Explore Spurious Correlations at the Concept Level in Language Models for Text Classification

Yuhang Zhou [1], **Paiheng Xu** [2], **Xiaoyu Liu** [2], **Bang An** [2], **Wei Ai** [1], **Furong Huang** [2]
[1] College of Information Studies, University of Maryland, College Park
[2] Department of Computer Science, University of Maryland, College Park
{tonyzhou, paiheng, xliu1231, bangan, aiwei, furongh}@umd.edu

Abstract

Language models (LMs) have achieved notable success in numerous NLP tasks, employing both fine-tuning and in-context learning (ICL) methods. While language models demonstrate exceptional performance, they face robustness challenges due to spurious correlations arising from imbalanced label distributions in training data or ICL exemplars. Previous research has primarily concentrated on word, phrase, and syntax features, neglecting the concept level, often due to the absence of concept labels and difficulty in identifying conceptual content in input texts. This paper introduces two main contributions. First, we employ ChatGPT to assign concept labels to texts, assessing concept bias in models during fine-tuning or ICL on test data. We find that LMs, when encountering spurious correlations between a concept and a label in training or prompts, resort to shortcuts for predictions. Second, we introduce a data rebalancing technique that incorporates ChatGPT-generated counterfactual data, thereby balancing label distribution and mitigating spurious correlations. Our method's efficacy, surpassing traditional token removal approaches, is validated through extensive testing. [1]

1 Introduction

Pre-trained language models (LMs), leveraging extensive text corpora in their pre-training phase, have demonstrated remarkable effectiveness in a variety of natural language understanding tasks (Wei et al., 2022; Devlin et al., 2018). Nevertheless, LMs encounter issues with spurious correlations during fine-tuning or instruction-following stages (Zhang et al., 2022; Wang et al., 2022; Tang et al., 2023). These correlations involve specific associations between features and labels that, while prevalent in training data, are erroneously generalized as rules, leading to reduced performance.

Fine-tuning / In-context Learning

Training / Demonstrations	Concept	Label
It is one of my favorite fast food burgers a must try for burger fans.	food	1
Love their happy hour in the bar! They have three different kinds of salsa, the green one is the best.	food	1
This is just an okay airport - the layout is not great, although I do have to give props to them for easy car rental terminals.	none	0
Been closed for months, was open less than 6 month enough said! Wasn't very good at all....	none	0

Test Sentence	Prediction	Label
The Thai steak is overdressed and underwhelming. The french dip looked delicious. I know better for next time.	1	0

Figure 1: **Example of concept-level spurious correlations.** In the training data or demonstrations, texts containing the concept "food" are mostly with label 1 (positive sentiment). During test, when encountering a sentence with the tokens "Thai steak," not appearing in the training/prompts but indicating the concept "food", the models rely on the shortcut between the concept "food" and label 1 to give the wrong prediction.

Current research on spurious correlations in LMs spans various dimensions, such as token-level shortcuts in text classification (Wang et al., 2022; Tang et al., 2023; Chew et al., 2023), syntactic heuristics in natural language inference (McCoy et al., 2019), sentence triggers in text classification (Tang et al., 2023; Jia and Liang, 2017), and topic shortcuts in machine translation (Borah et al., 2023). Moreover, spurious correlations with demographic concepts like race or sex, raise fairness concerns (Kleinberg et al., 2018). Yet, studies seldom address semantic spurious correlations at a broader concept level.

We define spurious correlations at the concept level as: Most texts featuring a certain concept in training data (or prompts) are linked with a specific label, leading LMs to inappropriately rely on this association for predictions. For instance, in Figure 1, terms like "salsa," "fast food burgers," or "Thai steak" denote the concept "food." A prevalent association between "food" and label 1 in training data

[1] Our code and data are available at `https://github.com/Tonyzhou98/concept-spurious-correlation`

or prompts results in LMs forming a concept-level spurious correlation, mistakenly assigning some "food"-related texts to label 1.

The tendency of LMs to learn concept-level shortcuts might stem from the formation of similar embeddings for expressions related to the same concept during fine-tuning or pre-training, driven by their semantic similarities. As Figure 2 suggests, various expressions of a concept cluster closely in the embedding space of fine-tuned or pre-trained LMs. When similar embeddings frequently coincide with a label in training or demonstrations, LMs tend to adopt the shortcut. We offer an in-depth analysis using a specific dataset in Section 3.2.

In the first part of our study, we assess and quantify concept-level spurious correlations in LMs across both fine-tuning and ICL scenarios within text classification tasks. Initially, we employ the advanced large language model (LLM), ChatGPT, to identify relevant concepts in each dataset (Ouyang et al., 2022) and to predict the presence of these concept labels. In the fine-tuning setting, we train LMs on both the original dataset and a concept-biased counterpart. Our findings indicate that LMs exhibit concept-level spurious correlations in standard benchmarks, with more pronounced prediction biases emerging from increasingly imbalanced data. In the ICL setting, we compare the performance of LMs on concept-balanced and concept-biased prompts, demonstrating that biased prompts lead to more skewed inferences.

The second part of the paper explores the use of data rebalancing techniques to counteract these spurious correlations in a fine-tuning framework. We introduce an upsampling strategy that incorporates counterfactual texts generated by ChatGPT, which effectively reduces bias while maintaining the utility (i.e., accuracy) of the LMs. In summary, our research makes three significant contributions:

- We are the first to investigate spurious correlations at a general concept level and introduce a metric to quantify these correlations.
- Through experiments on various benchmark data for text classification, we demonstrate that LMs are prone to adopting learned concept-level shortcuts in both fine-tuning and ICL settings.
- We introduce an effective upsampling approach, incorporating counterfactuals generated by LLMs, to mitigate concept-level bias.

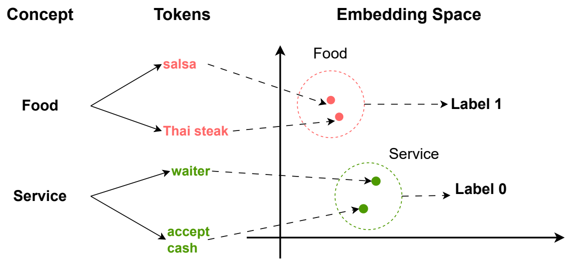

Figure 2: A concept can be expressed in multiple expressions, and in the embedding space of LMs, these expressions of one concept can be mapped into similar positions. LMs will form a shortcut between a specific concept and a label and utilize in the future prediction.

2 Exploring Concept-level Spurious Correlations

2.1 Obtaining the Concept Labels

Due to the lack of human-annotated metadata indicating concepts in most text classification datasets, and considering the superior capabilities of LLMs in text annotation tasks over human annotators (Gilardi et al., 2023), we utilize ChatGPT (GPT-3.5) to annotate concept labels for sentences in text classification datasets (Ouyang et al., 2022). Our annotation process involves an annotation prompt P_a that contains the annotation instruction and five demonstrations, a text input x, LLM M_a, and a candidate concept set $C = \{C_1, C_2, \cdots, C_k\}$ (we describe how we curate the candidate set in Section 3).

The annotation process is formalized as: $a(x) = M_a(P_a \| C \| x)$, where $a(x)$, the set of concept labels for text x, may contain zero or several concepts selected from the pre-defined concept set C ($a(x) \subset C$), and $\|$ denotes the concatenation operation. To ensure reliability, we repeat the annotation process twice with a temperature setting of 0.7 and retain only those examples and labels that are consistently identified by both LLM annotators.

2.2 Measuring Concept Spurious Correlations

For the text classification task, we consider an input $x \in \mathcal{X}$ accompanied by concept labels $a(x) \subset C$. Each input is associated with a ground truth classification label $y = l$ from the output label space $\mathcal{Y}$, $l \in \{0, 1, \cdots, n\}$. Given a LM classifier $M : \mathcal{X} \to \mathcal{Y}$, if the model avoids utilizing potential concept-level shortcuts from $c \to y, c \in C$, the following condition is satisfied:

$$\begin{aligned} &\mathbb{E}_x[p_M(\hat{y} = l | x, c \in a(x), y = l)] \qquad (1)\\ =&\mathbb{E}_{x'}[p_M(\hat{y} = l | x', c \notin a(x'), y = l)] \quad \forall l \in \mathcal{Y}. \end{aligned}$$

Here, $\hat{y}$ denotes the predicted label, while p_M represents the probability predicted by model M. The inputs x and x', belonging to the space $\mathcal{X}$, contain the concept c or do not contain it, respectively.

Equation 1 implies a critical condition: regardless of the presence of concept c in the input, the models should maintain an unbiased estimate of the predicted probability on average. The expression $\mathbb{E}_x[p_M(\hat{y} = l|x, c \in a(x), y = l)]$ can be interpreted as the model's accuracy on texts that are labeled l and incorporate the concept c.

Denote Δ_{c_i} as the difference in model accuracy between texts with or without concept c that have label $i \in \mathcal{Y}$. We further infer from Equation 1 that:

$$\begin{aligned}\Delta_{c_i} =&\mathbb{E}_x[p_M(\hat{y} = i|x, c, y = i)] \\ &-\mathbb{E}_{x'}[p_M(\hat{y} = i|x', \neg c, y = i)] = 0,\end{aligned} \tag{2}$$

where $\neg c$ denotes concept c is not in input x. We hypothesize, if there exists a spurious correlation in models between concept c and label i, the following conditions would hold:

$$\mathbb{E}_x[p_M(\hat{y} = i|x, c, y = i)] > \mathbb{E}_{x'}[p_M(\hat{y} = i|x', \neg c, y = i)]$$
$$\mathbb{E}_x[p_M(\hat{y} = j|x, c, y = j)] < \mathbb{E}_{x'}[p_M(\hat{y} = j|x', \neg c, y = j)]$$

Then we have $\Delta_{c_i} > 0 > \Delta_{c_j}$. Otherwise, if the spurious correlation is between c and j, then $\Delta_{c_j} > 0 > \Delta_{c_i}$. We propose to measure the discrepancy between Δ_{c_i} and Δ_{c_j} to quantify the spurious correlation. Hence, considering the output space $\mathcal{Y}$, we quantify the model's reliance on shortcut mapping as the average discrepancy in the accuracy difference $\Delta_{c_i} - \Delta_{c_j}$ across all label combinations.

$$\text{Bias@C} = \frac{1}{\binom{n}{2}} \sum_{i,j \in \mathcal{Y}} (\Delta_{c_i} - \Delta_{c_j}), i > j$$

For the binary classification task, the bias measurement is simplified to Bias@C $= \Delta_{c_1} - \Delta_{c_0}$

A Bias@C approaching 0 indicates minimal reliance on concept shortcuts. Conversely, a positive Bias@C value suggests that model is more likely to predict larger labels when the input includes concept c, and the opposite for a negative value.

2.3 Evaluation of Model Robustness to Concept Shortcut in Fine-tuning

To assess LMs' robustness against spurious correlations for concept c across varying scales of concept bias during fine-tuning, we fine-tune models on the original dataset $\mathcal{D}_{ori}$ and a concept-biased dataset $\mathcal{D}^c_{biased}$ separately. To further demonstrate the impact of concept-level spurious correlation, we construct $\mathcal{D}^c_{biased}$ of concept c by filtering $\mathcal{D}_{ori}$, where, for each data point, we only keep those with the majority labels under concept c. After fine-tuning on $\mathcal{D}_{ori}$ or $\mathcal{D}^c_{biased}$, we evaluate models on test data using Bias@C to quantify spurious correlations.

We report accuracy on the test data for utility performance. However, label distributions with or without the concept c may be imbalanced. Following previous work (Chew et al., 2023), we rebalance the test set by downsampling and report the inference accuracy (**robust accuracy**) on the balanced subset for examples with concept c (**Acc@C**) and without concept c (**Acc@NoC**), respectively.

2.4 Evaluation of Model Robustness to Concept Shortcut in ICL

As LLMs have shown outstanding performances with the ICL setting, we are interested in investigating the concept shortcut in the demonstrations. The prompt P for ICL contains three parts: 1) the instruction s, 2) the demonstrations with h exemplars (text + labels), and 3) the test input x_{test}.

We consider the sentiment classification task and concatenate the h exemplars together with the form "Input: x. The sentiment label is $v(y)$". The label verbalizer $v(y)$ will transfer 0 to "negative" and 1 to "positive" when the label is binary and will maintain the original numerical rating scales when multiple classes ($n \geq 3$). The ICL process is formulated as $f(x_{test}) = M(P \| x_{test})$, where $f(x_{test})$ is a categorical variable belonging to $\mathcal{Y}$.

We create two types of prompts: the biased prompt P_{biased} and the balanced prompt $P_{balanced}$ by changing the label distributions in the demonstrations. For P_{biased}, we insert $\frac{h}{2}$ numbers of exemplars containing the concept c with the label $l \in \{l_1, l_2, \cdots, l_k\}$ (the majority ground truth labels) and $\frac{h}{2}$ numbers of exemplars without c with the other labels. For $P_{balanced}$, we split the exemplars with concept c or not half by half, but ensure balanced labels. We compare the results of Bias@C and robust accuracy with two types of prompts. Since the ICL are sensitive to the exemplars, we repeat the experiments three times with differently selected exemplars and report the average values.

3 Dataset Construction and Analysis

Models We assess and mitigate concept-level bias in DistilBERT and LLAMA2 7B in fine-tuning

Dataset	# Training	# Test	# Labels	Concept
AS	70,117	8,000	5	size, color, style
IMDB	14,956	4,000	2	acting, comedy, music
Yelp	34,184	4,000	2	food, price, service
CeBaB	7,350	2,000	5	service, food, ambiance
BoolQ	2393	2,000	2	country, history

Table 1: Dataset statistics and the labeled concept for each dataset. AS represents the Amazon Shoe dataset.

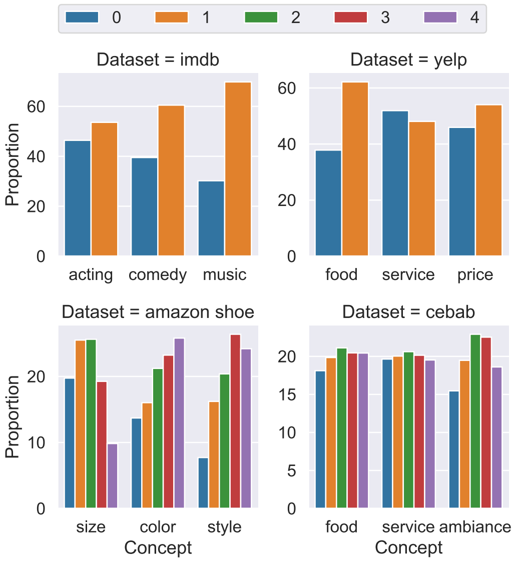

Figure 3: **Label distribution of the texts with a specific concept for each dataset.** We can observe the label distribution in multiple concepts, such as "music" in IMDB, "food" in Yelp datasets are highly imbalanced.

setting (Sanh et al., 2019; Touvron et al., 2023) and GPT3.5 in the ICL setting (Ouyang et al., 2022). We fully fine-tune the DistillBert. For LLAMA2, we apply the Lora method for efficient fine-tuning (Hu et al., 2021). Details of the model implementations are included in Appendix A.

Dataset We select four sentiment classification tasks to evaluate the model robustness: Yelp (Zhang et al., 2015), IMDB (Maas et al., 2011), Amazon Shoe (He and McAuley, 2016), and CeBaB (Abraham et al., 2022). Amazon shoe and CeBaB datasets with 5 classes, 0 indicating the most negative and 4 indicating the most positive, are reviews of shoes in Amazon and OpenTable. IMDB and Yelp are binary classification datasets (0 indicating negative and 1 indicating positive), with reviews from the IMDB and Yelp platforms.

Additionally, we include a binary question answering (QA) dataset BoolQ (Clark et al., 2019), which asks Yes/No questions. It takes a paired question and passage as the input to LMs and outputs 1 (Yes) or 0 (No). In the following part, we define the **positive** class as datapoints with Label 3 and 4 for the 5-way classification tasks and those with Label 1 for the binary classification task. We define the remaining datapoints as the **negative** class.

Concept For CeBaB, we adopt human-annotated concept labels. For Amazon Shoe, IMDB, Yelp, and BoolQ where there are no concept annotations, we first use ChatGPT to query the concepts embedded in each sentence and count the number of occurrences for each concept following (Fang and Zhang, 2022) to generate concept-level explanations. We then extract the most frequent concepts and identify the concepts whose existence should not influence the sentiments of the text or the Yes/No answer to the question (2 concepts for BoolQ due to more diverse topics and 3 concepts for other datasets). Finally, we use ChatGPT to annotate whether each text input contains the selected concept.

To examine the quality of the annotation, we experiment on the human-annotated "service" concept from the CeBaB dataset and ask the ChatGPT to label the concept again. We find that ChatGPT can achieve an accuracy of 90.4% to the gold standard concept labels, comparable to an average agreement of 92.9% for five human annotators given by CeBaB, indicating the reliability of LLM annotations. Table 1, presents dataset statistics and the labeled concept lists for the five datasets.

3.1 Biased Dataset Construction

We first visualize the label distribution for the input texts with the concept c for each sentiment classification in Figure 3. We observe that for the original datasets, the concept-label distributions are balanced for most concepts, but not as balanced for concepts such as "food" in Yelp dataset, "music" in IMDB, and "style" in Amazon Shoe. In 10/12 cases, positive labels comprise large proportions of the corpus with certain concepts. To further demonstrate the impact of concept-level spurious correlation caused by imbalanced concept-label distribution, we construct a biased dataset $\mathcal{D}^{c}_{biased}$ which, for each concept c, only includes the majority class (positive or negative). Specifically, we keep negative class for "size" in Amazon Shoe and "service" in Yelp. For other concepts in the sentiment datasets, we keep positive class. For BoolQ, we keep negative class with "country" and positive class with "history".

Concept	Top associated tokens extracted from each concept
Size	9m, small, c/d, sizing, 105, us, 95, 8w, 81/2, 7-75
Color	royal, camel, muted, champagne, color, taupe, maroon, teal, greenish, white
Style	stylish, vibe, comfort, swedish, look, all-time, (55), styling, yearround, frumpy

Table 2: Tokens with high associations (top 10 PMI values) to each concept in Amazon Shoe dataset.

3.2 Embedding Analysis of Associated Tokens

As shown in Figure 2, we hypothesize that expressions of a concept have similar semantic embeddings, leading to shortcut learning. To verify the hypothesis and further motivate the measurement of spurious correlations, we extract the embeddings of the associated tokens with each concept in the Amazon shoe dataset. We observe whether the embeddings of tokens associated with the same concept are similar using clustering.

We apply the point-wise mutual information (PMI) between the token and the concept to measure the association. For a dataset with a concept c, we calculate the PMI of each token t to concept c, which is $\text{PMI}(t, c) = \log \frac{p(t,c)}{p(t)p(c)}$, where $p(t)$, $p(c)$ and $p(t, c)$ refer to the probability of the text containing t, c and both together. The higher value of PMI suggests a stronger association between t and c. We present tokens with the top 10 PMI values for each concept in Table 2.

From the associated tokens in Table 2, we observe tokens with various semantics associated with one concept, such as "small," "sizing" and "9m" to express the "size" concept. We use the tokens in Table 2 to perform the clustering. We exclude tokens with special character, such as "c/d," "81/2" and "(55)" that affect the interpretation of the results. We feed the tokens into the DistilBERT fine-tuned on the Amazon Shoe and retrieve the corresponding embedding from the model last layer. If the token is tokenized into multiple sub-words, we follow the previous work and calculate the average as the token embedding (Wolfe and Caliskan, 2021). We calculate the cosine distance between their token embeddings and apply hierarchical clustering (Bar-Joseph et al., 2001).

From Figure 4, we can identify four small clusters, each representing a concept. We observe that the LMs will produce similar internal representations for tokens associated with the same concept label. If the label under a concept is imbalanced, the models may learn the undesired shortcut between similar embeddings and a label. This observation motivates the measurement of spurious correlation at the concept level.

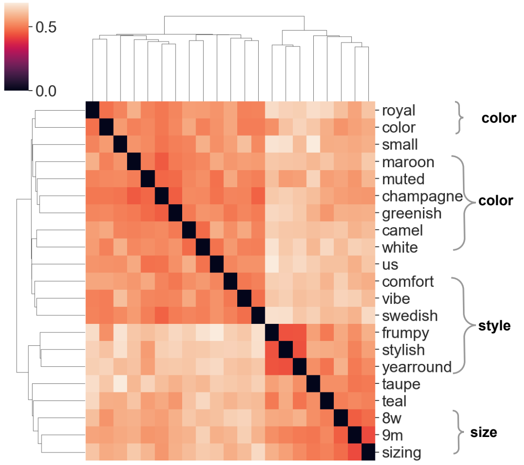

Figure 4: **Clusters of word embeddings of top associated tokens for each concept from Amazon shoe dataset.** The dendrogram on the side indicates the hierarchical clustering structure among the tokens.

4 Results of Spurious Correlation Measurement

4.1 Spurious Correlations in Fine-tuning

To evaluate the robustness to the concept shortcut in the fine-tuning setting, we fine-tune the models on the original dataset $\mathcal{D}_{ori}$ and the biased dataset $\mathcal{D}^{c}_{biased}$, respectively, and measure the concept bias. For each concept, we report the metric Bias@C to quantify the strength of spurious correlations and the robust accuracy for texts with and without concept, i.e., Acc@C and Acc@NoC, as the utility performance. For Bias@C, closer to 0 indicates weaker spurious correlations for concept C, and for robust accuracy, a higher value suggests better performance. We present the results for DistilBERT on sentiment classification in Table 3 and BoolQ dataset in Table 7 Appendix B.

Fine-tuned LMs present a clear concept bias when trained on both original and biased data. Table 3 shows that when the models are trained on $\mathcal{D}_{ori}$, they utilize spurious correlations in the training data to make inferences. For example, for "style" in the Amazon Shoe and "music" in IMDB, the Bias@C values in $\mathcal{D}_{ori}$ are large due to highly imbalanced label distribution. Since these datasets are well curated and widely adopted, the fact that we are able to identify several highly biased concepts by only investigating the top 3 fre-

Data: Amazon Shoe	Size (pos < neg)			Color (pos > neg)			Style (pos > neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**2.11**	**57.94**	**55.94**	**1.38**	57.19	**55.50**	**11.56**	**57.18**	**56.12**
Trained on $\mathcal{D}^{c}_{biased}$	-3.77	56.75	47.76	14.99	**57.56**	48.19	13.74	56.39	54.92
Data: IMDB	Acting (pos > neg)			Comedy (pos > neg)			Music (pos > neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**3.70**	88.49	**91.24**	**2.51**	**91.62**	**91.85**	12.07	**91.55**	88.93
Trained on $\mathcal{D}^{c}_{biased}$	8.69	**88.87**	88.67	5.14	90.55	90.50	**8.25**	90.55	**89.30**
Data: Yelp	Food (pos > neg)			Service (pos < neg)			Price (pos > neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**3.42**	93.86	97.09	**2.44**	**93.33**	**97.23**	**-0.32**	94.04	**94.44**
Trained on $\mathcal{D}^{c}_{biased}$	3.93	**93.98**	**97.58**	-3.85	90.23	95.14	5.62	**96.03**	94.00
Data: CeBaB	Food (pos > neg)			Service (pos > neg)			Ambiance (pos > neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**-0.71**	**69.48**	**74.12**	**-0.34**	**69.70**	**74.45**	**-0.90**	**72.62**	**75.68**
Trained on $\mathcal{D}^{c}_{biased}$	14.94	61.17	58.14	13.06	67.08	61.58	7.02	69.60	65.96

Table 3: **Model fine-tuning performance with training on original dataset and concept biased dataset for four datasets.** Models trained on the original dataset $\mathcal{D}_{ori}$ tend to behave a bias in some concepts, where the label distribution under concepts is pretty uneven. When fine-tuned on the concept-biased dataset $\mathcal{D}^{c}_{biased}$, both bias (Bias@C) and accuracy results (Acc@C and Acc@NoC) suffer from performance drop. pos > neg: for this concept, more positive texts are in $\mathcal{D}_{ori}$ and in $\mathcal{D}_{biased}$, all texts containing this concept are positive, and vice versa for "pos < neg". The lower absolute values of Bias@C (smaller bias) and the higher accuracy values are in bold.

quent concepts demonstrates the significance of spurious correlation.

Comparing the results between $\mathcal{D}_{ori}$ and $\mathcal{D}^{c}_{biased}$, we find that the absolute values of Bias@C are significantly higher when trained on $\mathcal{D}^{c}_{biased}$ in almost every concept, and the change direction of Bias@C is the same as the trend in the label distribution. For example, the value of Bias@C becomes negative for "service" in Yelp dataset, since we only keep negative reviews with the "service" in $\mathcal{D}^{c}_{biased}$. These observations indicate that a greater bias in the fine-tuning dataset causes the model to rely more on spurious correlations to make predictions.

Regarding utility performance (Acc@C and Acc@NoC), we observe that the models trained on $\mathcal{D}^{c}_{biased}$ have a dramatic performance drop on the texts with the concept in most cases, and the average Acc@C decreases from 79.38% to 74.31%. This pattern suggests that larger spurious correlations affect the utility performance of fine-tuned models. Meanwhile, the average Acc@NoC drops from 78.08% to 76.56%. Its performance drop is not as large as the one of Acc@C, indicating that texts without the concept suffer less from the concept bias in the datasets.

Moreover, we find that for some concepts, the fine-tuned LMs suffer from severe spurious correlation, but the effect of the bias is not fully reflected in the difference between Acc@C and Acc@NoC. For example, the "music" concept in the IMDB dataset has Bias@C = 12.07%, but the difference between Acc@C and Acc@NoC is less than 3%. This is because if the model is biased towards one label due to the spurious correlation, the accuracy improvement towards the biased label can often offset the performance drop of the other side.

We also verify that the concept bias is not simply due to the shortcut on a few words by masking out the associated tokens, and details are shown in Section 5.2. We show fine-tuning results of LLAMA2 models on $\mathcal{D}_{ori}$ and $\mathcal{D}^{c}_{biased}$ in Table 7 and 8 (Appendix B). Similar patterns suggest the generalizability of our findings on models of different sizes.

4.2 Spurious Correlations in ICL

As LMs exhibit clear evidence of utilizing the concept shortcuts in the fine-tuning data, we also want to ask whether LMs use the shortcuts in the exemplars of the prompts when performing ICL. As discussed in Section 2.4, for each concept c in five datasets, we construct a prompt with eight exemplars. Following a similar setup for fine-tuning, we only include the majority class (positive or negative) for exemplars with concept c. Specifically, for "size" in Amazon Shoe and "service" in Yelp, four exemplars with concept c have negative labels and the other four without concept c have positive labels. The labels are flipped for the rest of the concepts. For the balanced prompt $P_{balanced}$, the label is evenly distributed for the exemplars with or without the concept. With the bias in label distribution for both texts with and without the concept,

Data: Amazon Shoe	Size (pos <neg)			Color (pos >neg)			Style (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
ICL with $P_{balanced}$	**0.95**	**50.37**	45.63	**9.46**	**49.63**	**49.54**	**11.58**	**52.98**	53.91
ICL with P_{biased}	-2.64	50.18	**47.20**	10.99	49.58	46.59	12.56	50.21	**54.35**
Data: IMDB	Acting (pos >neg)			Comedy (pos >neg)			Music (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
ICL with $P_{balanced}$	**3.58**	94.94	**96.40**	**3.70**	**96.18**	**95.82**	6.40	**95.43**	**94.33**
ICL with P_{biased}	3.99	**95.05**	96.17	5.64	95.68	94.71	**5.45**	95.33	94.26
Data: Yelp	Food (pos >neg)			Service (pos >neg)			Price (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
ICL with $P_{balanced}$	2.97	**97.66**	98.21	**0.39**	**97.78**	**98.60**	**-0.87**	97.74	97.79
ICL with P_{biased}	**1.70**	97.54	**98.84**	0.92	97.50	98.58	1.17	**97.98**	**98.46**
Data: CeBaB	Food (pos >neg)			Service (pos <neg)			Ambiance (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
ICL with $P_{balanced}$	**0.66**	**61.06**	**64.35**	**0.65**	58.48	62.08	2.58	**64.13**	**64.96**
ICL with P_{biased}	3.24	54.55	59.90	-2.77	**61.91**	**65.53**	**2.21**	61.39	62.45

Table 4: **Model ICL performance with prompting on balanced prompts $P_{balanced}$ and biased prompts P_{biased}.** Larger absolute values for Bias@C indicate that the concept-biased prompts enlarge the extend of models to rely on the shortcut in the demonstrations. The meaning of "pos > neg" and the values in bold are the same as in Table 3.

we expect the LM to use two types of shortcuts: a) from texts with the concept c to one sentiment and b) from texts without the concept to the other sentiment. We present the utility and bias results of ICL for sentiment classification dataset in Table 4 and BoolQ in Table 7 (Appendix B).

Biased prompts enlarge the concept bias in ICL inference From Table 4, we observe a similar pattern for Bias@C as in the fine-tuning part. When the prompt changes from $P_{balanced}$ to P_{biased}, for "service" in Yelp and "size" in Amazon Shoe, where the exemplars with concept c are negative, the values of Bias@C flip from positive to negative, and for most other concepts, where conceptual exemplars are all positive, the value of Bias@C increases. Furthermore, in most cases, the absolute values of Bias@C when using P_{biased} are higher. These observations indicate that the LMs are affected by concept shortcuts within the prompt of ICL.

For utility performance, when changing from $P_{balanced}$ to P_{biased}, the average Acc@C and Acc@NoC decrease from 76.80% to 76.42% and from 76.37% to 75.58%, respectively, which means that spurious correlations harm utility performance regardless of the presence of concepts. For both Bias@C and accuracy, the relative change in the ICL scenario is less than the fine-tuning setting. We conjecture that a few exemplars in prompts make it hard to form a strong shortcut inside the LMs between conceptual contents and a specific label.

5 Mitigate Spurious Correlations

5.1 Mitigation via Rebalancing

We consider two lines of existing data-centric work to mitigate spurious correlations: remove spurious components and rebalance the training dataset (McCoy et al., 2019; Wang et al., 2022). Since it is challenging to identify the conceptual contents in each sentence, we apply dataset rebalancing methods to mitigate the bias at the concept level.

We first downsample the dataset to achieve a balanced label distribution with respect to concept c, denoted as $\mathcal{D}^c_{down-bal}$. The shortcoming of the method is that, for a highly biased dataset, it filters out a large proportion of examples with the majority labels, leading to a sacrifice of the utility performance. To address this, we propose an upsampling method using ChatGPT to generate counterfactual examples with minority labels. Some concurrent work also demonstrates the effectiveness of synthetic data in mitigating bias (Evuru et al., 2024). The resulting dataset is denoted as $\mathcal{D}^c_{up-bal}$.

Suppose that we need $\{a_0, \cdots, a_n\}$ number of examples for labels $\{0, \cdots, n\}$ to make a balanced subset for texts with concept c. We first sample a_0 to a_n numbers of examples from texts with labels 0 to n but without concept c. Then we ask ChatGPT, M_a, to inject the concept c into the selected texts while maintaining the sentiment or the answer to questions. Given the input text x' without concept c, the injection prompt P_i with the instruction, and the exemplars h_c with concept c, the concept injection process is $x_c = M_a(P_i \| h_c \| x')$, where x_c is the generated counterfactual for concept c and

Original	I was fairly disappointed with this zoo. Signage was unclear. Many of the exhibits were on loan
Counterfactual	I was fairly disappointed with this zoo. Signage was unclear. Many of the exhibits were on loan. **The food options consisted of a small cafe with limited choices.**

Table 5: An example of the generated counterfactual data for concept "food" in the Yelp dataset. Text in bold is the generated input with the injected "food" concept.

input x'. We iteratively generate the counterfactual input x_c and add it into the dataset $\mathcal{D}_{ori}$ to form a balanced dataset $\mathcal{D}^c_{up-bal}$ with upsampling. To demonstrate the effectiveness of concept injection, we conduct a case study on a review in the Yelp dataset. As suggested in Table 5, we inject the "food" concept into a review without this concept and observe that ChatGPT effectively injects the food concept, keeps other content unchanged, and maintains the sentiment of the review.

We also propose a baseline method that masks out words highly associated with the concept. This method is used to verify whether balancing distributions of a few tokens removes conceptual shortcuts. We replace words with the top 10 PMI for each concept (word examples are in Table 2) to the **[MASK]** token and name the masked dataset as $\mathcal{D}^c_{mask}$.

5.2 Results of Mitigation Methods

To evaluate the effectiveness of proposed methods, we select concepts with Bias@C greater than 1 in Table 3 and fine-tune on three de-biased datasets $\mathcal{D}^c_{down-bal}$, $\mathcal{D}^c_{up-bal}$, and $\mathcal{D}^c_{mask}$. We report results for DistilBERT in Table 6 and 7 in Appendix B.

Upsampling method reduces the bias and increases utility performance From Table 6, we observe that data rebalancing methods are effective in mitigating spurious correlations. For downsampling ($\mathcal{D}^c_{down-bal}$), it mitigates the mean absolute values of Bias@C from 4.90% to 3.43%, compared to trained on $\mathcal{D}_{ori}$. However, for utility performance, the downsampling obtains less accuracy in 4/8 cases for Acc@C and 5/8 cases for Acc@NoC, indicating that loss of data harms utility. For the upsampling method ($\mathcal{D}^c_{up-bal}$), the mean absolute values of Bias@C are effectively reduced from 4.90% to 2.74%. Furthermore, the average accuracy of Acc@C increases from 79. 24% to 80. 38%, and Acc@NoC is comparable. This observation suggests that adding counterfactual texts to rebalance the data can reduce spurious correlations in the concept level, and more data involved in the fine-tuning can boost the models' utility performance.

Masking out associated tokens ($\mathcal{D}^c_{mask}$) can reduce spurious correlations in most cases, but cannot fully eliminate bias. This observation suggests that due to the various concept expressions, the learned concept shortcut in the model is not equivalent to the shortcut on a few tokens. The utility performance of Acc@C is also lower than that of the proposed upsampling method in 6/8 comparisons.

In summary, among the three mitigation methods, adding the LLM-generated counterfactual inputs achieves the best performance in both the bias mitigation and utility aspects. The same analysis on LLAMA2 models (Table 7 and 9 in Appendix B) reveals similar patterns, which shows the generalizability of our methods.

6 Related Work

Robustness and Bias Current work on studying spurious correlations for LMs can be split into two categories: utilize the shortcuts during training or ICL. For shortcut learning in training, a series of works explores how models take shortcuts in the data for the causal or non-causal perspective (Tu et al., 2020; Sagawa et al., 2020; Geirhos et al., 2020; Ribeiro et al., 2020; Kaushik et al., 2019; Liu et al., 2024; Friedman et al., 2022) and which aspects of shortcuts will be taken for the predictions in different NLP tasks (McCoy et al., 2019; Jia and Liang, 2017; Lai et al., 2021; Zhao et al., 2018; Niu et al., 2020; Poliak et al., 2018), leading to low generalization in the out-of-distribution data or in the designed adversarial data.

Due to the increasing development of LLM on ICL, researchers find that the design of the prompt significantly decides the LLM predictions (Brown et al., 2020; Gao et al., 2020; Liu et al., 2023b; Zhou et al., 2023; Schick and Schütze, 2020). Another line of work finds that LLMs are sensitive to a certain aspect of prompts and not robust when injecting adversarial triggers into prompt (Lu et al., 2021; Zhao et al., 2021; Tang et al., 2023; Si et al., 2023; Zheng et al., 2023). Tang et al. (2023) shows that LLMs use multiple types of shortcuts in the prompts, from letters to words to text style, and Si et al. (2023) find that LLMs exhibit clear feature biases under the unspecified prompts. Previous work also develops multiple methods to identify the topic or concept of text input (Li et al., 2024a; Abraham et al., 2022; Blei et al., 2003). However, our paper is the first to focus on assessing whether the models use shortcuts at the general concept level.

Data	Amazon Shoe: Size			Amazon Shoe: Color			Amazon Shoe: Style			IMDB: Acting		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
$\mathcal{D}_{ori}$	2.11	**57.94**	55.94	**1.38**	57.19	55.50	11.56	57.18	56.12	3.70	88.49	91.24
$\mathcal{D}^c_{down-bal}$	2.25	57.19	54.29	2.54	56.71	55.70	9.79	**57.45**	56.02	2.04	91.03	91.92
$\mathcal{D}^c_{up-bal}$	**1.20**	56.72	55.64	2.06	**57.44**	**57.01**	**9.16**	56.41	**58.84**	**1.87**	**91.63**	**92.10**
$\mathcal{D}^c_{mask}$	1.88	57.31	**56.75**	5.94	57.40	56.21	10.29	56.80	55.12	4.75	90.67	91.95
	IMDB: Comedy			IMDB: Music			Yelp: Food			Yelp: Service		
$\mathcal{D}_{ori}$	2.51	91.62	91.85	12.07	91.55	88.93	3.42	93.86	**97.09**	2.44	93.33	97.23
$\mathcal{D}^c_{down-bal}$	-0.37	90.86	**92.88**	7.71	91.03	92.29	**-0.88**	93.63	95.35	1.57	93.78	97.02
$\mathcal{D}^c_{up-bal}$	**-0.32**	**91.65**	92.74	**4.05**	90.19	**92.80**	2.86	**94.15**	96.85	**0.39**	93.84	97.04
$\mathcal{D}^c_{mask}$	1.08	90.41	92.56	8.28	**91.88**	90.52	2.02	93.52	96.45	1.02	**94.67**	**97.41**

Table 6: **Performance of multiple shortcut mitigation methods (downsampling, upsampling and token removal).** Upsampling method with the counterfactual generated data can obtain the best average effects in the aspects of reducing bias and increasing the utility performance. $\mathcal{D}_{ori}$ represents fine-tuning on the $\mathcal{D}_{ori}$ dataset.

Spurious Correlation Mitigation An increasing number of methods have attempted to mitigate spurious correlations in models caused by bias in the dataset (Chew et al., 2023; Clark et al., 2020; Le Bras et al., 2020; Zhou and Bansal, 2020; Liu et al., 2021, 2023c; Zhu et al., 2023), by data augmentation (Jin et al., 2020; Alzantot et al., 2018; Wang et al., 2022; Minderer et al., 2020), data rebalancing (McCoy et al., 2019; Sharma et al., 2018; Zellers et al., 2019), multi-task learning (Tu et al., 2020), and model ensembling or adding regularization (Utama et al., 2020; He et al., 2019; Zhao et al., 2022; Liu et al., 2023d). To mitigate spurious correlations in a concept, we propose another data rebalancing method, which uses LLM to generate counterfactual sentences by injecting the concept and saves the human resource to compose them.

7 Conclusions

In this paper, we explore the spurious correlation at the general concept level in both fine-tuning and ICL settings. We find that LMs utilize the concept shortcut in training data (or in demonstrations) when inference on unseen data, and more biased training data (or prompts) lead to more biased predictions. To mitigate the learned shortcut, we propose a rebalancing method by adding counterfactual examples generated from ChatGPT to the training data, shown to be effective through extensive experiments. Our work indicates that LMs form strong spurious correlations on general concepts, encouraging future work to pay attention to unintended shortcut learning.

8 Limitations

Due to the limitation of the budget and the computation resource, we only fine-tuned the LLaMa2 7B model with the Lora method and used GPT3.5 for concept annotation. It could be interesting to fully fine-tune the LMs with a larger size. Moreover, in Section 3, we find that the ChatGPT annotation of the concept label still achieves slightly lower accuracy than human annotators. We can use a more advanced model, such as GPT4 (OpenAI, 2023), for annotation to increase the performance.

Our work focuses on five classification tasks. Three of them are binary classification tasks, and two are multiclass classification tasks. We apply the difference in accuracy for different groups (positive and negative) to measure bias at the concept level. Moreover, future work could extend our framework and generalize the measurement of concept bias to more complex tasks, such as the evaluation of LLM on QA tasks (Li et al., 2024b) or even on tasks with the vision language model (Wang et al., 2024b; Liu et al., 2023a; Wang et al., 2024a; Yue et al., 2023).

For in-context learning, we observe that the concept bias in the demonstrations leads to larger spurious correlations. However, we also detect that the balanced prompts cannot fully eliminate the bias, and we do not provide a method to mitigate this inner spurious correlation in LMs. We leave that direction to future work.

Acknowledgments

Zhou and Huang are supported by DARPA Transfer from Imprecise and Abstract Models to Autonomous Technologies (TIAMAT) 80321, National Science Foundation NSF-IIS-2147276 FAI, DOD-ONR-Office of Naval Research under award number N00014-22-1-2335, DOD-AFOSR-Air Force Office of Scientific Research under award number FA9550-23-1-0048, DOD-DARPA-Defense Advanced Research Projects Agency Guaranteeing AI Robustness against Deception (GARD) HR00112020007, Adobe, Capital One and JP Morgan faculty fellowships.

References

Eldar D Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior. *Advances in Neural Information Processing Systems*, 35:17582–17596.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*.

Ziv Bar-Joseph, David K Gifford, and Tommi S Jaakkola. 2001. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(suppl_1):S22–S29.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Angana Borah, Daria Pylypenko, Cristina Espana-Bonet, and Josef van Genabith. 2023. Measuring spurious correlation in classification:'clever hans' in translationese. *arXiv preprint arXiv:2308.13170*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Oscar Chew, Kuan-Hao Huang, Kai-Wei Chang, and Hsuan-Tien Lin. 2023. Understanding and mitigating spurious correlations in text classification. *arXiv preprint arXiv:2305.13654*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. *arXiv preprint arXiv:2011.03856*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chandra Kiran Reddy Evuru, Sreyan Ghosh, Sonal Kumar, Ramaneswaran S, Utkarsh Tyagi, and Dinesh Manocha. 2024. Coda: Constrained generation based data augmentation for low-resource nlp.

Yanbo Fang and Yongfeng Zhang. 2022. Data-efficient concept extraction from pre-trained language models for commonsense explanation generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5883–5893.

Dan Friedman, Alexander Wettig, and Danqi Chen. 2022. Finding dataset shortcuts with grammar induction. *arXiv preprint arXiv:2210.11560*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. *arXiv preprint arXiv:1908.10763*.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Jon Kleinberg, Jens Ludwig, Sendhil Mullainathan, and Ashesh Rambachan. 2018. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, pages 22–27. American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 989–1002, Online. Association for Computational Linguistics.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish

Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *International conference on machine learning*, pages 1078–1088. PMLR.

Zongxia Li, Andrew Mao, Daniel Stephens, Pranav Goel, Emily Walpole, Alden Dima, Juan Fung, and Jordan Boyd-Graber. 2024a. Improving the tenor of labeling: Re-evaluating topic models for content analysis.

Zongxia Li, Ishani Mondal, Yijun Liang, Huy Nghiem, and Jordan Lee Boyd-Graber. 2024b. Panda (pedantic answer-correctness determination and adjudication):improving automatic evaluation for question answering and text generation.

Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR.

Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2023a. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. *arXiv preprint arXiv:2310.14566*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiaoyu Liu, Hanlin Lu, Jianbo Yuan, and Xinyu Li. 2023c. Cat: Causal audio transformer for audio classification. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. 2024. Large language models and causal inference in collaboration: A comprehensive survey. *arXiv preprint arXiv:2403.09606*.

Xiaoyu Liu, Jiaxin Yuan, Bang An, Yuancheng Xu, Yifan Yang, and Furong Huang. 2023d. C-disentanglement: Discovering causally-independent generative factors under an inductive bias of confounder. *arXiv preprint arXiv:2310.17325*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*.

Matthias Minderer, Olivier Bachem, Neil Houlsby, and Michael Tschannen. 2020. Automatic shortcut removal for self-supervised representation learning. In *International Conference on Machine Learning*, pages 6927–6937. PMLR.

Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. 2020. Evaluating robustness to input perturbations for neural machine translation. *arXiv preprint arXiv:2005.00580*.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. 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.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. 2023. Measuring inductive biases of in-context learning with underspecified demonstrations. *arXiv preprint arXiv:2305.13299*.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. *arXiv preprint arXiv:2305.17256*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance. *arXiv preprint arXiv:2005.00315*.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022. 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. Association for Computational Linguistics.

Xiyao Wang, Jiuhai Chen, Zhaoyang Wang, Yuhang Zhou, Yiyang Zhou, Huaxiu Yao, Tianyi Zhou, Tom Goldstein, Parminder Bhatia, Furong Huang, and Cao Xiao. 2024a. Enhancing visual-language modality alignment in large vision language models via self-improvement.

Xiyao Wang, Yuhang Zhou, Xiaoyu Liu, Hongjin Lu, Yuancheng Xu, Feihong He, Jaehong Yoon, Taixi Lu, Gedas Bertasius, Mohit Bansal, et al. 2024b. Mementos: A comprehensive benchmark for multimodal large language model reasoning over image sequences. *arXiv preprint arXiv:2401.10529*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Robert Wolfe and Aylin Caliskan. 2021. Low frequency names exhibit bias and overfitting in contextualizing language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 518–532, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2023. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Michael Zhang, Nimit S Sohoni, Hongyang R Zhang, Chelsea Finn, and Christopher Ré. 2022. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. *arXiv preprint arXiv:2203.01517*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

Jieyu Zhao, Xuezhi Wang, Yao Qin, Jilin Chen, and Kai-Wei Chang. 2022. Investigating ensemble methods for model robustness improvement of text classifiers. *arXiv preprint arXiv:2210.16298*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. On large language models' selection bias in multi-choice questions. *arXiv preprint arXiv:2309.03882*.

Xiang Zhou and Mohit Bansal. 2020. Towards robustifying NLI models against lexical dataset biases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8759–8771, Online. Association for Computational Linguistics.

Yuhang Zhou, Suraj Maharjan, and Beiye Liu. 2023. Scalable prompt generation for semi-supervised learning with language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 758–769.

Jing Zhu, Yuhang Zhou, Vassilis N Ioannidis, Shengyi Qian, Wei Ai, Xiang Song, and Danai Koutra. 2023. Spottarget: Rethinking the effect of target edges for link prediction in graph neural networks. *arXiv preprint arXiv:2306.00899*.

A Implementation Details

A.1 Fine-tuning Experiments

We use the DistilBERT and LLAMA2 model (Sanh et al., 2019; Touvron et al., 2023) as our LMs for all of our fine-tuning experiments. For the DistilBERT model, we use AdamW as our optimizer with a learning rate of 2e−5 and a weight decay of 0.01 with linear scheduler, batch size of 16, and trained for 3 epochs. For the LLAMA2 model, we use AdamW as our optimizer with a learning rate of 2e−4, batch size of 32, warm-up ratio of 0.03, and trained for 3 epochs. We base our implementation on the Pytorch[2], Huggingface transformer[3] frameworks, and the LLAMA2 weights from Meta[4].

A.2 ICL Setup

We utilize greedy search in decoding for all ICL experiments and counterfactual data generation, except for the annotation of concepts for each text, where we use stochastic temperature sampling with the temperature value 0.7 to obtain diverse answers. The template of the prompts for the ICL experiments, concept annotations and counterfactual data generations are suggested in Table 10, Table 11 and Table 12.

We call the gpt-3.5-turbo (4k) function from OpenAI to generate the concept labels, ICL experiments and concept injection. The price of this API is $0.0015 / 1K tokens for inputs and $0.002 / 1K tokens for output. The total expenditure on API usage is about $ 300.00, including preliminary exploration.

B Prompt Details and Supplementary Results

In Table 7, we perform the same analysis on the BoolQ question and answering dataset for all experiments (ICL and fine-tuning) in Section 4. In Table 8 and Table 9, we repeat the experiments for fine-tuning LLAMA2 7B models for Section 4.1 and 5.2. In Table 10, 11, and 12, we present the details of the prompts in the annotation of the concept, the ICL experiments, and the countergactual sentence generation.

Distilbert	BoolQ Country (pos <neg)			BoolQ History (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**-1.22**	**61.23**	**60.95**	5.21	59.93	57.85
Trained on $\mathcal{D}^c_{biased}$	-18.90	55.92	55.46	50.63	57.37	55.22
Trained on $\mathcal{D}^c_{down-bal}$	4.84	57.93	58.79	-8.95	59.87	58.60
Trained on $\mathcal{D}^c_{up-bal}$	2.45	59.54	59.74	**-0.85**	60.13	59.70
Trained on $\mathcal{D}^c_{mask}$	2.90	60.71	59.69	-1.55	**60.94**	**60.84**
LLAMA2	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	-9.72	67.78	72.68	**0.82**	78.87	80.43
Trained on $\mathcal{D}^c_{biased}$	-9.67	67.23	70.86	2.73	75.86	78.79
Trained on $\mathcal{D}^c_{down-bal}$	-10.18	77.11	**81.61**	1.20	76.11	77.64
Trained on $\mathcal{D}^c_{up-bal}$	**-7.81**	77.30	78.16	-3.23	**80.62**	**82.40**
Trained on $\mathcal{D}^c_{mask}$	-10.55	**77.53**	79.10	-7.73	78.91	80.73
GPT3.5 ICL	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
ICL with $P_{balanced}$	**-1.80**	81.98	**83.99**	**-3.88**	**83.12**	**83.90**
ICL with P_{biased}	-2.90	**82.93**	83.11	-3.93	82.03	82.82

Table 7: Fine-tuning and ICL performance for all experiments in Section 4 on BoolQ dataset of DistilBert, LLAMA2 (fine-tuning) and GPT3.5 (ICL) models. The smaller absolute values of Bias@C (smaller bias) and larger values of Acc are in bold.

[2] https://pytorch.org/

[3] https://huggingface.co/

[4] https://ai.meta.com/llama/

Method	AS Size (pos <neg)			AS Color (pos >neg)			AS Style (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**-1.46**	59.46	**57.24**	**3.87**	**59.62**	59.07	**16.01**	59.54	**58.89**
Trained on $\mathcal{D}^c_{biased}$	-7.69	**59.57**	51.86	7.93	58.54	**59.43**	17.31	**59.91**	58.14
Method	IMDB Acting (pos >neg)			IMDB Comedy (pos >neg)			IMDB Music (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**3.51**	**95.87**	**97.55**	1.23	96.30	**97.55**	**5.35**	96.84	95.69
Trained on $\mathcal{D}^c_{biased}$	4.61	95.73	97.54	**0.40**	**96.79**	97.19	6.65	**96.88**	**96.24**
Method	Yelp Food (pos >neg)			Yelp Service (pos <neg)			Yelp Price (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**2.62**	**98.30**	**98.80**	1.41	**97.79**	**99.24**	**0.32**	**98.59**	**98.52**
Trained on $\mathcal{D}^c_{biased}$	2.64	98.23	98.80	**1.11**	97.78	99.19	0.78	98.37	98.37
Method	CeBaB Food (pos >neg)			CeBaB Service (pos >neg)			CeBaB Ambiance (pos >neg)		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
Trained on $\mathcal{D}_{ori}$	**3.04**	**74.01**	**75.57**	-3.44	**69.58**	**75.21**	**0.41**	**73.05**	**75.60**
Trained on $\mathcal{D}^c_{biased}$	3.67	61.96	65.09	**3.12**	65.05	67.22	1.35	68.97	73.63

Table 8: **Model fine-tuning performance with training on original dataset and concept biased dataset for LLAMA2 fine-tuning.** pos > neg: The number of positive texts is larger than the number of negative texts in the original data and in biased dataset, all texts containing this concept are positive, and vice versa for "pos < neg". The smaller absolute values of Bias@C (smaller bias) and larger values of Acc are in bold.

Method	Amazon Shoe: Size			Amazon Shoe: Color			Amazon Shoe: Style			IMDB: Acting			CeBaB: Food		
	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C	Bias@C	Acc@NoC	Acc@C
$\mathcal{D}_{ori}$	-1.46	59.46	57.24	3.87	59.62	59.07	16.01	**59.54**	58.89	3.51	95.87	97.55	3.04	**74.01**	**75.57**
$\mathcal{D}^c_{down-bal}$	2.88	59.17	**57.93**	4.10	59.57	58.48	11.62	58.13	**61.83**	2.28	**96.14**	97.59	2.16	68.86	71.09
$\mathcal{D}^c_{up-bal}$	**-0.62**	**59.94**	56.53	**-2.01**	**59.92**	**60.90**	**11.37**	59.32	60.43	**1.91**	96.12	97.72	1.83	70.71	74.75
$\mathcal{D}^c_{mask}$	-2.41	59.65	53.65	4.12	58.23	58.33	13.33	59.12	59.48	2.50	96.02	**97.73**	**0.02**	73.45	73.42
	IMDB: Comedy			IMDB: Music			Yelp: Food			Yelp: Service			CeBaB: Service		
$\mathcal{D}_{ori}$	1.23	96.30	97.55	5.35	96.84	95.69	2.62	98.30	98.80	1.41	97.79	99.24	-3.44	69.58	75.21
$\mathcal{D}^c_{down-bal}$	0.89	96.32	97.32	8.56	**97.27**	95.12	4.39	98.18	98.92	-0.35	**98.00**	99.20	-1.86	**70.44**	74.84
$\mathcal{D}^c_{up-bal}$	0.77	**97.59**	**97.74**	8.01	96.78	94.57	**1.83**	97.91	**99.04**	0.41	97.62	**99.30**	**0.21**	70.25	**75.88**
$\mathcal{D}^c_{mask}$	**0.53**	96.89	97.18	**2.68**	96.98	**96.71**	3.18	**98.36**	98.61	**0.08**	97.96	98.82	-0.90	69.83	75.05

Table 9: **Performance of multiple shortcut mitigation methods (downsampling, upsampling and token removal) for LLAMA2 fine-tuning.** Upsampling method with the counterfactual generated data can obtain the best average effects in the aspects of reducing bias and increasing the utility performance.

I will provide you 5 reviews in {dataset name} dataset. Please find the concepts explicitly mentioned in this review only from the set with three concepts: {candidate concepts}. Do not include other concepts. If you can not find any of these concepts in the concept set, please annotate this review with "none". Wrap your answer for a review in a word sequence separated by the comma and for each answer, start with a new line with an index.
Here are a few examples:
{demonstrations}
The output is:
{output concepts}
Here is the review list of 5 OpenTable reviews:
{text lists}
The output is:

Table 10: Prompt P_a for concept annotation in all datasets. {dataset name} and {candidate concepts} are placeholders to put the name of dataset and the candidate concepts. For example, for Amazon shoe dataset, they are "Amazon shoe" and "size, color, and style". {demonstrations} and {output concepts} are placeholders to add five demonstrations with provided ground-truth concept labels. {text lists} is a placeholder to add the text to be annotated.

Given a review, you need to predict whether the sentiment of the review is positive or negative. Here are the examples:

Review: {review 1} Sentiment label: {label 1}

Review: {review 2} Sentiment label: {label 2}

Review: {review 3} Sentiment label: {label 3}

Review: {review 4} Sentiment label: {label 4}

Review: {review 5} Sentiment label: {label 5}

Review: {review 6} Sentiment label: {label 6}

Review: {review 7} Sentiment label: {label 7}

Review: {review 8} Sentiment label: {label 8}

Here is the review to predict sentiment:

Review: {x_{test}} Sentiment label:

(a) Prompt $P_{balanced}$ or P_{biased} for IMDB and Yelp dataset.

Given a review, you need to predict whether the sentiment label of the review from 0 to 4, total 5 classes. Label 0 represents the most negative review and Label 4 represents the most positive review. Here are the examples:

Review: {review 1} Sentiment label: {label 1}

Review: {review 2} Sentiment label: {label 2}

Review: {review 3} Sentiment label: {label 3}

Review: {review 4} Sentiment label: {label 4}

Review: {review 5} Sentiment label: {label 5}

Review: {review 6} Sentiment label: {label 6}

Review: {review 7} Sentiment label: {label 7}

Review: {review 8} Sentiment label: {label 8}

Here is the review to predict sentiment:

Review: {x_{test}} Sentiment label:

(b) Prompt $P_{balanced}$ or P_{biased} for CeBaB and Amazon shoe dataset.

Table 11: Prompt $P_{balanced}$ or P_{biased} for the ICL experiments for all datasets. {review} and {label} is a placeholder to add 8 demonstrations with provided ground-truth sentiment labels for each dataset. {x_{test}} is the place to insert the predicted text.

Here are 5 exemplars with the {concept} concept:

{texts with concept}

Here are another 5 exemplars without the {concept} concept:

{texts without concept}

Please inject the “concept” concept into a statement and maintain the sentiment level of this statement.

The statement is:

{text for counterfactual}

The output statement with the {concept} concept is:

Table 12: Prompt P_i for counterfactual data generation in all datasets. {concept} are a placeholder to put the concept for generating the counterfactual data. {texts with concept} and {texts without concept} are placeholders to add five demonstrations with or without the concepts. for counterfactual} is a placeholder to add the text to make the counterfactual in the concept level.