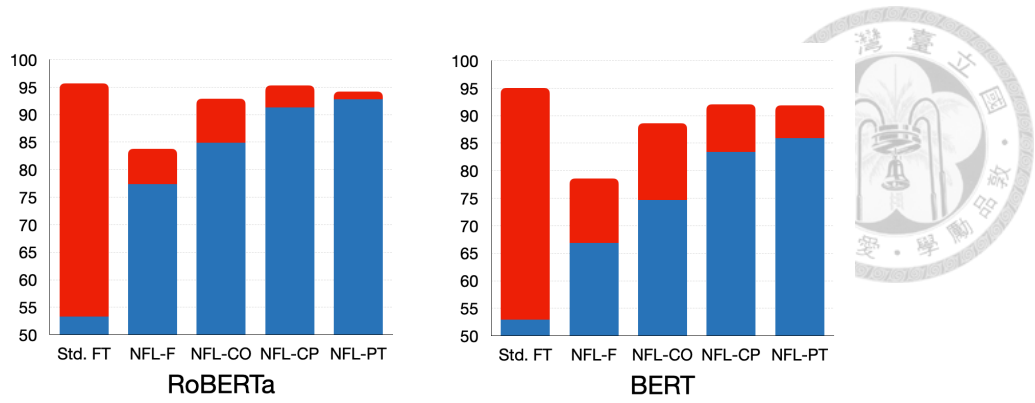


Figure 4.3: Results of Amazon binary with different pretrained language models.

represent robustness gaps. The robustness gaps in RoBERTa is smaller than that of BERT.

While NFL is useful across different choices of pretrained language models, the robustness gap is smaller in RoBERTa than that of BERT when using a regularization term.



Chapter 5 Naturally Occuring Spurious Correlation

We continue to study naturally occurring spurious correlations with our neighborhood analysis. Spurious correlations are naturally present in datasets due to various reasons such as annotation artifacts, flaws in data collection and distribution shifts [9, 11, 33]. Previous work (28; 27) pointed out in SST2, the token *spielberg* has high co-occurrences with positive but the token itself does not cause the label to be positive. Therefore it is likely to be spurious. Borkan et al. (2019) [2] revealed that models tend to capture the spurious correlations in the toxicity detection dataset by relating the names of frequently targeted identity groups such as *gay* and *black* with toxic content.

5.1 Dataset

SST2 This dataset consists of texts from movie reviews [22]. It is also a part of the GLUE [26] benchmark for evaluating NLU systems. This dataset contains 67,300 training examples. We use 10% of the training data for validations and use the original 872 validation data for testing.

Target token	Neighbors before fine-tuning	Neighbors after fine-tuning
spielberg (SST2)	spiel, spiegel, rosenberg, goldberg zimmerman, iceberg, bewild	exquisite, dedicated, freedom important, leadings, remarkable
gay (Jigsaw)	beard, bomb, dog, wood moral, fat, fruit, cam, boy	whites, lesbians, fucked, black foreigner, shoot, arse, upsetting
black (Jigsaw)	white, racist, brown, silver, gray green, blue, south, liberal, generic	ass, demon, fuck, muslim homosexual, fools, obnoxious
Canada (Jigsaw)	Spain, Australia, California, Italy Britain, Germany, France, Brazil	hypocrisy, ridiculous, bullshit, fuck, stupid, damn, morals, idiots, pissed

Table 5.1: Nearest neighbors of the spurious tokens before and after fine-tuning.

5.2 Neighborhood Analysis of Naturally Occuring Spurious Correlations

Table 5.1 shows the nearest neighbors of the naturally occurring spurious tokens before and after fine-tuning. Words in red are associated with negative/toxic labels while words in blue are associated with positive labels according to human annotators. our framework can explain the spurious tokens pointed out by previous work. These naturally occurring spurious tokens demonstrate similar behavior as that of synthetic spurious tokens, *spielberg* is aligned with genuine tokens of positive movie reviews and the names of targeted identity groups (*gay* and *black*) are aligned with vulgar, offensive words as well as other targeted names.

5.3 Detecting Spurious Tokens

There has been a growing interest in detecting spurious correlations automatically to enhance the interpretability of models' prediction. Practitioners may also decide whether they need to collect more data from other sources or simply masking the spurious tokens based on the results of detection. [6, 27, 28]. In this section, we show that our proposed

Top naturally occurring spurious tokens in each dataset	
SST2	allow, void, practice, sleeps, not, problem, taste, bottom
Amazon	liberal, flashy, reck, reverted, passive, average, washed, empty
Jigsaw	Canada , witches, sprites, rites, pitches, monkeys, defeating, animals

Table 5.2: List of top spurious tokens according to their spurious scores verified by human annotators.

spurious score can also be used to detect naturally occurring spurious tokens. As we do not have access to a f^* that is trained on $\mathcal{D}_{\text{unbiased}}$ in this setting, we simply use the model fine-tuned on the potentially biased dataset that we would like to perform detections. We compute the spurious score of every token according to Equation 3.1. The tokens with largest spurious score are listed in Table 5.2, where the genuine tokens are filtered by human annotators. Take the top spurious token **Canada** as an example, our observation of the changes in neighborhood analysis still holds true (Table 5.1). The precision of our detection scheme for top 10/20/30 spurious tokens are evaluated by human annotators and listed in Table 5.3.

Dataset	Precision		
	Top 10	Top 20	Top 30
SST2	0.60	0.50	0.53
Amazon	0.50	0.40	0.40
Jigsaw	0.50	0.45	0.43

Table 5.3: Precision of the detected spurious tokens according to human annotators.

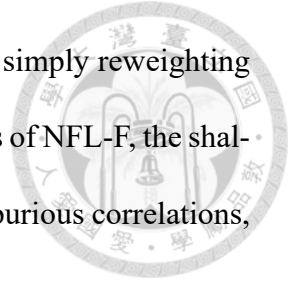


Chapter 6 Related Work

6.1 Mitigating spurious correlations

Existing mitigation approaches can be classified into two categories—data-based and model-based [18]. Data-based approaches modify the datasets to eliminate spurious correlations, often under the assumption that the correlations are known beforehand [19, 20, 29]. Model-based approaches aim to make the models less vulnerable to spurious correlations by model ensembling or regularization [10, 24, 31]. On the contrary, our methods do not assume the knowledge of spurious correlations. With the condition that the spurious correlation is not known beforehand, some model-based approaches make assumptions on the properties of spurious correlations and prevent models from learning the patterns. Clark et al. (2020) [4] propose an ensemble learning framework that relies on the assumption that spurious correlations are patterns that are overly simple. They leverage a shallow model to capture overly simplistic patterns and avoid the main model from learning the same patterns. In the domain of Computer Vision, Kirichenko et al. (2022) [14] show that state-of-the-art performance can be recovered by re-training only the classification layer on a small reweighting data where the spurious correlation does not hold, reducing reliance on the spurious background features. Different from their findings, we discover that spurious correlations in text classification tasks corrupt the feature extractor by align-

ing the representations of spurious tokens and genuine tokens. Thus, simply reweighting the features learned by ERM is undesired. And also in the experiments of NFL-F, the shallow classification layer is capable of showing its robustness against spurious correlations, indicating the importance of learning a robust feature extractor.



6.2 Model-based detection of spurious tokens

In the context of text classification, some of the previous studies are interested in understanding spurious correlations by detecting spurious tokens. They generally work by finding tokens that contribute the most to models' prediction [27, 28], but do not uncover the internal mechanism of how those spurious tokens acquire excessive importance weights and dominate models' predictions. [28] requires human annotated examples of genuine/spurious tokens while [27] requires data from multiple domains for the same task. As such external data might be too expensive to collect, our work is motivated to use the widely available pretrained language models as an anchor point and eventually able to obtain similar performances with a more reasonable assumption.



Chapter 7 Conclusion

In this paper, we present our neighborhood analysis to explain how models interact with spurious correlation. Through the analysis, we learn that the corrupted language models capture spurious correlations in text classification tasks by mis-aligning the representation of spurious tokens and genuine tokens. The analysis not only provides a deeper understanding of the spurious correlation issue but can additionally be used to detect spurious tokens. In addition, our observation from the analysis allows designing an effective family of regularization methods that prevent the models from capturing spurious correlations by preventing mis-alignments and preserving the semantic knowledge with the help of off-the-shelf pretrained language models.



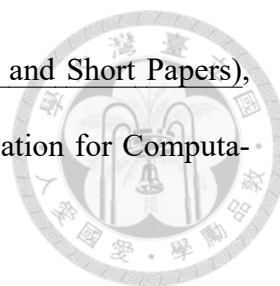
Chapter 8 Limitation

Our proposed NFL family is built on the assumption that off-the-shelf pretrained language models are unlikely to be affected by spurious correlation as the self-supervised learning procedures behind the models do not involve any labels from downstream tasks. Erroneous alignments formed by biases in the pretraining corpora are then beyond the scope of this work. As per our observation in Section 4.4, we echo the importance of pretraining language models with richer contexts and diverse sources to prevent biases in off-the-shelf pretrained language models in future studies.



References

- [1] P. Bansal and A. Sharma. Controlling learned effects to reduce spurious correlations in text classifiers, 2023.
- [2] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification, 2019.
- [3] C. Clark, M. Yatskar, and L. Zettlemoyer. 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, Nov. 2019. Association for Computational Linguistics.
- [4] C. Clark, M. Yatskar, and L. Zettlemoyer. Learning to model and ignore dataset bias with mixed capacity ensembles. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3031–3045, Online, Nov. 2020. Association for Computational Linguistics.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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



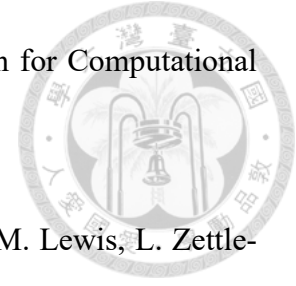
- Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [6] D. Friedman, A. Wettig, and D. Chen. Finding dataset shortcuts with grammar induction. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4345–4363, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.
- [7] M. Gardner, W. Merrill, J. Dodge, M. Peters, A. Ross, S. Singh, and N. A. Smith. 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, Nov. 2021. Association for Computational Linguistics.
- [8] M. Glockner, V. Shwartz, and Y. Goldberg. Breaking NLI systems with sentences that require simple lexical inferences. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 650–655, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [9] S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith. 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, June 2018. Association for Computational Linguistics.
- [10] H. He, S. Zha, and H. Wang. 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, Nov. 2019. Association for Computational Linguistics.

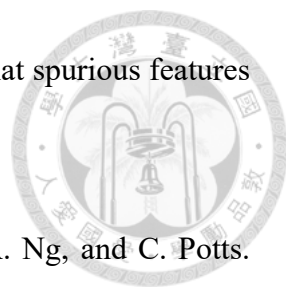


- [11] C. Herlihy and R. Rudinger. MedNLI is not immune: Natural language inference artifacts in the clinical domain. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 1020–1027, Online, Aug. 2021. Association for Computational Linguistics.
- [12] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In H. D. III and A. Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 4411–4421. PMLR, 13–18 Jul 2020.
- [13] N. Joshi, X. Pan, and H. He. Are all spurious features in natural language alike? an analysis through a causal lens. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 9804–9817, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.
- [14] P. Kirichenko, P. Izmailov, and A. G. Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In The Eleventh International Conference on Learning Representations, 2023.
- [15] X. Liu, K. Ji, Y. Fu, W. Tam, Z. Du, Z. Yang, and J. Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short

Papers), pages 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics.



- [16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.
- [17] A. Liusie, V. Raina, V. Raina, and M. Gales. Analyzing biases to spurious correlations in text classification tasks. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 78–84, Online only, Nov. 2022. Association for Computational Linguistics.
- [18] J. M. Ludan, Y. Meng, T. Nguyen, S. Shah, Q. Lyu, M. Apidianaki, and C. Callison-Burch. Explanation-based finetuning makes models more robust to spurious cues, 2023.
- [19] T. McCoy, E. Pavlick, and T. Linzen. 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, July 2019. Association for Computational Linguistics.
- [20] R. Sharma, J. Allen, O. Bakhshandeh, and N. Mostafazadeh. Tackling the story ending biases in the story cloze test. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 752–757, Melbourne, Australia, July 2018. Association for Computational Linguistics.

- 
- [21] C. Si, D. Friedman, N. Joshi, S. Feng, D. Chen, and H. He. What spurious features can pretrained language models combat?, 2023.
- [22] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics.
- [23] L. Tu, G. Lalwani, S. Gella, and H. He. An empirical study on robustness to spurious correlations using pre-trained language models. Transactions of the Association for Computational Linguistics, 8:621–633, 2020.
- [24] P. A. Utama, N. S. Moosavi, and I. Gurevych. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8717–8729, Online, July 2020. Association for Computational Linguistics.
- [25] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.
- [26] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. 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, Nov. 2018. Association for Computational Linguistics.



- [27] T. Wang, R. Sridhar, D. Yang, and X. Wang. Identifying and mitigating spurious correlations for improving robustness in NLP models. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1719–1729, Seattle, United States, July 2022. Association for Computational Linguistics.
- [28] Z. Wang and A. Culotta. Identifying spurious correlations for robust text classification. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3431–3440, Online, Nov. 2020. Association for Computational Linguistics.
- [29] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. HellaSwag: Can a machine really finish your sentence? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics.
- [30] X. Zhang and Y. LeCun. Which encoding is the best for text classification in chinese, english, japanese and korean? CoRR, abs/1708.02657, 2017.
- [31] J. Zhao, X. Wang, Y. Qin, J. Chen, and K.-W. Chang. Investigating ensemble methods for model robustness improvement of text classifiers. In Findings of the Association for Computational Linguistics: EMNLP 2022, pages 1634–1640, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.
- [32] Y. Zheng, J. Zhou, Y. Qian, M. Ding, C. Liao, L. Jian, R. Salakhutdinov, J. Tang, S. Ruder, and Z. Yang. FewNLU: Benchmarking state-of-the-art methods for few-shot natural language understanding. In Proceedings of the 60th Annual Meeting

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 501–516, Dublin, Ireland, May 2022. Association for Computational Linguistics.

- [33] C. Zhou, X. Ma, P. Michel, and G. Neubig. Examining and combating spurious features under distribution shift. In M. Meila and T. Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 12857–12867. PMLR, 18–24 Jul 2021.



Appendix A — Training details

We use pretrained BERT, RoBERTa and the default hyperparameters in Trainer, offered by Huggingface in all of our experiments. We also use the implementation from Liu et al. (2022) [15] for NFL-PT. The models are trained for 6 epochs except for NFL-PT which takes 100 epochs. The sequence length of continuous prompts in NFL-PT is set to 40.



Appendix B — Weights of regularization terms

In the experiment of Amazon binary, we search the hyperparameter of the weights of NFL-CO and NFL-CP regularization terms over $\{1, 10, 100, 1000, 10000, 15000, 20000\}$. Generally there is a trade-off between in-distribution (biased) accuracy and out-of-distribution (robust) accuracy. Nonetheless, we can observe from the graph above that as we increase the weights of the regularization term, the drop in-distribution accuracy is insignificant while the improvement in robustness is tremendous. In all of the experiments, we set the weights to be 15000.

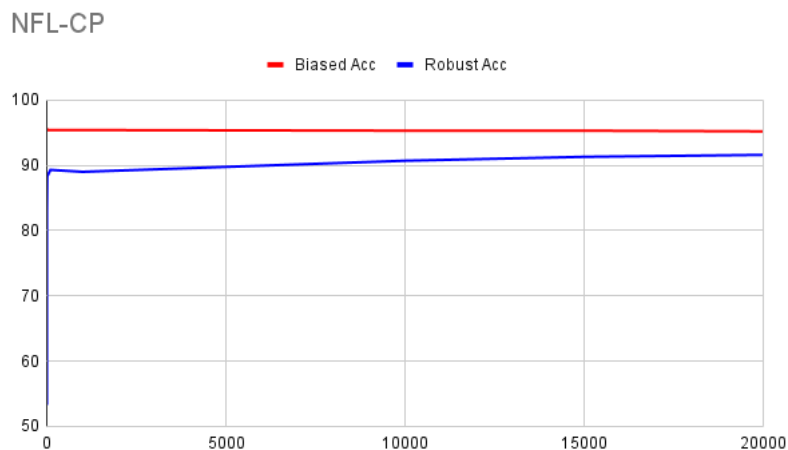


Figure B.1: Performance of NFL-CP under different choices of λ .

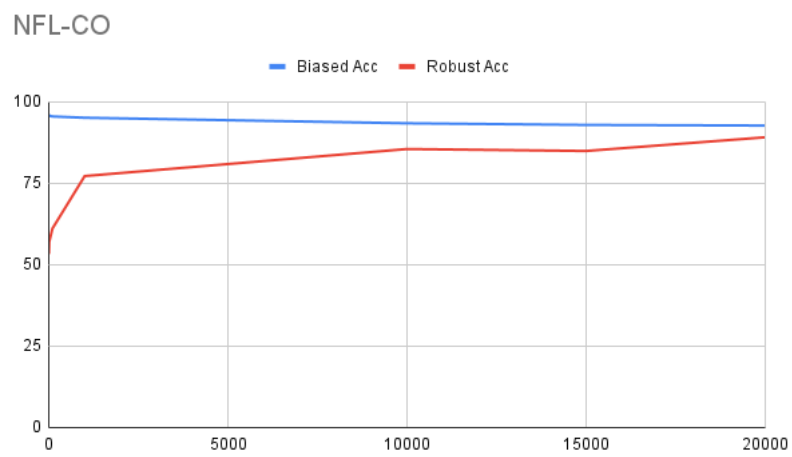


Figure B.2: Performance of NFL-CO under different choices of λ .