# Identifying and Mitigating Annotation Bias in Natural Language Understanding using Causal Mediation Analysis

**Sitiporn Sae Lim**[†*], **Can Udomcharoenchaikit**[†*], **Peerat Limkonchotiwat**[†],
**Ekapol Chuangsuwanich**[♣], **Sarana Nutanong**[†]
[†]School of Information Science and Technology, VISTEC, Thailand
[♣]Department of Computer Engineering, Faculty of Engineering,
Chulalongkorn University, Thailand
{sitiporn.s,canu_pro,peerat.l_s19,snutanon}@vistec.ac.th,
ekapolc@cp.eng.chula.ac.th

## Abstract

NLU models have achieved promising results on standard benchmarks. Despite state-of-the-art accuracy, analysis reveals that many models make predictions using annotation bias rather than the properties we intend the model to learn. Consequently, these models perform poorly on out-of-distribution datasets. Recent advances in bias mitigation show that annotation bias can be alleviated through fine-tuning debiasing objectives. In this paper, we apply causal mediation analysis to gauge how much each model component mediates annotation biases. Using the knowledge from the causal analysis, we improve the model's robustness against annotation bias through two bias mitigation methods: causal-grounded masking and gradient unlearning. Causal analysis reveals that biases concentrated in specific components, even after employing other training-time debiasing techniques. Manipulating these components by masking out neurons' activations or updating specific weight blocks both demonstrably improve robustness against annotation artifacts. [1]

## 1 Introduction

Current Natural Language Understanding (NLU) models obtain state-of-the-art accuracy on in-distribution benchmarks. However, researchers (Gururangan et al., 2018; McCoy et al., 2019) have found that these models utilize annotation bias to make predictions, negatively affecting the models' generalizability. Consequently, bias analysis and mitigation are crucial topics in NLU.

One popular approach to assessing bias in NLU models is behavioral analysis. This approach views models as test subjects and evaluates them by collecting their behavioral data in specific test cases without considering their internal components. Results from behavioral analyses (Sanchez et al., 2018; Gururangan et al., 2018; McCoy et al., 2019) reveal that NLU models rely on superficial patterns to make predictions[2]. Although using such superficial patterns as shortcuts can be advantageous when used in-distribution, they can mislead models to wrong answers when used in out-of-distribution settings.

Based on behavioral analysis, researchers develop techniques to address annotation artifacts directly by considering an entire NLU model as a black box. While this approach is beneficial in terms of simplicity, results from structural analysis show that learned features are localized within specific groups of components (Giulianelli et al., 2018). Consequently, we suggest that the study of bias mitigation would greatly benefit from analyzing an NLU model as a collection of components to direct the mitigation effort strategically.

Causal mediation analysis (CMA) has been applied to conduct structural analysis on NLP tasks (Vig et al., 2020; Finlayson et al., 2021; Mueller et al., 2022; Geiger et al., 2021; Wang et al., 2023b; Meng et al., 2022; Stolfo et al., 2023). CMA provides a framework to identify specific components within a model that contribute to a behavior of interest. Nevertheless, CMA for NLU is underexplored, and common counterfactual generation techniques for CMA rely on automatic text edits for each sample. This practice is inadequate to capture the spurious correlations we are interested in. Furthermore, most CMA works within NLP are limited to behavioral-structural analysis without using this valuable knowledge to enhance the performance of the models.

To accommodate the need to interpret and mitigate biases, we argue that an ideal CMA framework

[*] Equal Contribution.

[1] The code is available at https://github.com/sitiporn/DebiasNeuro-components.

[2] Our preliminary analysis also suggests that a BERT-based model is more confident when predicting samples correctly identified by the bias model compared to those predicted incorrectly. The average Softmax outputs for ground-truth answers are 0.84 and 0.78 for these two groups of samples.

*Findings of the Association for Computational Linguistics ACL 2024*, pages 11548–11563
August 11-16, 2024 ©2024 Association for Computational Linguistics

for NLU should have two following features:

1. The ability to specify model components responsible for mediating biases caused by annotation artifacts in NLU tasks.
2. The ability to apply the knowledge acquired from CMA to mitigate bias for the existing models without the need to train from scratch.

We formulate the impact of annotation artifacts in the NLU problem using a causal graph, as shown in Figure 1. With this paradigm, we model the annotation artifact, neuron, and classification outputs as a control variable $A$, mediator $Z$, and response variable $Y$, respectively. This allows us to generate a biased counterfactual sample at $A$ and observe how $Z$ produces an indirect effect on the classification output $Y$.

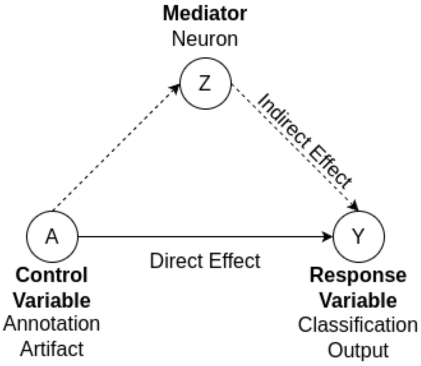


Figure 1: Causal graph for an NLU task: An annotation artifact $A$ has an influence over the neuron $Z$. $Z$ is then used to predict $Y$. Therefore, We view $Z$ as a mediator between $A$ and $Y$. We can decompose an effect of $A$ on $Y$ into direct and indirect effects.

To show whether CMA can identify the model components mediating high-bias information, we apply CMA in the bias mitigation process to reduce the impact of annotation bias. We experimented with two bias mitigation methods: *causal-grounded masking* and *gradient unlearning*. The crux of our methods lies in solely utilizing the top components that mediate the most effect from the annotation artifact, which are identified by the CMA process. The bias mitigation method serves *not only* as a method for improving performance on challenge sets *but also* as an evaluation tool that demonstrates the effectiveness of the CMA process. Furthermore, the bias mitigation methods can be applied post-hoc to any existing models without training from scratch or adding an additional module. In this way, users can import our set of weights and use the exact same code for their inference module without requiring an additional engineering effort.

We apply the CMA to analyze the NLI models using the standard MNLI dataset. In addition, our bias mitigation modules show improvements on the standard NLU spurious correlation mitigation benchmark that includes the following tasks: NLI, fact verification, and paraphrase detection.

CMA shows that annotation bias effects are mainly concentrated in specific components, especially in neurons of the last few layers of the model. Our experimental results on bias mitigation demonstrate that modifying components identified by the CMA process can effectively improve the robustness against annotation artifacts.

## 2 Related Work

### 2.1 Causal Mediation Analysis

Causality studies how a response variable changes when we apply an intervention. Causal mediation analysis (CMA) is a tool to study the effect of mediators on a response variable (Pearl, 2001). In the context of machine learning and NLP, Vig et al. (2020) proposed a CMA framework to investigate how each mechanism within Transformers is affected by gender bias. This framework views each model component, such as a neuron, as a mediator. This inspires various NLP works to apply the CMA method to investigate how specific components in LMs mediate an attribute that researchers are interested.

Prominent CMA examples include grammatical inflections in the subject-verb agreement task (Finlayson et al., 2021; Mueller et al., 2022), an abstract causal process in natural language inference (Geiger et al., 2021), names in indirect object identification (Wang et al., 2023b), facts in text generation (Meng et al., 2022), and operands in arithmetic reasoning (Stolfo et al., 2023). However, most of these studies do not use the knowledge gained from the causal analysis to improve the performance.

Only Meng et al. (2022) utilize this analysis by updating weights in the MLP layers, while Chintam et al. (2023) apply the idea of counterfactuals from Vig et al. (2020) without engaging in causal analysis to learn a mask that identifies the components of a model responsible for bias. They design a loss function based on a comparison between factual and counterfactual samples to learn a mask for debiasing.

For annotation artifacts on NLU tasks, Udomcharoenchaikit et al. (2022) identify causal graph that reveals the relationships between annotation artifacts and the prediction outcomes for NLU tasks. They also provide a CMA by viewing text as the mediator. However, their CMA method views the entire model as one component and, therefore, cannot provide a structural analysis for NLU tasks.

### 2.2 Bias Mitigation Methods in NLU

A common approach to mitigating bias in NLU is reweighting the cross-entropy losses. Instead of treating all training samples equally, the reweighting paradigm allocates different weights to samples based on how much they are affected by biases. Typically, a reweighting method involves the following steps: training a bias model and subsequently assigning weights to training examples based on the prediction scores generated by the bias model for the ground-truth labels (Clark et al., 2019; Schuster et al., 2019; Karimi Mahabadi et al., 2020; Utama et al., 2020b; Ghaddar et al., 2021).

Another common technique is model ensembling. Researchers have applied Product-of-Experts (PoE) (Hinton, 2002) for NLU debiasing (Clark et al., 2019; Sanh et al., 2021; Wang et al., 2023a; Du et al., 2023). They integrate a bias model with the main model to create an ensemble, with the goal of ensuring that the main model learns all the relevant information except for the biases. Self-distillation frameworks have also been proposed. They use a bias model (Utama et al., 2020a) or an integrated gradient score (Du et al., 2021) to produce confident scaling scores to adjust the confidence of the model's predictions.

Recently, a post-hoc counterfactual inference approach has been explored. Tian et al. (2022) and Udomcharoenchaikit et al. (2022) replace the inference step based on the Softmax output with the counterfactual inference output.

### 2.3 Discussion

Although these bias mitigation methods can obtain reasonable results on in-distribution datasets and improve performance on challenge sets, they can only be used to train new models. They address the whole model instead of specific components responsible for bias sensitivity. As a result, we cannot apply it to the existing models without retraining from scratch.

Recently, Meissner et al. (2022) show that we can alter specific parameters of the model to improve the robustness in NLU. Building on this insight, our proposed methods can still improve on Meissner et al. (2022) by further refining the activation outcomes or the parameters of components that are proned to bias.

## 3 Proposed Method

In this section, we formulate a method to identify model components prone to bias and mitigate bias in these components accordingly. As a structural analysis method, *Causal Mediation Analysis (CMA)* provides an effective means to analyze behavioral-structural relations within a pretrained language model (PLM). In particular, this method has been used to investigate specific information in each PLM component (Vig et al., 2020; Finlayson et al., 2021; Mueller et al., 2022). However, no prior works explore how to manipulate the bias-containing components to mitigate spurious correlation in NLU tasks.

The overview of our proposed framework is shown in Figure 2. Sec. 3.1 presents our counterfactual generation method, enabling us to exploit known bias features to identify bias samples. Sec. 3.2 describes the process that employed the generated counterfactuals to pinpoint components mediating biases. Sec. 3.3 presents our bias mitigation methods that exploit the knowledge of these biased components to mitigate annotation biases.

### 3.1 Counterfactual Generation

To study how each neuron mediates annotation artifacts, we assign a counterfactual value $z_j^*$ to each neuron in the model. A popular method to create counterfactual samples is automatic text editing. For instance, in gender bias research, one can simply replace the subject in the sentence to examine the bias. This allows us to compare whether the pronoun will be more likely to be "he" or "she" when we change the subject of the sentence, as shown in the following example.

(1) Prompt: An engineer said that [blank].
Counterfactual: The woman said that [blank].

However, for an NLU task, such replacement does not generate a sample that reflects an annotation artifact (e.g., word overlap), as shown in the following example.

(2) Premise: A black race car starts up in front of a crowd of people.
Hypothesis: A man is driving down a lonely road.

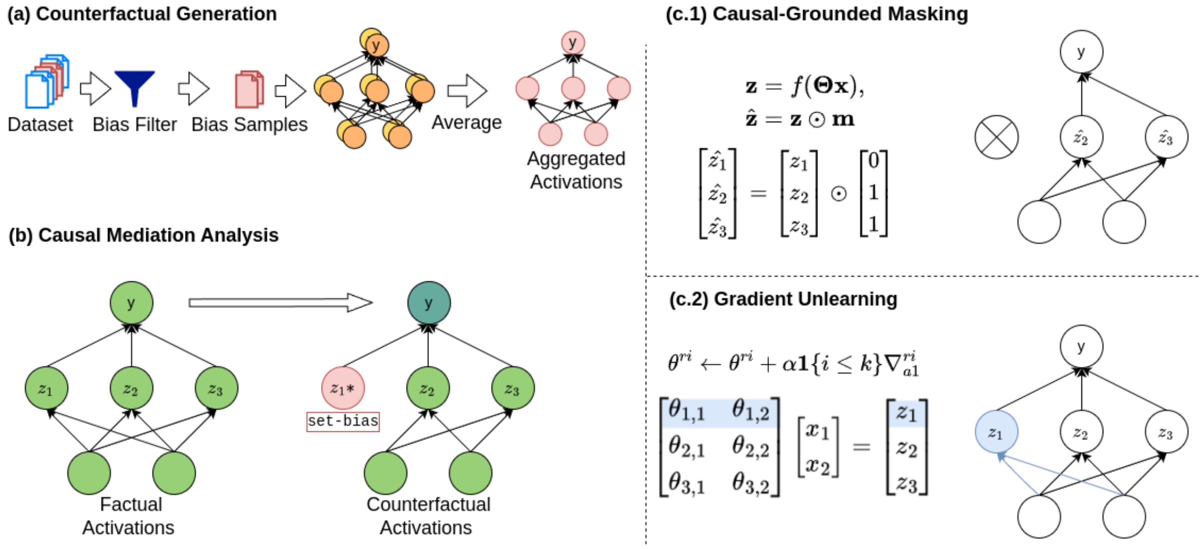

(a) Counterfactual Generation
Dataset
Bias Filter
Bias Samples
Average
Aggregated Activations
(b) Causal Mediation Analysis
Factual Activations
Counterfactual Activations
set-bias
(c.1) Causal-Grounded Masking
$\mathbf{z} = f(\boldsymbol{\Theta}\mathbf{x})$,
$\hat{\mathbf{z}} = \mathbf{z} \odot \mathbf{m}$
(c.2) Gradient Unlearning
$\theta^{ri} \leftarrow \theta^{ri} + \alpha \mathbf{1}\{i \leq k\}\nabla^{ri}_{a1}$

Figure 2: The CMA Bias Mitigation Framework: (a) We start by collecting neurons' activations based on bias inputs, then create an aggregated representation for the whole network. (b) We create a counterfactual network by using `set-bias` operation. Then, we compare the intervention outcome with the original outcome. (c) We can then apply one of the following bias mitigation methods to neurons with the highest indirect effects: (c.1) We select the top neurons mediating the highest effects from the annotation artifact and then mask them by multiplying with zero. (c.2) We only update weights that are tied directly to the top neurons.

In contrast to previous research (Vig et al., 2020; Finlayson et al., 2021), our method does not rely on automatic text editing to create a counterfactual scenario for NLU tasks. This allows us to cover a wider range of NLU tasks, e.g., NLI, fact verification, and paraphrase identification.

We propose a generalized counterfactual generation approach for annotation artifacts in NLU. Our approach focuses on generating a counterfactual state at the representation level instead of the textual input level. First, we create *a counterfactual state* by aggregating the activation outcomes from the samples of interest, which we select by choosing samples that contain the most bias-leaning features. For instance, in the NLI task, we select samples with high word overlap (See Sec. 4.3). Second, we define an intervention operation, `set-bias`, to influence the model to move toward the bias direction. We use the neuron activation values from the counterfactual state to replace those in the factual scenario. Using activation outcomes from the counterfactual state, we can perform causal inference operations without creating a counterfactual through text modification.

### 3.2 Causal Mediation Analysis for NLU

Let us consider the annotation process of the standard NLI dataset, MNLI. Based on a provided premise, an annotator generates one hypothesis for each class. Previous investigations (Gururangan et al., 2018; McCoy et al., 2019) suggest that annotators use specific annotation strategies that may introduce superficial patterns into their generated hypotheses. These shallow patterns can indirectly affect predictions through neurons in deep learning models, leading to poor performance on out-of-distribution data. These patterns introduced by the annotator's writing strategy are called "annotation artifacts".

To facilitate the discussion of our methodology, we adopt a causal graph to denote causal relationships in an NLU task. Figure 1 displays the causal relations and the implications of annotation artifacts within those relations. An annotation artifact has an influence over each neuron, and the neuron is used to make a prediction. Therefore, we can see the neuron $Z$ as a mediator between the annotation artifact $A$ and the output $Y$ ($A \rightarrow Z \rightarrow Y$).

The central idea is to use CMA to quantify how each specific model component mediates biases in NLU tasks. We can identify the impact of particular model components on model predictions by assessing both direct and indirect effects of interventions on the model inputs. We use natural indirect effect (NIE) to quantify how much an intervention on an annotation artifact can change an outcome indirectly through specific model components (intermediate variables). We compute NIE by setting $z$ to a value that it would be under an intervention, then compare it to the scenario where this intervention is not applied.

In this research, we apply intervention through two do-operations: (i) `set-bias`: replace a normal text pair input with a counterfactual input (Sec. 3.1); (ii) `null`: do nothing to the input. We then define $\boldsymbol{y}_a$ as the value that y would be for an input $u$ under the intervention $do(\boldsymbol{a} = a)$.

To study how each neuron mediates annotation artifacts, we apply an intervention $do(z_j = z_j^*)$ to an individual neuron $z_j$, where $z_j^*$ is the counterfactual value of that neuron.

$$\text{NIE} = E_u \left[ \frac{\boldsymbol{y}_{\text{null}, \boldsymbol{z}_{\text{set-bias}}(u)}(u)}{\boldsymbol{y}_{\text{null}}(u)} - 1 \right] \tag{1}$$

NIE can be interpreted as the change in the amount of bias when the original input $u$ remains unchanged while modifying the value of neuron $z_i$ to $z_i^*$. This work focuses only on neurons in each transformer layer except for the embedding and classification layers.

## 3.3 Bias Mitigation Methods

As discussed in the previous subsection, we can quantify how much each model component mediates biases using CMA. This allows us to apply post-hoc bias mitigation methods to highly biased components, which is advantageous in terms of applicability to existing models with little or no training costs. This subsection presents two post-hoc bias mitigation alternatives: (1) causal-grounded masking and (2) gradient unlearning. These two methods also allow us to select the neurons with and without the knowledge of CMA to assess the merit of CMA in an ablation study.

### 3.3.1 Causal-Grounded Masking

Let us now consider how to integrate the knowledge obtained from CMA to mitigate biases. In this work, we focus only on neurons in each transformers' layer except for embeddings and classification layers. We obtain a CMA result from the process discussed in Section 3.2. Then, we introduce an activation mask $\mathbf{m}$ assigned according to the NIE scores. The main idea is to mask out the top neurons have the highest mediated effect on the biases.

$$\mathbf{z} = f(\Theta \mathbf{x}), \tag{2}$$

$$\hat{\mathbf{z}} = \mathbf{z} \odot \mathbf{m} \tag{3}$$

Let $\mathbf{z}$ be a generic activation outcome (neurons), $\mathbf{x}$ be an arbitrary input, and $\Theta$ be a generic weight matrix. $\odot$ denotes the Hadamard product. $\hat{\mathbf{z}}$ is a masked activation outcome.

We determine the activation mask $\mathbf{m}$ according to the NIE score of each element $m_i$.

$$m_i = \begin{cases} 0 & \text{if } NIE\,(z_i) \geq \tau \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where $\tau$ is a threshold equivalent to the 700-th highest NIE score. We select 700 neurons with the highest NIE values to mask because this provides a result on the MNLI-matched dataset within an acceptable trade-off range (See Appendix A.5 for further explanation). Similar to Dropout (Srivastava et al., 2014), we scale up the activation outcomes of the remaining neurons by $1/p$, which is the inverse proportion of the unmasked neurons.

### 3.3.2 Gradient Unlearning

Similar to the masking method, we obtain a CMA result from the process described in Section 3.2. Then, we introduce a partition selection scheme for selecting weight blocks to unlearn bias. For each neuron, there is a weight block responsible for its activation value. We select these weight blocks according to the NIE scores of the neurons.

To explain how we choose weight blocks, we use a fully connected layer as an example.

$$\underbrace{\begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix}}_{\mathbf{z} \in R^{m \times 1}} = \underbrace{\begin{bmatrix} -\theta_1^\top - \\ -\theta_2^\top - \\ \vdots \\ -\theta_m^\top - \end{bmatrix}}_{\Theta \in R^{m \times d}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}}_{\mathbf{x} \in R^{d \times 1}} \tag{5}$$

Given a subset of bias neurons obtained using CMA, we update those neurons by selectively applying gradient updates. We select the neuron $z_j$ according to its NIE, where the NIE must be greater or equal to the top-700th neuron with the highest NIE. If the selected neuron is $z_j$, the neuron $z_j$ in Eq. 5 is calculated by $z_j = \theta_j^\top \mathbf{x}$. Then, only the selected weight block $\theta_j$ will be updated.

We reverse the direction of gradient $\nabla_j$ of $\theta_j$ on the selected text samples in our training set. Note that we only update the immediate weight blocks that are directly responsible for the target activation outcomes while keeping other weight blocks frozen.

We adopt the First-order Gradient Optimization step by Yu et al. (2023) to approximate the bias gradient used to unlearn annotation biases. We only further train the model on a small number of selected samples called the "advantaged samples"Yu et al. (2023), which are expected to be more preferred by a bias model. To determine the type of samples, we relabel samples of sentences based on inferences from two models, considering samples with correct prediction by the bias model and incorrect prediction by the main model to be the "advantaged samples". We use only advantaged samples to train the main model, moving parameters in the direction that increases the loss on the advantaged samples. We select and sort the top-k neurons with the highest NIEs $\{z^{r1}, z^{r2}, z^{r3}, \ldots, z^{rk}\}$, where $r1, r2, \ldots, rk$ represent the ordering.

$$\theta^{ri} \leftarrow \theta^{ri} + \alpha \mathbf{1}\{i \leq k\} \nabla_{a1}^{ri} \tag{6}$$

where $a1$ refers to the advantaged sample, and $\alpha$ refers to the learning rate. We set $k$ to 700.

# 4 Experimental Setup

## 4.1 NLU tasks

We conduct experiments on three NLU tasks that are commonly used for bias mitigation evaluation: natural language inference (NLI), fact verification, and paraphrase identification (See Appendix A.1 for descriptions of the datasets).

## 4.2 Main and Bias Models

For the main model, we study the bert-base-uncased model (Devlin et al., 2019). The model performs effectively in the three NLU tasks; however, it still relies on surface-level cues (Gururangan et al., 2018; McCoy et al., 2019). Hence, recent studies on NLU debiasing often benchmark their methods using the BERT base model. For bias models, we use hand-crafted features based on known biases to train logistic regression models for NLI and paraphrase identification tasks. For fact verification, we train the bert-base-uncased model using only claims as inputs. Further implementation details are provided in Appendix. A.4.

## 4.3 CMA Implementation details

**Counterfactual Generation.** As NLU tasks inherently involve classification, we exclusively focus on the activations of the transformer model at the [CLS] token position. For each task, we generate a counterfactual embedding to represent an annotation bias with respect to its training dataset. We sample the same number of examples from each class to ensure that the counterfactual representation does not skew toward a particular class because of the number of filtered samples. We sample exclusively from a validation dataset.

- **NLI:** We retain samples with word overlap above the 95th percentile to create an aggregated representation (115 samples in total).
- **Fact Verification:** We retain samples that do not contain any negation in their claims, and have two or more bi-grams that are in the list of top-50 bi-grams with the highest local mutual information (LMI) for the support class (69 samples in total).
- **Paraphrase Identification:** Zhang et al. (2019) observe that out of ~1000 sentence pairs with the same BoW, only 20% of them are not paraphrases. Identical BoW is a subcase of word overlap, which also aligns with the fact that the positions of the two text pairs are exchangeable without changing the answer. We retain samples with the top percentile of word overlap to create an aggregated representation (64 samples in total).

## 4.4 Gradient Unlearning Implementation

We use a specific bias model for each task to find advantage examples. For each task, we select samples within the class that the bias model performs best on the in-distribution training set. In the selected class, we filter these samples again by selecting samples where the main model predicts incorrectly, but the bias model can predict correctly.

## 4.5 Competitive Methods

We reimplement two popular approaches for NLU debiasing: reweighting and product-of-experts. We also show that using our CMA bias mitigation framework on top of these methods can significantly improve the results on the challenge sets.

**Reweighting.** We follow the strong baseline provided by Clark et al. (2019) and scale the loss by $1 - \mathcal{F}_{b_y}$ which scales down the loss based on the probability assigned to the ground truth label by the bias model ($\mathcal{F}_b$).

**Product-of-Experts.** We reimplement Clark et al. (2019)'s PoE approach. We train the main model ($\mathcal{F}_m$) by forming an ensemble with a bias model. The PoE method integrates these two models by

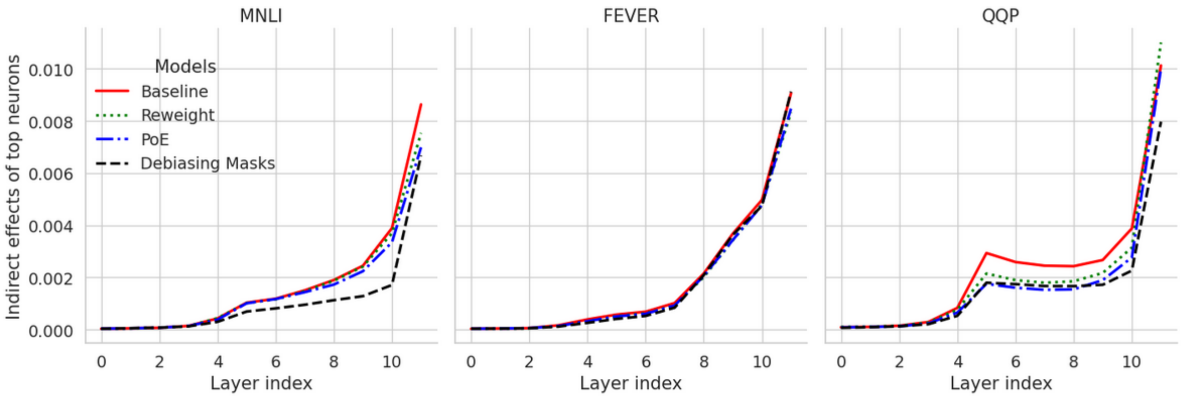


Figure 3: The natural indirect effects of top 5% neurons on the MNLI, FEVER, and QQP datasets. Final layers have the largest effects across the three datasets. Note that we do not include the embedding and classification layers.

computing the element-wise product of their prediction outcomes in logarithmic space.

$$\hat{p} = \mathrm{softmax}\left(\log\left(\mathcal{F}_m\right) + \log\left(\mathcal{F}_b\right)\right) \qquad (7)$$

Next, we compute the cross-entropy loss by comparing $\hat{p}$ with the ground truth. We update only the main model's weights and use only the main model for making predictions at inference time.

**Debiasing Masks.** We reimplement Meissner et al. (2022)'s Debiasing Masks. We train the masks to prune the original weights of a fine-tuned model with focal loss following Clark et al. (2019) and use a bias model to adjust this loss, which is equivalent to reweighting. Meissner et al. (2022) combine a debiasing loss, a biased model, and a score-based pruning technique similar to Zhao et al. (2020) and Sanh et al. (2020) with the addition of weight-freezing. It learns masks used to prune bias-inducing parameters.

## 5 Experimental Results

In this section, we show the results of causal mediation analysis (CMA) on NLU tasks (Section 5.1). To verify that the CMA can pinpoint neurons that mediate annotation biases, we also provide bias mitigation results based on manipulating neurons identified by the CMA as having high bias (Section 5.2). Moreover, we conduct ablation studies on the bias mitigation step to show that selecting neurons and "advantaged samples" based on the knowledge of annotation artifacts is an important factor in improving the robustness (Section 5.3). In addition, we examine the effectiveness of mitigating the model-level effect of annotation artifacts on the prediction outcome (Section 5.4).

### 5.1 Causal Effects of Annotation Bias

Natural Indirect Effect (NIE) measures the effect that intervention $A$ (annotation artifact) has on $Y$ through neurons $Z$. Figure 3 shows the average NIE of the top 5% neurons in each layer for all datasets, excluding the embedding and classification layers. The CMA results suggest that the top 5% of neurons in the last few layers have the highest effect, especially layers 10 and 11.

We also compare NIEs of the baseline method with reweighting, PoE, and debiasing masks methods. NIEs of these methods also have a similar pattern to the baseline's NIEs. As expected, when we consider the area under the curve of each method in Figure 3 as an accumulated effect, we can conclude that neurons in the baseline model exhibit the highest NIE values. For FEVER, all debiasing methods only yield small improvements on the challenge sets. This also means that all models still contain a similar level of bias. Nevertheless, this analysis suggests that model components still mediate annotation biases despite using the debiased fine-tuning methods.

Figure 4 shows NIEs as we greedily select more neurons using the top-k algorithm to intervene. We provide results on both individual layers and full model interventions. As the size of intervened neurons gets larger, the NIE will saturate at around 10,000 neurons. We also contrast the results between layers with the highest NIE (layer 11) and the second highest NIE (layer 10). The results show that intervening on the layer with the highest NIE can also have a much larger impact.

| Method | MNLI (acc) | | FEVER (acc) | | | QQP (MaF1) | |
|---|---|---|---|---|---|---|---|
| | dev-mm | HANS | test | Symm v1 | Symm v2 | test | PAWS |
| Baseline | **82.62** | 59.35 | **85.14** | **54.81** | **61.52** | **90.80** | 31.93 |
| + Masking | 80.87 | **64.05*** | 83.07 | 53.64 | 60.51 | 88.23 | **39.08*** |
| + Gradient Unlearning | 81.68 | 62.35* | 84.43 | 54.03 | 60.42 | 89.68 | 37.71* |
| Reweighting | **81.91** | 62.04 | **85.36** | **55.45** | **62.11** | **90.13** | 42.08 |
| + Masking | 79.95 | **65.87*** | 84.08 | 54.78 | 61.99 | 87.94 | **51.05*** |
| + Gradient Unlearning | 80.69 | 65.12* | 84.71 | 55.23 | 61.34 | 88.71 | 48.79* |
| PoE | **82.08** | 63.45 | **84.97** | 54.92 | 61.54 | **90.36** | 37.43 |
| + Masking | 80.71 | **68.12*** | 84.48 | **55.84** | **62.33** | 87.22 | **49.94*** |
| + Gradient Unlearning | 81.00 | 66.31* | 84.26 | 54.59 | 60.87 | 89.06 | 45.97* |
| Debiasing Masks | **83.71** | 66.79 | **84.72** | 55.01 | **61.74** | **91.22** | 43.04 |
| + Masking | 81.09 | 69.78* | 83.30 | **55.20** | 61.63 | 89.73 | 44.25 |
| + Gradient Unlearning | 82.23 | **70.05*** | 84.11 | 55.03 | 61.52 | 88.14 | **54.42*** |

Table 1: Results from the main experiment evaluated on both in-distribution and out-of-distribution (grey columns) test sets across three NLU tasks. We compare the efficacy of our bias mitigation approach against the conventional fine-tuning methods. The average scores from five runs with different random seeds are presented, and * indicates a statistically significant improvement achieved by our bias mitigation method compared to a fine-tuning method. We use the Almost Stochastic Dominance test (Dror et al., 2019) with the significant level of 0.05.

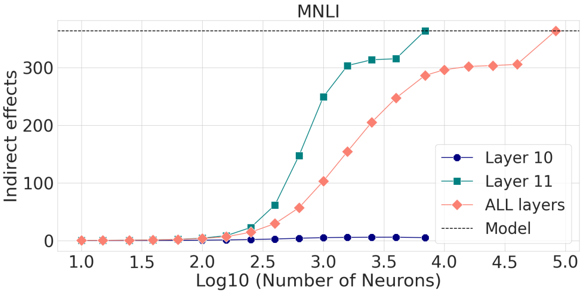


Figure 4: The natural indirect effects of the baseline model (BERT-base-uncased) as we select more neurons from all layers or individual layers to intervene using the top-k approach. Note that layer 11 is the final layer. The dash line denotes NIE when we intervene all neurons.

### 5.2 Results of Bias Mitigation based on CMA

To show the effectiveness of the CMA process in identifying bias-mediating neurons, we experiment with two post-hoc bias mitigation methods to assess improvement on challenge sets.

Table 1 shows that when we apply bias mitigation methods based on the neurons obtained from the CMA process, we can significantly improve the performances on the challenge sets compared to the baseline on MNLI and QQP. Furthermore, both bias mitigation methods can can be used on models trained using competitive methods for more improvement. This suggests we can modify their behavior by identifying the biased neurons to reduce the sensitivity against known spurious correlation features.

The robustness improvements for NLI and paraphrase identification tasks are more substantial than the fact verification task. When we apply the causal-grounded masking method, small improvements can only be observed on both FEVER Symmetric challenge sets for the PoE model and on the Fever Symmetric Version 1 for the debiasing masks method. Nevertheless, the parity of accuracy between classes is also smaller when a bias mitigation method is applied (See Appendix A.7). We conjecture that this is because the samples for the counterfactual representation generation are easier to identify in MNLI and QQP datasets.

### 5.3 Impact of CMA and Advantaged Samples on Bias Mitigation

In this section, we explore the impact of incorporating knowledge from CMA for bias mitigation methods. As shown in Table 2, we can improve the performance by applying knowledge from CMA to select neurons for both masking and gradient unlearning processes. If random neurons are used (w/o CMA), the HANS dataset results drop from 64.05 to 59.27 for causal-grounded masking, and from 62.35 to 59.86 for gradient unlearning. This suggests that neurons selected by the CMA process are components that mediate more bias, which can be rectified through a bias mitigation process.

For the gradient unlearning process, the result shows that using samples other than advantaged samples can be detrimental to the performance.

The performance on the HANS dataset drops from 62.35 to 59.34 when advantaged samples are selected randomly. This highlights the importance of identifying biased samples.

| Method | MNLI Results | |
|---|---|---|
| | dev-mm | HANS |
| Masking | 80.87 | 64.05 |
| - wo/ CMA | 82.65 | 59.27 |
| Unlearning | 81.68 | 62.35 |
| - wo/ CMA | 82.40 | 59.86 |
| - wo/ adv. samples | 82.46 | 59.34 |

Table 2: The ablation results of using CMA to select neurons to mitigate bias. We show the results of applying the two bias mitigation methods on the baseline model. Then, we show the results of applying the two methods without the CMA knowledge or without the advantaged samples (for gradient unlearning).

### 5.4 Model-level Bias Effect Analysis Through CMA

In contrast to the causal graph in Figure 1, we follow Udomcharoenchaikit et al. (2022) and apply CMA to measure the effect of annotation artifact that flows through the whole model by viewing the input and model as the mediator instead of the set of neurons. We then conduct the bias analysis to find the average total indirect effect (ATIE) of the annotation bias in the HANS dataset. Table 3 shows the ATIE and accuracy for each class and for each heuristic. Both causal-based debiasing methods reduces the ATIE for all cases.

| | Class | Lexical Overlap | | Subsequence | | Constituent | |
|---|---|---|---|---|---|---|---|
| | | ATIE | ACC | ATIE | ACC | ATIE | ACC |
| Baseline | E | 0.2755 | 95.01 | 0.3080 | 99.67 | 0.3146 | 99.65 |
| | N | 0.0639 | 42.64 | 0.2525 | 7.60 | 0.2458 | 11.50 |
| | Overall | 0.1697 | 68.83 | 0.2802 | 53.63 | 0.2802 | 55.58 |
| + Masking | E | 0.1701 | 85.53 | 0.2264 | 94.98 | 0.2339 | 95.72 |
| | N | -0.0157 | 59.56 | 0.1515 | 18.76 | 0.1538 | 29.72 |
| | Overall | 0.0772 | 72.54 | 0.1890 | 56.87 | 0.1938 | 62.72 |
| + Gradient Unlearning | E | 0.2586 | 92.56 | 0.3015 | 99.21 | 0.3068 | 98.98 |
| | N | 0.0214 | 51.61 | 0.2290 | 11.67 | 0.2146 | 20.05 |
| | Overall | 0.1400 | 72.09 | 0.2652 | 55.44 | 0.2607 | 59.52 |

Table 3: ATIE and accuracy of each syntactic heuristic in the HANS dataset. E denotes the entailment class, and N denotes the non-entailment class.

## 6 Conclusion

Our investigation presents a novel approach to NLU bias mitigation research through Causal Mediation Analysis (CMA). In contrast to existing works that treat models as black boxes and primarily concentrate on loss engineering, CMA performs component-wise analysis to pinpoint the neurons susceptible to bias.

By employing CMA, we have found that annotation bias effects are primarily concentrated within neurons located in the final layers of the model. Through this newfound knowledge, we develop a strategy to direct our mitigation efforts to specific neurons and apply the two bias mitigation methods accordingly. Experimental results show that we can reduce the impact of annotation artifacts through two bias mitigation methods. Results from the ablation study confirm that CMA is effective in identifying neurons that need to be rectified. By addressing the sensitivity of model modules to annotation bias, we pave the way for enhancing the robustness of NLU models against spurious correlations in an interpretable manner.

## Limitations

In this study, we utilize CMA to explore the mediation mechanisms for annotation artifacts within NLU models. One of the steps that has a large impact on the CMA framework is the counterfactual generation step. However, this step is still challenging for NLU tasks. Despite its ease of use and generality, the proposed aggregated counterfactual generation does not consider the uniqueness of each sample.

The need to identify known bias features to select bias samples is also a limitation of this work. It does not allow us to select samples with unknown biases. Training a deep learning model with limited samples to create a bias model is also a promising direction (Utama et al., 2020b).

One of the potential risks of this work is that by reversing the gradient direction of the gradient unlearning method, one can also amplify the biases. This can be concerning when dealing with societal or political biases.

## References

Abhijith Chintam, Rahel Beloch, Willem Zuidema, Michael Hanna, and Oskar van der Wal. 2023. Identifying and adapting transformer-components responsible for gender bias in an English language model. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 379–394, Singapore. Association for Computational Linguistics.

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

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

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.

Li Du, Xiao Ding, Zhouhao Sun, Ting Liu, Bing Qin, and Jingshuo Liu. 2023. Towards stable natural language understanding via information entropy guided debiasing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2868–2882, Toronto, Canada. 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.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

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.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. 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.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

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

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.

Johannes Mario Meissner, Saku Sugawara, and Akiko Aizawa. 2022. Debiasing masks: A new framework for shortcut mitigation in NLU. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7607–7613, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Aaron Mueller, Yu Xia, and Tal Linzen. 2022. Causal analysis of syntactic agreement neurons in multilingual language models. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 95–109, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA.

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.

Ivan Sanchez, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of NLI models: Uncovering the influence of three factors on robustness. 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 1975–1985, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by finetuning. *CoRR*, abs/2005.07683.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. 2023. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7035–7052, Singapore. 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.

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. 2022. Debiasing nlu models via causal intervention and counterfactual reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11376–11384.

Can Udomcharoenchaikit, Wuttikorn Ponwitayarat, Patomporn Payoungkhamdee, Kanruethai Masuk, Weerayut Buaphet, Ekapol Chuangsuwanich, and Sarana Nutanong. 2022. Mitigating spurious correlation in natural language understanding with counterfactual inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11308–11321, Abu Dhabi, United Arab Emirates. 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.

Fei Wang, James Y. Huang, Tianyi Yan, Wenxuan Zhou, and Muhao Chen. 2023a. Robust natural language understanding with residual attention debiasing. In

*Findings of the Association for Computational Linguistics: ACL 2023*, pages 504–519, Toronto, Canada. Association for Computational Linguistics.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023b. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, 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.

Charles Yu, Sullam Jeoung, Anish Kasi, Pengfei Yu, and Heng Ji. 2023. Unlearning bias in language models by partitioning gradients. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6032–6048, Toronto, Canada. 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.

Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. 2020. Masking as an efficient alternative to finetuning for pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2226–2241, Online. Association for Computational Linguistics.

# A Appendix

## A.1 Datasets

- **Natural Language Inference:** We use the MNLI 1.0 dataset (Williams et al., 2018) to train, validate (dev-matched), and test (dev-mismatched) our method. To measure the robustness, we use HANS (McCoy et al., 2019) as a challenge set. HANS is used to study lexical overlap bias, where models are likely to predict samples with high word overlap as entailments.
- **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. We use the same split as Udomcharoenchaikit et al. (2022). FEVER Symmetric is used to examine claim-only bias, where models predict solely based on claims without looking at evidence.
- **Paraphrase Identification:** We use QQP[3] as a benchmark. Since there is no standard train/test split, we follow Udomcharoenchaikit et al. (2022) and split 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. PAWS is used to examine lexical overlap bias.

## A.2 Data Statistics

The data statistics for the datasets employed in our experiments are presented in Table 4. There is an imbalance in the ratio of non-paraphrase to paraphrase pairs within the QQP in-distribution test set and the PAWS challenge set. Consequently, we employ Macro F1 to communicate the scores for the paraphrase identification task. Moreover, we follow Udomcharoenchaikit et al. (2022) and use the same data splits for all tasks. All the datasets are in English.

## A.3 Computing Resources

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. It requires us approximately up to 3 hours to train one model. CMA analysis can take up to 12 hours for each model. The gradient unlearning process is approximately 1-hour long. For each fine-tuning method, we train using five different random seeds. Hence, we approximate that replicating our results could take at least 720 GPU hours.

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

| 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.4 Model Implementation details

**Main Model.** We mainly study the bert-base-uncased model (Devlin et al., 2019). The model performs effectively in the three NLU tasks; however, it still relies on surface-level cues (Gururangan et al., 2018; McCoy et al., 2019).

We take the contextualized embedding found at the [CLS] position in the last layer of the BERT model and input it into a feed-forward layer with Softmax activation, following a similar approach as seen in previous studie (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. We use the learning rate of `5e-5` to train on the MNLI training set. For FEVER and QQP training sets, we follow Utama et al. (2020a,b) and use the learning rate of `2e-5`.

We implement the slanted triangular learning rate schedule, allocating 0.06 fraction of the steps for learning rate increase. The batch size is set at 32, and we incorporate automatic mixed-precision training in our training process. We mainly use PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2020).

**Bias Model.** For NLI and paraphrase identification tasks, we follow Clark et al. (2019) and use a simple logistic regression model with the following bias features: (1) whether the hypothesis is a sub-sequence of the premise, (2) whether all words in the hypothesis are in the premise, (3) the lexical overlap fraction, (4) the average minimum cosine distance between fastText word vectors (Joulin et al., 2017) of each premise word and each hypothesis word, and (5) the maximum of those cosine distances. For each sample in the paraphrase identification task, we replace the premise and hypothesis with the first and second sentences of the sample, respectively. For the fact verification task, we train the baseline model using only claim sentences. For logistic regression models' implementation, we use scikit-learn (Pedregosa et al., 2011).

### A.5 Recommendation on number of neurons to manipulate for bias mitigation

Since debiasing methods aim to reduce the effect of shortcuts, it is common for debiasing methods to have a trade-off between in-distribution (when shortcuts are advantageous) and out-of-distribution (when shortcuts are disadvantageous) settings (Utama et al., 2020a). Furthermore, it can also be hard to adjust hyperparameters that directly impact the debiasing strength. Because the available validation set often has a similar distribution to the training set. We propose that instead of optimizing the performance of a validation set, we should define an acceptable drop for a validation performance.

For this study, we set an acceptable validation performance to be over 80. This is approximately less than two points short of the validation performance of the baseline model (81.81 on dev-matched). We select the number of top neurons that can pass these criteria for both masking and gradient unlearning methods. In addition, for the gradient unlearning method, we also use these criteria to select an amplification scale for the learning rate.

As shown in Table 5, we use the top 700 neurons and the amplification scale of 5 for all tasks. In order to test its sensitivity and generalizability, we use the same number of top neurons for all tasks. However, for further customization for each task, one can use the same criteria to adjust the desirable trade-off.

The gradient unlearning method uses different advantages sample sizes depending on the outcomes of the main and the bias models, which are sensitive to the learning rate we choose. We interpolate the learning rate $\alpha_o$ based on the hyperparameters reported by Yu et al. (2023) and our advantaged sample size.

$$\alpha_o = \frac{N_i \alpha_i c}{N_o} \tag{8}$$

$N_i$ and $\alpha_i$ denote the number of training steps and the learning rate used in Yu et al. (2023). $N_o$

| Methods | LR scale ($c$) | # of neurons | MNLI (acc) | | |
|---|---|---|---|---|---|
| | | | dev-matched | dev-mm | HANS |
| Masking | - | 100 | 81.40 | 82.04 | 61.39 |
| | - | 500 | 80.64 | 81.31 | 63.28 |
| | - | 600 | 80.47 | 81.17 | 63.55 |
| | - | **700** | **80.21** | 80.87 | 64.05 |
| | - | 800 | 79.74 | 80.54 | 64.86 |
| | - | 900 | 79.51 | 80.34 | 64.97 |
| | - | 1,000 | 79.16 | 80.06 | 65.42 |
| | - | 1,500 | 75.62 | 76.56 | 65.54 |
| | - | 2,500 | 68.34 | 69.39 | 59.55 |
| | - | 5,000 | 53.69 | 54.63 | 53.90 |
| | - | 10,000 | 42.74 | 43.08 | 50.72 |
| Gradient Unlearning | 1/5/10 | 100 | 81.78/81.62/81.4 | 82.59/82.49/82.30 | 59.50/60.05/60.76 |
| | 1/5/10 | 500 | 81.66/81.09/ 80.06 | 82.53/81.97/80.99 | 59.84/61.55/63.98 |
| | 1/5/10 | 600 | 81.65/80.88/79.74 | 82.52/81.80/80.64 | 59.91/62.00/64.49 |
| | 1/**5**/10 | **700** | 81.64/**80.69**/79.32 | 82.48/81.68/80.21 | 59.98/62.35/65.11 |
| | 1/5/10 | 800 | 81.63/81.63/79.22 | 82.48/81.48/79.96 | 60.03/62.39/ 65.14 |
| | 1/5/10 | 900 | 81.61/80.43/78.51 | 82.45/ 81.35/79.35 | 60.08/62.86/65.75 |
| | 1/5/10 | 1,000 | 81.58/80.27/ 78.07 | 82.42/81.19/78.89 | 60.14/ 63.11/66.08 |
| | 1/5/10 | 1,500 | 81.43/79.31/ 74.44 | 82.29/80.22/75.32 | 60.45/ 64.44/66.61 |
| | 1/5/10 | 2,500 | 81.1/ 76.93/ 62.12 | 82.04/ 77.46/ 62.79 | 60.99/66.03/59.28 |
| | 1/5/10 | 5,000 | 80.45/65.20/42.55 | 81.31/65.45/ 42.81 | 62.08/ 60.70/ 53.12 |
| | 1/5/10 | 10,000 | 79.2/ 50.45/32.59 | 80/50.33/32.75 | 63.38/57.47/50.00 |

Table 5: Results of the bias mitigation methods on different hyperparameter settings. The bold text indicates the setting that we chose for our experiment which met the criteria (above 80 on dev-matched). The grey columns represent the scores on the validation set. From layer 0 to layer 11, there are 82,944 neurons in total.

and $\alpha_o$ denote the number of training steps and the learning rate used in our study. Since we are not training the whole model, we compensate for this by multiplying the learning rate with $c$. Additionally, we hyper-tune only the learning rate scaler $c$., ensuring an 80% accuracy on the MNLI validation set of a baseline model and then applying this to all models.

We also show the results of various hyperparameter configurations on the MNLI test set (dev-mismatched) and the HANS challenge set in Table 5. The number of selected neurons can be as low as 100, and we still yield an improvement on the challenge set. The trade-off is also smaller when the number of neurons is tiny, but the robustness improvement is also small. When the number of neurons is large (from 2,500 onwards), the masking method fails to gain any improvement and also fails to maintain the in-distribution scores. The gradient unlearning method also fails when the number of neurons and the learning rate amplification are extreme.

### A.6 Counterfactual Analysis

In this section, we analyze the indirect effects for each models with two different aggregated counterfactuals:

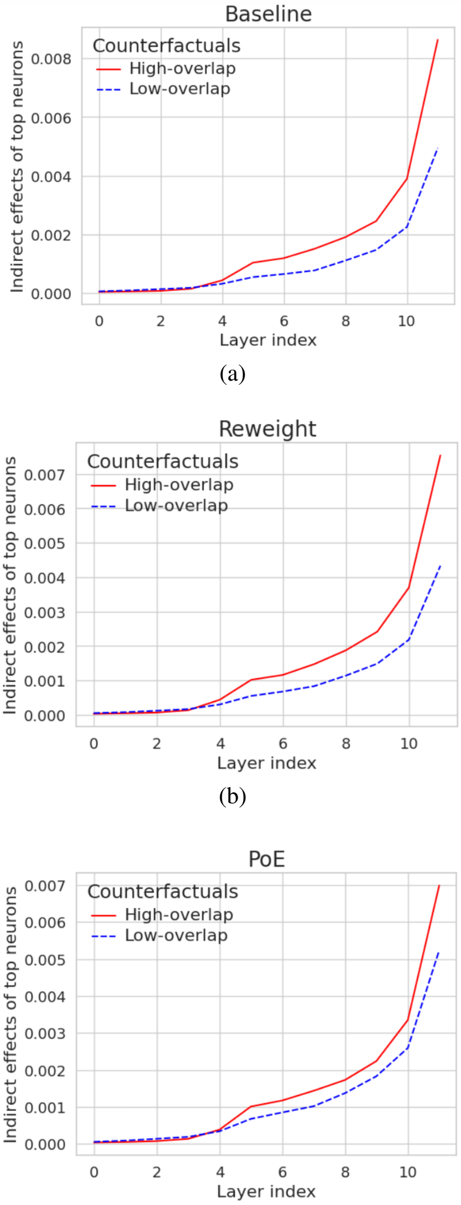
1. A high-overlap counterfactual is the average of activations from biased inputs with high-lexical overlap (equal or above the 95th-percentile).

2. A low-overlap counterfactual is the average of activations from biased inputs with low-lexical overlap (equal or below the 5th-percentile).

Figure 5 shows that the high-overlap counterfactuals of the three models have greater indirect effects on top neurons, compared to the low-overlap counterfactuals. It also shows that neurons from the top layers are more sensitive to the change in counterfactuals.

### A.7 Parity of Accuracy

In this section, we examine the parity of accuracy between classes in NLU tasks as shown in Table 6. By focusing on the distribution of accuracy across various classes, we aim to highlight any disparities that may exist and investigate how our bias mitigation strategies affect them. For NLI, we show a parity between entailment and non-entailment



(a)

(b)

(c)

Figure 5: The natural indirect effects of top 5% neurons on MNLI dataset of (5a) the baseline model, (5b) the reweighting model, and (5c) the PoE model.

| Method | MNLI (diff acc) | | FEVER (diff acc) | | | QQP (diff F1) | |
|---|---|---|---|---|---|---|---|
| | dev-mm | HANS | test | Symm v1 | Symm v2 | test | PAWS |
| Baseline | 1.33 | 77.53 | 15.10 | 30.26 | 20.56 | 4.66 | 24.37 |
| + Masking | 9.01 | 56.07 | 10.91 | 26.70 | 15.96 | 7.19 | 19.67 |
| + Gradient Unlearning | 6.89 | 69.14 | 11.77 | 24.42 | 14.33 | 6.04 | 18.15 |
| Reweighting | 2.66 | 70.29 | 14.82 | 29.68 | 20.62 | 5.31 | 13.81 |
| + Masking | 10.64 | 54.94 | 12.46 | 27.52 | 18.93 | 7.48 | 22.50 |
| + Gradient Unlearning | 9.76 | 60.52 | 10.71 | 22.32 | 13.03 | 6.87 | 11.37 |
| PoE | 2.25 | 67.03 | 14.94 | 30.58 | 21.52 | 5.11 | 14.51 |
| + Masking | 7.07 | 47.43 | 13.51 | 29.75 | 20.96 | 8.25 | 20.33 |
| + Gradient Unlearning | 7.11 | 57.40 | 11.02 | 23.79 | 13.54 | 6.63 | 7.00 |
| Debiasing Masks | 2.15 | 61.36 | 14.18 | 30.85 | 19.89 | 4.65 | 4.68 |
| + Masking | 12.26 | 42.80 | 8.86 | 24.51 | 13.48 | 5.69 | 17.69 |
| + Gradient Unlearning | 10.10 | 49.86 | 9.68 | 21.73 | 11.12 | 7.75 | 27.08 |

Table 6: The parity of accuracy between classes in NLU tasks. Lower parity may suggest smaller inclination towards a specific class.

classes in the HANS challenge set. Note that for dev-mm which has three classes, we show an average parity between entailment vs contradiction and entailment vs neutral. For fact verification, we show parity between support and refute classes. For paraphrase identification, we show parity between paraphrase and non-paraphrase classes

For challenge sets, the bias mitigation methods can reduce the parity of accuracy between classes for almost all cases. Except for the results of the masking method in the PAWS challenge set, we observe the rise in the parity. However, this is due to the fact that the masking method sharply increases the F1 score of the non-paraphrase class, but it does not improve the F1 score for the paraphrase class. We also observe a similar phenomenon when we apply gradient unlearning on top of the Debiasing Masks method.

For the in-distribution test sets, the bias mitigation methods increase the parity for NLI and paraphrase identification tasks. However, this gap increases in a much smaller scale than what we observe in the challenge sets.