Mitigating Spurious Correlation in Natural Language Understanding with Counterfactual Inference

Can Udomcharoenchaikit[†], **Wuttikorn Ponwitayarat**[†], **Patomporn Payoungkhamdee**[†],
Kanruethai Masuk[‡], **Weerayut Buaphet**[†], **Ekapol Chuangsuwanich**[♣], **Sarana Nutanong**[†]
[†]School of Information Science and Technology, VISTEC, Thailand
[‡]VISAI AI, Thailand
[♣]Department of Computer Engineering, Chulalongkorn University, Thailand
{canu_pro,wuttikornp.p_s22,patomporn.p_s21,kanruethaim_pro,
weerayut.b_s20,snutanon}@vistec.ac.th, ekapolc@cp.eng.chula.ac.th

Abstract

Despite their promising results on standard benchmarks, NLU models are still prone to make predictions based on shortcuts caused by unintended bias in the dataset. For example, an NLI model may use lexical overlap as a shortcut to make entailment predictions due to repetitive data generation patterns from annotators, also called annotation artifacts. In this paper, we propose a causal analysis framework to help debias NLU models. We show that (1) by defining causal relationships, we can introspect how much annotation artifacts affect the outcomes. (2) We can utilize counterfactual inference to mitigate bias with this knowledge. We found that viewing a model as a treatment can mitigate bias more effectively than viewing annotation artifacts as treatment. (3) In addition to bias mitigation, we can interpret how much each debiasing strategy is affected by annotation artifacts. Our experimental results show that using counterfactual inference can improve out-of-distribution performance in all settings while maintaining high in-distribution performance. [1]

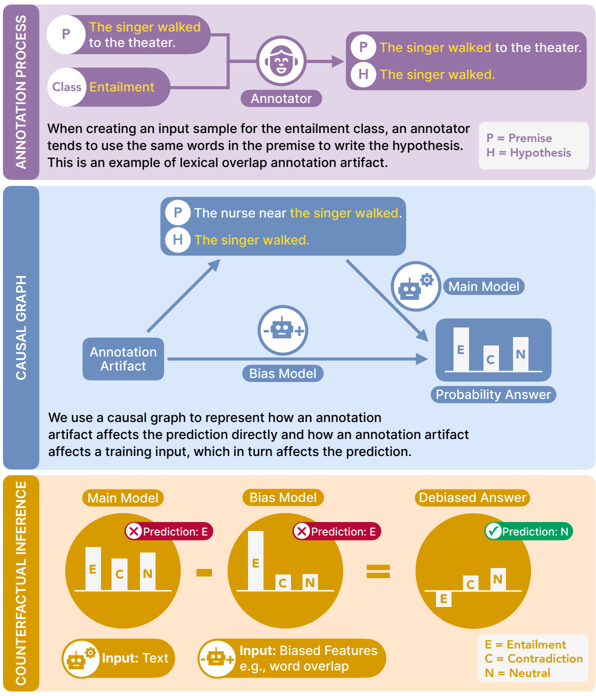
ANNOTATION PROCESS
P The singer walked to the theater.
Class Entailment
Annotator
P The singer walked to the theater.
H The singer walked.
When creating an input sample for the entailment class, an annotator tends to use the same words in the premise to write the hypothesis. This is an example of lexical overlap annotation artifact.
P = Premise
H = Hypothesis
CAUSAL GRAPH
P The nurse near the singer walked.
H The singer walked.
Main Model
Annotation Artifact
Bias Model
E C N
Probability Answer
We use a causal graph to represent how an annotation artifact affects the prediction directly and how an annotation artifact affects a training input, which in turn affects the prediction.
COUNTERFACTUAL INFERENCE
Main Model
Bias Model
Debiased Answer
Prediction: E
Prediction: E
Prediction: N
E C N
E C N
E C N
Input: Text
Input: Biased Features e.g., word overlap
E = Entailment
C = Contradiction
N = Neutral

Figure 1: (Top) Annotation Process: an annotator uses a specific strategy to write an input text for a designated class. (Mid) Causal graph represents causal relations between an annotation artifact, an input text, and a prediction. (Bottom) Example of counterfactual inference.

1 Introduction

Mitigating spurious correlations is crucial to the robustness of any learning method. Although deep learning models can perform well on conventional natural language understanding (NLU) benchmarks, researchers have found that these models leverage superficial patterns to produce correct predictions rather than learning the underlying tasks (Gururangan et al., 2018; McCoy et al., 2019). As a result, these models perform poorly when applied on out-of-distribution datasets, particularly on challenge sets that highlight models' reliance on spurious patterns by testing them on counterexamples. For example, McCoy et al. (2019) show that natural language inference (NLI) models wrongly exploit lexical overlap to make predictions for the entailment class. Consequently, when these models encounter any sample with high lexical overlap, their predictions are almost always entailment even though the sample is non-entailment. The following sample is an example of lexical overlap:

(1) Premise: The nurse near the singer walked.
Hypothesis: The singer walked.

While designing a data collection procedure for bias reduction is promising (Sakaguchi et al., 2020; Le Bras et al., 2020), creating a new dataset can be expensive and may also introduce new biases (Sharma et al., 2018). Therefore, it is important to develop learning and inference methods for

[1] The work was performed while Kanruethai Masuk was an intern at VISTEC. The code is available at https://github.com/c4n/debias_nlu.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11308–11321
December 7-11, 2022 ©2022 Association for Computational Linguistics

bias mitigation. The main advantage of debiasing from the model side is that we can mitigate bias without relying on the quality of data collection.

Early attempts at debiasing rely on intentionally creating a biased model to identify samples at risk of eliciting superficial patterns to produce results. One can then adjust how each training sample influences the training process accordingly.

Main approaches for NLU debiasing include: (1) reweighting loss for each training instance (Clark et al., 2019; Karimi Mahabadi et al., 2020; Ghaddar et al., 2021), (2) training a Product-of-Experts (PoE) between the main model and the biased model to encourage the main model to ignore biases (Clark et al., 2019), and (3) self-distillation where the soft labels from the teacher model are regularized by the scores from a biased model (Utama et al., 2020a; Du et al., 2021). These methods are effective in improving the performance in challenge sets. However, they cannot be applied to existing models without retraining. Our method, on the other hand, can operate at the inference stage and can be used to debias existing models hosted by a third party.

Recently, researchers have applied counterfactual inference for debiasing across various tasks in multiple fields, including computer vision, recommendation, as well as NLP (Niu et al., 2021; Wei et al., 2021; Wang et al., 2021; Qian et al., 2021; Nan et al., 2021). Counterfactual inference reduces bias by capturing its causal effect on predictions. It then utilizes this knowledge to assess how much bias it should remove. One common approach is to apply causal mediation analysis (CMA) to decompose effects from biases. This allows them to remove the direct bias effect from the total bias effect. Empirical results show that it improves robustness against biases across various tasks.

In this paper, we explore NLU debiasing from a causal perspective. Bias in NLU is often a product of specific annotation strategies to create training data known as annotation artifacts. As illustrated in Figure 1, by identifying the causal relationship between annotation artifacts and prediction outcomes, we can apply counterfactual inference to distinguish effects from annotation artifacts. With this knowledge, we can mitigate unintended effects caused by annotation artifacts, as well as quantify how much each debiasing method can prevent the unintended effect from influencing prediction outcomes.

Previous causal frameworks for debiasing often exploit CMA to find the total indirect effect (TIE), how much a bias affects an outcome indirectly through a mediator. Then, they use the the argmax of TIE as the prediction. Instead of using counterfactual inference for only prediction, we also extend it for bias analysis. The formalization of spurious correlation with causality allows us to not only reduce the prediction bias but also gives a methodological synergy when applying a causal analysis technique to interpret NLU models. In addition, we also experiment with a different causal viewpoint which considers models as treatment instead of bias as treatment.

We benchmark our counterfactual inference frameworks across three NLU tasks (NLI, fact verification, and paraphrase detection) on both in-distribution test sets and challenge sets. Our experimental results show that using counterfactual inference can significantly improve out-of-distribution robustness for all tasks. Moreover, the bias analysis through CMA shows that our counterfactual inference frameworks have smaller effects from annotation artifacts.

Our contributions are as follows:

1. We present the problem of annotation artifacts through a causal perspective and capture the annotation process with a causal graph.

2. We propose a counterfactual inference framework for NLU, which consistently improves robustness across multiple NLU tasks.

3. We provide a bias analysis using CMA to quantify the capability of debiasing methods.

2 Related Work

Earlier debiasing techniques involve reweighting the cross-entropy losses of bias-sensitive samples. Samples that can be handled effectively by a bias model are considered bias-sensitive. In contrast to the traditional training paradigm where each training example has the same importance, reweighting assigns different weights to examples based on their sensitivity to biases. A common way to reweight is to create a bias model trained on handcrafted features (Clark et al., 2019), and assign a weight to each training sample using the probability of the correct label from the bias model. Karimi Mahabadi et al. (2020) extended this idea by combining scores from multiple bias-only models. In addition, Karimi Mahabadi et al. (2020)

introduced a variant of focal loss (Lin et al., 2017) to leverage predictions from the bias model. Alternatively, Schuster et al. (2019) used n-grams co-occurrence to calculate the reweighting scores. However, recent research has been shifting toward acquiring reweighting scores without using domain knowledge and hand-crafted features (Utama et al., 2020b; Ghaddar et al., 2021).

Another prevailing approach is model ensembling. Clark et al. (2019) reinterpreted Product-of-Experts (PoE) (Hinton, 2002) for NLU debiasing. They used a bias model to create an ensemble with the main model with the aim that the main model learns all the information except the bias. The ensemble decreases the training loss for a sample that the bias model correctly predicts when updating the main model. Sanh et al. (2021) proposed a domain-knowledge-free approach where the bias model was a neural network with small parameters.

Utama et al. (2020a) proposed a self-distillation framework, confidence regularization (Conf-Reg), that used a bias model to produce confident scaling scores to calibrate the confidence of the model's predictions. Alternatively, Du et al. (2021) calculated the scaling scores from the local mutual information and integrated gradient explanation. Although these methods have successfully improved the robustness on the challenge sets, they can only be used to train new models. As a result, we cannot apply it to the existing models without retraining.

Counterfactual Inference: In contrast to traditional inference techniques in machine learning, where the argmax of the posterior probability is the prediction outcome, we can instead base the prediction on causal effects. Recent debiasing techniques integrate the idea of counterfactual inference into their frameworks across multiple tasks such as question answering (Niu and Zhang, 2021), visual question answering (Niu et al., 2021), text classification (Qian et al., 2021), recommendation (Wei et al., 2021; Wang et al., 2021), and information extraction (Nan et al., 2021).

One dominant technique is based on causal mediation analysis (CMA). It involves decomposing the total effect (TE) into pure direct effect (PDE) and total indirect effect (TIE). Niu et al. (2021), Wei et al. (2021), and Wang et al. (2021) make a prediction by selecting the class with the highest TIE. TE can also be decomposed into total direct effect (TDE) and pure indirect effect (PIE). Nan et al. (2021) utilize TDE to make predictions. Apart from debiasing, CMA can be used to analyze biases in transformer language models (Vig et al., 2020; Finlayson et al., 2021).

Alternatively, Qian et al. (2021) use TE to predict by calculating TE between a normal input (factual) and masked/partially masked input (counterfactual), without decomposing TE.

While reweighting, model ensembling, and self-distillation are effective NLU debiasing methods; we can further incorporate the idea of counterfactual inference to improve the debiasing performance further as well as using CMA as a tool to investigate the ability to debias.

3 Methodology

This section introduces key causal inference concepts used in this paper. First, we show how an NLU task can be presented via a causal graph to show causal relationships and implications of the biases in the causal relationships. Then, we discuss the causal effect and CMA—the central idea that we use to decompose effects from biases. We also discuss a counterfactual inference framework—an alternative to a standard inference framework that uses the argmax from the softmax as the prediction. Lastly, we discuss how we can apply CMA to quantify the bias effect of an annotation artifact.

3.1 Causal Graph: Text as a Mediator

Consider the MNLI dataset creation process: an annotator is given a text source premise and asked to write one hypothesis for each class. Due to the annotator's writing strategy, the annotator tends to use repetitive patterns to create an input text for each class. Machine learning models exploit these shallow repetitive patterns to make predictions. These shallow patterns are known as annotation artifacts. Hence, there is a causal relationship between an annotation artifact and an input text, where an annotation artifact causes an input text. We then use an input text to make a prediction. This produces a chain of causal relations.

We employ a causal graph to show causal relationships between variables as shown in Figure 2. Each node represents its corresponding variable. Each directed edge represents a direct effect. A represents an annotation artifact (e.g., lexical overlap, negation phrases). X represents an input text. Y represents a prediction.

As shown in previous studies (Gururangan et al., 2018), we can build a model that can predict Y

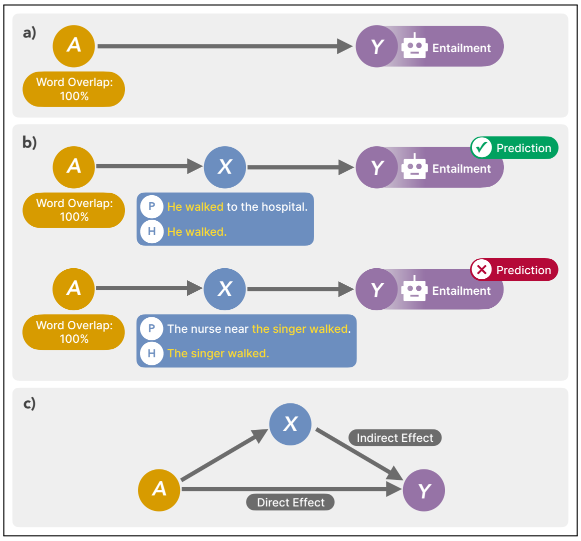

Figure 2: Causal graph for the NLI task (a) An annotation artifact A affects a prediction outcome Y. (b) An annotation artifact creates an input text X. X is then used to predict Y. We view X as a mediator between A and Y. (c) We can decompose an effect of A on Y into direct and indirect effects.

based on biased features alone. The performance of a biased model is considerably better than random. Therefore, any changes on A have an effect on Y. A directed edge from a variable A to a variable Y ($A \rightarrow Y$) illustrates that A is a direct cause of Y as shown in Figure 2a. Since A causes X, and we use X to predict Y in practice, we can also view X as a *mediator* between A and Y ($A \rightarrow X \rightarrow Y$).

A indirectly affects Y through an input text X. Figure 2b shows that learning from annotation artifacts does not always hold. Sample with 100% word overlap can also be non-entailment.

In addition, A has two directed edges pointing toward X and Y ($X \leftarrow A \rightarrow Y$), this represents a *common cause* that affects both X and Y causing a spurious correlation. Figure 2c shows that we can decouple the total effect of A on Y into direct and indirect effects. This causal graph allows us to apply causal mediation analysis to quantify and reduce the bias from the annotation strategy.

Structural Causal Model From the causal graph in Figure 2c, we can represent the causal graph using a structural causal model (SCM):

$$\begin{aligned} X_a &= x = f_X(A = a, N_X) \\ Y_{a,x} &= f_Y(A = a, X = x) \end{aligned} \tag{1}$$

A capital letter denotes a random variable (e.g., A), and a lowercase letter denotes an observed value (e.g., a). $Y_{a,x}$ denotes the prediction output of an individual sample with an annotation artifact $A = a$, and an input text $X = x$. $f_X(\cdot)$ and $f_Y(\cdot)$ refer to the structural causal equations of X and Y, respectively. $f_X(\cdot)$ represents an input text generation/collection process, where an annotator uses an annotation artifact $A = a$ to create an input text $X = x$. $f_Y(\cdot)$ represents a prediction function. N_X is the noise distribution of X. Hence, f_X can create different input texts with the same annotation strategy $A = a$.

We calculate $Y_{a,x}$ by combining the outputs from a bias model $P_b = \mathcal{F}_b(a)$ and a main model $P_m = \mathcal{F}_m(x)$ using the SUM fusion function h (Niu et al., 2021):

$$\begin{aligned} Y_{a,x} &= f_Y(A = a, X = x) \triangleq h\left(P_b, P_m\right) \\ &= \log \sigma\left(P_b + P_m\right) \end{aligned} \tag{2}$$

We can view A as a treatment variable. Assuming that $A = a$ represents treatment and $A = a^*$ represents no-treatment. For no-treatment, we replace an effect from the treatment with u a constant uniform distribution instead to represent the fact that the $A = a^*$ has no effect over a particular class. In order to include no-treatment condition into P_b, we represent P_b as follows.

$$P_b = \begin{cases} P_b = \mathcal{F}_b(a) & \text{if } A = a \\ P_b = u = \frac{1}{K} & \text{if } A = a^*, \end{cases} \tag{3}$$

where K denotes the number of classes.

When A is set to a^*, the variable X is also affected: $X_{a^*} = x^* = f_X(A = a^*)$. Hence, we represent P_m as follows.

$$P_m = \begin{cases} P_m = \mathcal{F}_m(x) & \text{if } X = x \\ P_m = u = \frac{1}{K} & \text{if } X = x^* \end{cases} \tag{4}$$

Note that both $\mathcal{F}_b(\cdot)$ and $\mathcal{F}_m(\cdot)$ can be parameterized and learned. In this work, $\mathcal{F}_b(\cdot)$ is a simple logistic regression model with only biased features, while $\mathcal{F}_m(\cdot)$ is a deep learning NLU model.

3.2 Causal Mediation Analysis

CMA often decomposes the total effect into two parts: direct and indirect effects. CMA allows us to quantify an indirect impact of a treatment on an outcome via a mediator.

3.2.1 Total Effect

For counterfactual inference, we focus on a causal effect, which is the difference between the outcomes of two hypothetical states of the world. For example, we can compare two situations of the treatment variable when $A = a$ and $A = a^*$ by computing the total effect (TE) of A:

$$TE_A = Y_{a,x} - Y_{a^*,x^*} \tag{5}$$

Y_{a^*,x^*} represents a situation where A is set to a^* which results in $X_{a^*} = x^*$ and Y_{a^*,x^*}. It can be represented as the following SCM:

$$\begin{aligned} X_{a^*} &= x^* = f_X(A = a^*) \\ Y_{a^*,x^*} &= f_Y(A = a^*, X = x^*) \end{aligned} \tag{6}$$

3.2.2 Decomposing Total Effect

The total effect can be decomposed into the following form (Robins and Greenland, 1992):

$$TE = PDE + TIE \tag{7}$$

The first component, pure direct effect (PDE), measures how treatment changes an outcome directly without acting through a mediator. It is calculated by applying a treatment while holding a mediator fixed. For the NLI task, PDE_A measures the difference in the entailment prediction Y when A changes from a^* to a, while X is set to a fixed constant value when $A = a^*$:

$$PDE_A = Y_{a,x^*} - Y_{a^*,x^*} \tag{8}$$

Y_{a,x^*} represents a situation where X is set to x^* while we keep $A = a$ as input for f_Y, it can be represented as the following SCM:

$$\begin{aligned} X_{a^*} &= x^* = f_X(A = a^*) \\ Y_{a,x^*} &= f_Y(A = a, X = x^*) \end{aligned} \tag{9}$$

The second component, total indirect effect (TIE), measures how treatment changes Y indirectly through X. We use this measure to examine how much our model (mediator) allows A to flow to Y. We calculate TIE_A by subtracting PDE_A from Eq. 7.

$$TIE_A = TE_A - PDE_A = Y_{a,x} - Y_{a,x^*} \tag{10}$$

Alternatively, we can view TIE_A as the difference on an outcome Y when we change the input of X from a^* to a while keeping everything else fixed as it would have been when $A = a$. One can also view $Y_{a,x}$ as an ensemble between the main and the bias models, while Y_{a,x^*} is the bias model fused with a constant.

3.3 Debiasing with Counterfactual Inference

At inference time, we make predictions based on causal effects. A prediction output is a class with the largest causal effect. In this work, we experiment with two causal queries: TIE_A and TE_{model}.

As shown in Eq. 10, TIE_A removes direct effect from bias by subtraction. Yet, the term $Y_{a,x}$ in TIE_A relies on an ensemble method to fuse outputs from a bias-only model.

For TE_{model} instead of viewing an annotation artifact as a treatment, we view a model as a treatment instead. From this viewpoint, we construct a causal query asking the following causal question: "what will the prediction be if we use a deep learning model instead of a bias model?"

$$TE_{model} = Y_{A\to X\to Y} - Y_{A\to Y}, \tag{11}$$

where $Y_{A\to X\to Y}$ is when we use the deep learning model.

$$\begin{aligned} X_a &= x = f_X(A = a) \\ Y_{A\to X\to Y} &= \mathcal{F}_m(X = x), \end{aligned} \tag{12}$$

and $Y_{A\to Y}$ is when we use the bias model.

$$Y_{A\to Y} = \mathcal{F}_b(A = a) \tag{13}$$

This differs from previous counterfactual methods that mainly use TIE and require an ensemble fusion that can fuse in biases. It also differs from Qian et al. (2021) by using the bias model as a counterfactual instead of masked inputs.

In practice, we only need to prepare the bias-only model $\mathcal{F}_b$ in order to apply counterfactual inference to any existing NLU model $\mathcal{F}_m$ without having to train or finetune $\mathcal{F}_m$.

Sharpness Control We control the strength of bias removal by using a learnable parameter c. For TIE_A, similar to Niu et al. (2021), we set the no-treatment constant u to c. For TE_{model}, we use c as a trade-off parameter to readjust the strength of $Y_{A\to Y}$ as follows.

$$TE_{model} = Y_{A\to X\to Y} - c * Y_{A\to Y} \tag{14}$$

We treat c as a hyperparameter in which we optimize it on a validation set. This is done by minimizing the kl-divergence between Y_{a,x^*} and $Y_{a,x}$ for TIE_A, and between softmax$(c * Y_{A\to Y})$ and $Y_{A\to X\to Y}$ for TE_{model}. Higher c suggests that, on average, $Y_{A\to Y}$ is less confident relative to $Y_{A\to X\to Y}$, and we may give it more weight.

3.4 Bias Effect Analysis Through CMA

We follow the CMA approach used by Vig et al. (2020) and Finlayson et al. (2021) for bias interpretation. In contrast to the previous works, we apply CMA to measure the effect of bias that flows through the whole model instead of some set of neurons. To measure the bias in the model, we use relative probabilities between the bias and anti-bias classes.

$$B = \frac{P_Y(\text{bias})}{P_Y(\text{anti-bias}_{\text{max}})}, \tag{15}$$

where $B > 1$ indicates a preference for the bias class and $B < 1$ indicates an anti-bias preference. If there are multiple anti-bias classes, we use the anti-bias class with the highest probability to represent an anti-bias probability. P_Y represents f_Y that has been normalized into a probability distribution using the softmax function. We use CMA to analyze the indirect effect of the known bias through the mediator on the outcome. By treating an input text and a model as a single mediation component, we can measure the effectiveness of each debiasing strategy. We are interested in how annotation artifacts affect the outcomes differently for each debiasing strategy. We examine the average total indirect effect (ATIE) of the bias class, which shows the effect of annotation artifacts on the biased class due to the mediated path.

$$ATIE = \mathbb{E}\left[\frac{B_{a,x} - B_{a,x^*}}{B_{a,x^*}}\right] = \mathbb{E}\left[\frac{B_{a,x}}{B_{a,x^*}}\right] - 1 \tag{16}$$

where B_{a,x^*} represents a situation when X is set to x^* as an input for f_Y. ATIE > 0 indicates that the mediated path prefers the biased class. While ATIE < 0 indicates that the mediated path prefers the anti-bias class. For a balanced test set, ATIE = 0 indicates no bias, and a good debiasing method should be able to lower the ATIE.

4 Experiments

Our experiments address the following research questions: (1) Can we exploit knowledge from causal relationships to debias NLU tasks? (2) Do different causal queries lead to different debias performances? (3) Can we measure the impact of annotation artifacts on the prediction outcomes? We benchmark our method on three English relation-identification tasks: natural language inference, fact verification, and paraphrase identification. NLI represents a generic benchmark for testing a machine's ability to identify a relationship between two texts. The other two tasks benchmark relationship-identification ability in real-world applications. Each task contains both standard in-distribution and challenge datasets to test the generalization ability. We train all our models only on the in-distribution datasets and test them on both standard benchmark datasets and challenge sets.

4.1 Experimental Setup

4.1.1 Datasets

Natural Language Inference We use the MNLI 1.0 dataset (Williams et al., 2018) to train, validate (MNLI-matched), and test (MNLI-mismatched) our method. To measure the robustness against spurious correlations, we use HANS (McCoy et al., 2019) as a challenge set.

Fact Verification We use the FEVER dataset (Thorne et al., 2018). We randomly split 5,000 samples from the original training data for the validation set. We use FEVER Symmetric (Schuster et al., 2019) as a challenge set.

Paraphrase Identification We use QQP[2] as a standard benchmark. Since there is no standard train/test split, we divide the original dataset into validation and testing data where each of them contains 5,000 pair of sentences. We use PAWS (Zhang et al., 2019) as a challenge set.

4.1.2 Implementation details

Main Model We apply the debiasing methods on the BERT base model (uncased) (Devlin et al., 2019). The model performs well on the three NLU tasks, but previous studies show that it relies on superficial clues (e.g., McCoy et al., 2019).

We use the contextualized embedding at the [CLS] location from the BERT model as an input to the feedforward layer with tanh activation, then pass the feedforward's output through the softmax layer to calculate the prediction. Similar to previous studies (Clark et al., 2019), we train the model for three epochs using AdamW optimizer (Loshchilov and Hutter, 2019) with the weight decay of 0.1. For the MNLI training set, we use the learning rate of 5e-5. For FEVER and QQP training sets, we follow Utama et al. (2020a,b) and use the learning rate of 2e-5.

[2] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

We use the slanted triangular learning rate schedule with 0.06 fraction of the steps to increase the learning rate. The batch size is 32. We also train with automatic mixed-precision training.

Bias Model We use a simple logistic regression model with only bias features. For NLI and paraphrase detection tasks, we use the following bias features: (1) whether the input text has the subsequence heuristic, (2) whether the input text has the constituent heuristic, and (3) the lexical overlap fraction. For the fact verification task, we extract the following superficial features: the number of negation phrases, the top 50 uni-grams and bigrams with the highest local mutual information, the lexical overlap, and the entity overlap fractions between claim and evidence.

4.1.3 Competitive Methods

In order to compare with previous methods, we reimplement three popular approaches for NLU debiasing: reweighting, product-of-experts, and self-distillation. We also show that using our counterfactual inference framework on top of these methods can significantly improve the results on the challenge sets. In addition, we also include the results from previous to studies in Table 1 for comparison.

Reweighting We use inverse probability weighting to reweight training samples. We scale the loss by $\frac{1}{\mathcal{F}_{b_y}}$ which is an inverse of a probability assigned to the correct label by the bias model.

Product-of-Experts We reimplement Clark et al. (2019)'s PoE model using our bias model. We train the main model by creating an ensemble between the main and the bias models. PoE combines the two models by calculating the element-wise product between their prediction outcomes in the logarithmic space as follows.

$$\hat{p} = \operatorname{softmax}\left(\log\left(\mathcal{F}_m\right) + \log\left(\mathcal{F}_b\right)\right) \tag{17}$$

Then we calculate the cross-entropy loss by comparing $\hat{p}$ with the ground truth. We only update the weights of the main model and only use the main model at inference time.

Self-Distillation We reimplement Utama et al. (2020a)'s Conf-Reg by using our bias model. Conf-Reg makes the main model less confident on samples that the bias model can predict correctly with high confidence. The scaling function $\mathcal{S}$ for distilling soft labels from the teacher model $\mathcal{F}_t$ for confidence regularization is as follows.

$$\mathcal{S}\left(\mathcal{F}_t, \mathcal{F}_{b_y}\right)_j = \frac{\mathcal{F}_{t_j}^{\left(1-\mathcal{F}_{b_y}\right)}}{\sum_{k=1}^{K} \mathcal{F}_{t_k}^{\left(1-\mathcal{F}_{b_y}\right)}} \tag{18}$$

For each label j = 1, . . . , K, we use $\mathcal{S}\left(\mathcal{F}_t, \mathcal{F}_{b_y}\right)$ as a soft label for the cross-entropy loss.

4.2 Experimental Results

4.2.1 Effectiveness of Counterfactual Inference Debias

We investigate whether causal knowledge can be used to improve NLU robustness via counterfactual inference. In addition, we compare the effectiveness of two different causal queries. Table 1 shows the results on three NLU tasks. Both counterfactual inference methods can significantly[3] improve the robustness for all out-of-distribution test sets on top of almost all debiasing methods across multiple tasks. Without any debiasing training methods, both causal queries alone obtain superior results than the vanilla baseline. When combined with a debiasing training method, they can provide superior results over previous NLU debiasing literature for the NLI and QQP tasks. TIE$_A$ inference improves the performance on all challenge sets without hurting the performances on the standard test sets. TE$_{model}$ inference further improves the performances on all challenge sets; however, the performances on in-distribution test sets are lower. Nevertheless, we can use TIE$_A$ to show the indirect effects of annotation artifacts that flow through the model.

It is important to note that both TIE$_A$ and TE$_{model}$ can be applied to existing NLU models without retraining as shown in Appendix A.1.

4.3 Bias Analysis

We conduct fine-grained analysis on HANS. This challenge set annotates each sample with its subcase, which contains information about its lexical overlap type and grammatical pattern.

Table 2 shows ATIE and accuracy for each class and for each heuristic. For almost all cases, the ATIE is positive, indicating that the previous debiasing methods still contain a substantial preference for the bias class (entailment). The only case where the ATIE is negative for the previous debiasing

[3] In order to statistically compare debiasing methods, we use Almost Stochastic Dominance test (Dror et al., 2019) with the significant level of 0.05.

Method	MNLI (acc)		Fever (acc)			QQP (MaF1)	
	dev-mm	HANS	dev	symm v1	symm v2	dev	PAWS
Baseline	84.83	65.55	86.25	57.41	63.85	**94.03**	31.34
Baseline + TIE$_A$	84.82	66.66*	86.46*	57.82*	64.19*	93.97	31.96*
Baseline + TE$_{model}$	84.78	68.08*	**86.47***	58.02*	64.30*	93.81	35.12*
Reweighting	84.84	66.96	82.53	60.73	62.92	93.10	44.16
Reweighting + TIE$_A$	84.84	68.18*	81.87	61.06*	63.32*	93.02	44.92*
Reweighting + TE$_{model}$	84.80	69.47*	81.50	61.37*	63.32*	92.77	49.36*
PoE (Clark et al., 2019)	84.21	69.86	85.45	60.92	65.37	93.23	43.97
PoE + TIE$_A$	84.13	70.94*	85.04	61.51*	65.73*	93.16	45.08*
PoE + TE$_{model}$	83.94	**72.24***	84.98	**61.70***	65.67*	92.81	50.70*
Conf-Reg (Utama et al., 2020a)	**85.10**	66.32	86.38	60.08	66.18	92.73	32.69
Conf-Reg + TIE$_A$	85.03	68.23*	85.01	61.12*	66.24*	92.62	35.03*
Conf-Reg + TE$_{model}$	84.92	70.99*	80.74	61.59*	**66.26***	91.51	**58.50***
Results reported in reference papers							
PoE (Clark et al., 2019)	82.97	67.92	-	-	-	-	-
Conf-Reg (Utama et al., 2020a)	84.8	69.1	86.4	60.5	66.2	90.45	55.4
Debiased Focal Loss (Karimi Mahabadi et al., 2020)	82.76	71.95	83.07	64.02	-	-	-
PoE (Sanh et al., 2021)	81.35	68.77	81.97	59.95	-	-	-

Table 1: Performance results evaluated on in-distribution and out-of-distribution (grey columns) test sets across 3 NLU tasks. We compare the effectiveness of the two causal queries compared to the standard inference. We report the average scores across five runs on different random seeds. * denotes a significant improvement of a counterfactual inference over the standard inference method.

methods is the non-entailment class of the lexical overlap heuristic, in which the non-entailment accuracy is the highest for all methods. The overall accuracy for each heuristic is correlated with ATIE. Methods with lower ATIE tend to have higher overall accuracy. The two causal queries, TIE_A and TE_{model}, can improve accuracy for all heuristics. Counterfactual inference can greatly reduce ATIE for all cases. However, measuring ATIE gives a preference towards TIE_A over TE_{model}. This is due to the fact when we fuse the bias and the main models to create $Y_{a,x}$ for TIE_A, the confidence of a prediction becomes lower.

Comparing to the baseline, all debiasing methods provide the biggest improvement for the constituent heuristic. The subsequence heuristic is the most challenging of the three syntactic heuristics. ATIE remains high even for the non-entailment samples. Subcases with high error rates are often syntactically ambiguous, for example, NP/S and NP/Z. They are known as "garden-path" sentences that cause reading difficulty even for humans. We include detailed results for all subcases along with examples in Appendix A.8.

5 Conclusion

We introduce a counterfactual framework for debiasing NLU models. Our framework can improve robustness across three NLU tasks. It also allows us to analyze the impact of annotation artifacts.

	Class	Lexical Overlap		Subsequence		Constituent	
		ATIE	ACC	ATIE	ACC	ATIE	ACC
baseline	E	0.2797	97.54	0.3009	99.48	0.3020	99.79
	N	-0.0879	70.99	0.2301	10.63	0.2142	14.84
	overall	0.0959	84.26	0.2655	55.05	0.2581	57.32
reweighting	E	0.2854	97.90	0.3031	99.58	0.3026	99.76
	N	-0.0805	69.77	0.2324	11.03	0.1963	23.74
	overall	0.1024	83.83	0.2678	55.31	0.2495	61.75
PoE	E	0.2584	94.65	0.2911	98.93	0.2920	99.07
	N	-0.1053	75.72	0.2101	13.76	0.1590	37.00
	overall	0.0766	85.18	0.2506	56.35	0.2255	68.04
Conf-Reg	E	0.1872	95.56	0.2193	99.48	0.2189	99.34
	N	-0.0970	72.77	0.1570	8.18	0.1252	22.60
	overall	0.0451	84.16	0.1882	53.83	0.1720	60.97
PoE + TIE$_A$	E	0.0202	93.42	0.0233	98.54	0.0235	98.48
	N	-0.0188	77.66	0.0156	16.22	0.0119	41.32
	overall	0.0007	85.54	0.0194	57.38	0.0177	69.90
PoE + TE$_{model}$	E	0.0561	91.71	0.0676	98.10	0.0673	97.26
	N	-0.0723	79.90	0.0391	19.82	0.0232	46.65
	overall	-0.0081	85.81	0.0534	58.96	0.0452	71.95

Table 2: ATIE and accuracy of each syntactic heuristic in the HANS dataset. E denotes the entailment class, and N denotes the non-entailment class.

In contrast to previous debiasing NLU literature that focuses on reweighting, model-ensemble, and self-distillation, we present a causal perspective on this problem that can be combined with previous methods to enhance the performances further. Counterfactual inference gives us the flexibility to apply it to any existing model without changing its parameters by focusing on the inference stage. Unlike previous counterfactual inference for debiasing literature, where the main focus is using TIE for inference only, we also used TIE for analysis.

Moreover, we also provide another causal viewpoint by looking at the model as the treatment to construct a new causal query that outperforms TIE across multiple challenge sets.

Causality can help us improve the generalization and robustness of NLU models. In the future, we will utilize causality techniques to identify and remove biases in NLU tasks.

Limitations

The decomposition of the total effect is only guaranteed in linear models (Pearl, 2001). However, all deep learning models are non-linear. Nevertheless, Vig et al. (2020) found that the sum of direct and indirect effects in GPT2-small, a non-linear model, can roughly approximate the total effect.

Our bias effect analysis has a preference toward low confidence models. One of the risks is that one could maliciously use our approach to validate bias models with low confidence predictions.

Acknowledgements

The authors would like to thank Jitkapat Sawatphol for his diligent proofreading of this paper, and Bundit Boonyarit for his help with Figures 1 and 2.

References

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep dominance - how to properly compare deep neural models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785, Florence, Italy. Association for Computational Linguistics.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800.

Guido W. Imbens and Donald B. Rubin. 2015. *Causality: The Basic Framework*, page 3–22. Cambridge University Press.

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

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.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Guoshun Nan, Jiaqi Zeng, Rui Qiao, Zhijiang Guo, and Wei Lu. 2021. Uncovering main causalities for long-tailed information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9683–9695, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12700–12710.

Yulei Niu and Hanwang Zhang. 2021. Introspective distillation for robust question answering. In *Thirty-Fifth Conference on Neural Information Processing Systems*.

Judea Pearl. 2001. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5434–5445, Online. Association for Computational Linguistics.

James M Robins and Sander Greenland. 1992. Identifiability and exchangeability for direct and indirect effects. *Epidemiology*, pages 143–155.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. 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 3419–3425, Hong Kong, China. Association for Computational Linguistics.

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, Melbourne, Australia. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. 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. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1288–1297, New York, NY, USA. Association for Computing Machinery.

Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, KDD '21, page 1791–1800, New York, NY, USA. Association for Computing Machinery.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix

A.1 Effect of Counterfactual Inference on Existing Models

We download existing models from Huggingface (Wolf et al., 2020) and apply counterfactual inference debias directly on these models without finetuning them further. Table 3 shows that counterfactual inference can consistently improve the out-of-distribution performance for all three models without updating their parameters.

	dev-mm	HANS
ishan/bert-base-uncased-mnli	83.45	56.72
$+TIE_A$	83.47	58.47
$+TE_{model}$	83.45	60.57
roberta-large-mnli	88.61	73.13
$+TIE_A$	88.65	73.96
$+TE_{model}$	88.63	75.07
facebook/bart-large-mnli	88.52	71.36
$+TIE_A$	88.44	72.39
$+TE_{model}$	88.42	73.71

Table 3: Results of the existing NLI models

A.2 Data Statistics

Table 4 shows data statistics for all the datasets used in our experiments. It is important to note that the QQP in-distribution test set and the PAWS challenge set have a considerable imbalance in the ratio between non-paraphrase and paraphrase pairs. Hence, we use Macro F1 to report the scores for the paraphrase identification task.

MNLI	
train	392,702
dev-m (validation)	10,000
dev-mm (in-distribution test)	10,000
HANS (challenge set)	30,000
FEVER	
train	242,911
validation	5,000
dev (in-distribution test)	16,664
symmetric v1 (challenge set)	717
symmetric v2 (challenge set)	712
QQP	
train	394,287
validation	5,000
dev (in-distribution test)	5,000
PAWS (challenge set)	677

Table 4: Number of samples in each dataset used in our experiments

A.3 Licenses

Datasets: For the MNLI dataset, most of the data are available under OANC's license. HANS is available under the MIT license. These licenses allow users to freely use, share, and distribute the datasets under non-restrictive agreements.

Softwares: For all tasks, we use AllenNLP (Gardner et al., 2018) to train all deep learning models. And we use scikit-learn (Pedregosa et al., 2011) to train the bias models. These libraries are available under permissive licenses Apache license 2.0 and BSD license. These licenses allow both academic and commercial usages. To make our code accessible and compatible with these licenses, we release our code under Apache license 2.0.

A.4 Computing Infrastructure

We train all 110M parameters bert-base-uncased models on the NVIDIA DGX-1 with 8 Volta V100 GPUs. We train each model on one GPU at a time. We train four debiasing training methods on three different tasks, and for each model we train it using five different random seeds. It requires us approximately 50 minutes to train one model for one random seed on the MNLI training set. Hence, we approximate that it could take at least 50-60 GPU hours to train all the models needed to replicate our results. The training times for shallow models are negligible, since all of them are simple logistic regression models.

A.5 Counterfactual Inference Visualization

Figure 3 shows how we intervene the causal graph to create counterfactuals (Y_{a,x^*}, and Y_{a^*,x^*}) for calculating causal effects (TE, PDE, and TIE).

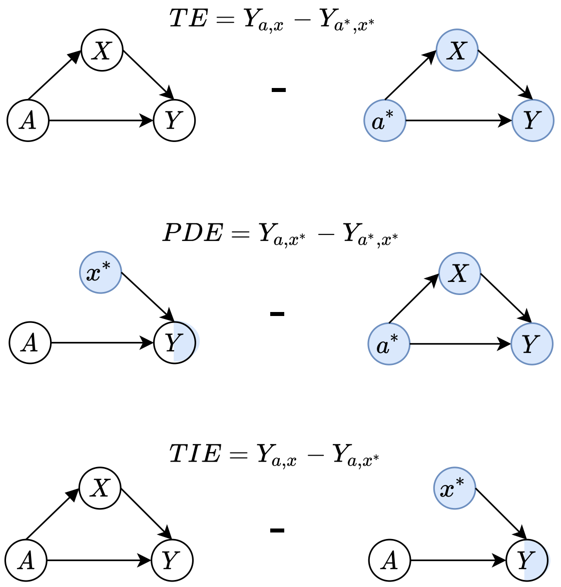
$TE = Y_{a,x} - Y_{a^*,x^*}$
X
A
Y
-
X
a^*
Y
$PDE = Y_{a,x^*} - Y_{a^*,x^*}$
x^*
A
Y
-
X
a^*
Y
$TIE = Y_{a,x} - Y_{a,x^*}$
X
A
Y
-
x^*
A
Y

Figure 3: Graphical interpretation of TE, PDE, and TIE

A.6 Reweighting Techniques Comparison

We use inverse probability weighting (IPW) technique to reweight the loss function, instead of using the reweight baseline (Clark et al., 2019) and the debiased focal loss (Karimi Mahabadi et al., 2020). Table 5 shows that IPW is superior to the two previous reweighting techniques. This result gives us an empirical reason to use IPW over the two other reweighting methods.

	dev-mm	HANS
Reweight Baseline (Clark et al., 2019)	84.88	65.48
Debiased Focal Loss (Karimi Mahabadi et al., 2020)	84.95	65.31
IPW (Ours)	84.84	66.96

Table 5: Results of different reweighting methods on the MNLI's mismatched development set and the HANS challenge set.

A.7 Effect of Sharpness Control

Table 6 compares the results between counterfactual inference with and without sharpness control. Sharpness control limits the strength of bias removal. It balances the trade-off between the bias and the anti-bias performances. Without sharpness control, the results on the in-distribution test sets can drop drastically. For example, on FEVER in-distribution test set, the accuracy of Conf-Reg with TE inference can drop from 79.22 to 66.83.

A.8 Fine-grained and Qualitative Analyses of HANS

We utilize ATIE to analyse the baseline BERT-based model along with the three debiasing techniques. Table 7 shows results for all subcases. The differences in performances between subcases are large. Subcases with high error rates are challenging not only because they contain annotation artifacts, but they are also syntactically ambiguous. Since the subsequence heuristic has the lowest accuracy, we examine the three worst performing subsequence subcases (sn_NP/S, sn_NP/Z, sn_past_participle) and provide qualitative analysis along with the examples. These subcases contain "garden-path" sentences. A garden path sentence contains a group of words with temporary ambiguity which can be resolved by reading an entire sentence.

sn_NP/S

(A1) Premise: The artist believed the scientists slept.
Hypothesis: The artist believed the scientists.

In example A1, when we only consider "The artist believed the scientists" part of the premise. We would conclude that this example is an entailment class. However, by adding an extra verb "slept" to the end of the premise, the answer changes from entailment to non-entailment.

Method	MNLI		Fever			QQP	
	dev-mm	HANS	dev	symm v1	symm v2	dev	PAWS
w/ sharpness correction							
baseline + TIE$_A$	84.82	66.66	86.46	57.82	64.19	93.97	31.96
Reweighting + TIE$_A$	84.84	68.18	81.87	61.06	63.32	93.02	44.92
PoE + TIE$_A$	84.13	70.94	85.04	61.51	65.73	93.16	45.08
Conf-Reg + TIE$_A$	85.03	68.23	85.01	61.12	66.24	92.62	35.03
wo/ sharpness correction							
baseline + TIE$_A$	84.77 ↓	65.85 ↓	86.36 ↓	57.66 ↓	64.02 ↓	94.03 ↑	31.34 ↓
Reweighting + TIE$_A$	84.84 ~	67.34 ↓	82.24 ↑	60.78 ↓	63.12 ↓	93.10 ↑	44.16 ↓
PoE + TIE$_A$	84.31 ↑	69.69 ↓	85.25 ↑	61.26 ↓	65.56 ↓	93.23 ↑	43.97 ↓
Conf-Reg + TIE$_A$	85.16 ↑	66.15 ↓	86.42 ↑	60.39 ↓	66.29 ↑	92.73 ↑	32.74 ↓
w/ sharpness correction							
baseline + TE$_{model}$	84.78	68.08	86.47	58.02	64.30	93.81	35.12
Reweighting + TE$_{model}$	84.80	69.47	81.66	61.28	63.34	92.77	49.36
PoE + TE$_{model}$	83.94	72.24	84.98	61.70	65.67	92.81	50.70
Conf-Reg + TE$_{model}$	84.92	70.99	80.74	61.59	66.26	91.51	58.50
wo/ sharpness correction							
baseline + TE$_{model}$	84.12 ↓	70.46 ↑	85.96 ↓	61.31 ↑	65.90 ↑	93.41 ↓	44.44 ↑
Reweighting + TE$_{model}$	84.69 ↓	71.71 ↑	73.72 ↓	62.99 ↑	62.30 ↓	91.88 ↓	55.97 ↑
PoE + TE$_{model}$	83.75 ↓	73.71 ↑	79.49 ↓	62.82 ↑	64.41 ↓	91.93 ↓	56.69 ↑
Conf-Reg + TE$_{model}$	84.53 ↓	72.49 ↑	67.41 ↓	61.42 ↓	60.76 ↓	80.52 ↓	42.21 ↓

Table 6: The effects of sharpness correction compared to the inference performance without the correction. The vertical arrow (↓, ↑) denotes the performance dropping and increasing when the sharpness correction is removed.

sn_NP/Z

(A2) Premise: After the author paid the actor ran.
Hypothesis: The author paid the actor.

For the premise in example A2, the dependent and the independent clauses are joined without proper punctuation causing confusion even for the human. We hypothesize that by recreating sn_NP/Z with proper punctuation, the models would have performed better.

sn_past_participle

(A3) Premise: The senators paid in the office danced.
Hypothesis: The senators paid in the office.

The example A3 contains the main verb/reduce relative (MV/RR) ambiguity. The word “paid” is a main verb in the hypothesis. Still, adding “danced” as a main verb at the end of the premise makes “paid” in the sentence become a reduced relative clause for “who were paid.” In other words, "paid" is a past participle of the premise sentence.

A.9 Definition of Causal Effect

Here we give a definition of causal effect. A causal effect is the difference between two hypothetical states of the world. A causal effect can reveal a comparison between two different treatments. Given a treatment T (1:treatment , 0: no treatment) and an outcome Y, for an individual instance, we say that the treatment T has a causal effect on the outcome Y, if $Y^{t=1} \neq Y^{t=0}$.

Consider a scenario where one has a headache and is deciding whether one should buy a drug to help one feel better (Imbens and Rubin, 2015). If the headache goes away after taking the drug ($t = 1$), we cannot say that the drug has a causal effect on the headache. What if the headache also goes away without taking the drug ($t = 0$)? In this case, the drug has no causal effect on the headache $Y^{t=1} = Y^{t=0}$. In contrast, the drug has a causal effect on the headache if the headache does not go away without taking the drug $Y^{t=1} \neq Y^{t=0}$.

	Baseline		Reweight		PoE		Conf-Reg		PoE + TIE		PoE + TE	
subcase	ATIE	acc.	ATIE	acc.	ATIE	acc.	ATIE	acc.	ATIE	acc.	ATIE	acc.
ce_adverb	0.3134	100.00	0.3141	100.00	0.3094	100.00	0.2305	100.00	0.0248	100.00	0.0729	100.00
ce_after_since_clause	0.3130	100.00	0.3137	100.00	0.3100	100.00	0.2211	100.00	0.0248	100.00	0.0730	100.00
ce_conjunction	0.3127	100.00	0.3133	100.00	0.3079	100.00	0.2252	100.00	0.0246	100.00	0.0722	99.96
ce_embedded_under_since	0.2970	99.30	0.2989	99.56	0.2735	97.96	0.2023	98.18	0.0211	97.02	0.0591	94.66
ce_embedded_under_verb	0.2944	99.68	0.2894	99.26	0.2588	97.40	0.1806	98.52	0.0221	95.36	0.0592	91.68
cn_adverb	0.2316	8.46	0.1933	21.00	0.1220	37.88	0.1135	18.62	0.0146	44.20	0.0278	51.56
cn_after_if_clause	0.3033	0.38	0.3029	0.54	0.2963	0.42	0.2024	1.28	0.0237	0.66	0.0685	1.28
cn_disjunction	0.2860	1.00	0.2657	6.40	0.2078	18.08	0.1713	4.64	0.0162	21.50	0.0406	26.26
cn_embedded_under_if	0.1345	25.26	0.0920	38.84	0.0206	65.84	0.0584	39.02	-0.0018	72.14	-0.0233	80.02
cn_embedded_under_verb	0.1088	39.12	0.0757	51.90	0.0362	62.80	0.0448	49.42	0.0068	68.12	0.0022	74.12
le_around_prepositional_phrase	0.3055	99.84	0.3072	99.84	0.2852	98.88	0.1844	99.50	0.0224	98.56	0.0637	97.94
le_around_relative_clause	0.3011	99.06	0.3025	98.68	0.2833	97.86	0.1885	98.28	0.0225	97.44	0.0638	96.58
le_conjunction	0.2637	93.72	0.2716	94.44	0.2102	84.10	0.1310	83.16	0.0128	81.38	0.0304	77.56
le_passive	0.3113	99.98	0.3111	99.92	0.3103	99.80	0.2626	100.00	0.0245	99.72	0.0724	99.64
le_relative_clause	0.2694	95.10	0.2783	96.62	0.2454	92.62	0.1923	96.84	0.0189	90.02	0.0502	86.84
ln_conjunction	-0.0218	82.44	-0.0266	83.72	-0.0583	90.76	-0.0342	87.20	-0.0248	92.52	-0.0925	94.64
ln_passive	0.2427	11.94	0.2526	8.82	0.2132	17.68	0.1791	8.08	0.0120	20.24	0.0290	23.56
ln_preposition	-0.0367	86.84	-0.0333	85.70	-0.0511	89.48	-0.0403	89.12	-0.0260	91.22	-0.0960	93.30
ln_relative_clause	-0.0170	80.62	-0.0078	77.82	-0.0359	84.74	-0.0286	83.88	-0.0232	87.60	-0.0873	90.60
ln_subject/object_swap	-0.0633	93.10	-0.0620	92.78	-0.0786	95.92	-0.0715	95.56	-0.0321	96.72	-0.1146	97.42
se_PP_on_obj	0.3101	99.96	0.3115	100.00	0.2988	99.76	0.2097	100.00	0.0236	99.54	0.0685	99.32
se_adjective	0.3129	100.00	0.3135	100.00	0.3101	100.00	0.2335	100.00	0.0248	100.00	0.0731	100.00
se_conjunction	0.2887	97.44	0.2926	97.92	0.2657	95.16	0.1871	97.40	0.0195	93.62	0.0542	91.90
se_relative_clause_on_obj	0.3113	100.00	0.3128	100.00	0.3017	99.74	0.2133	99.98	0.0238	99.54	0.0696	99.38
se_understood_object	0.3105	100.00	0.3132	100.00	0.3097	100.00	0.2344	100.00	0.0245	99.98	0.0724	99.92
sn_NP/S	0.3049	0.32	0.3046	1.02	0.2889	0.78	0.1962	0.50	0.0234	0.90	0.0665	1.20
sn_NP/Z	0.2895	0.80	0.2944	0.90	0.2548	5.24	0.1724	5.16	0.0180	7.22	0.0492	9.88
sn_PP_on_subject	0.1691	29.88	0.1683	30.50	0.1461	32.74	0.1177	23.42	0.0045	36.36	0.0026	41.34
sn_past_participle	0.2994	1.00	0.2934	2.24	0.2443	6.66	0.1850	1.44	0.0209	8.98	0.0549	13.28
sn_relative_clause_on_subject	0.1941	21.14	0.1966	20.50	0.1723	23.38	0.1499	10.40	0.0111	27.64	0.0225	33.38

Table 7: ATIE and accuracy of each subtask and each debiasing method. The highlighted rows are the three subsequence subcases with the poorest performances. Note that the first letter of the subcase's name denotes its heuristic (**l**exical overlap, **s**ubsequence, **c**onstituent). The second letter denotes its ground-truth (**e**ntailment, **n**on-entailment).