Unlearning the Spurious Correlations for Improving Generalization of Language Models

Xinyi Sun[a], Hongye Tan[†a,b], Dongzhi Han[a], Zhichao Yan[a], Xiaoli Li[c], Ru Li[a,b], Hu Zhang[a]

[a]*School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China*
[b]*Institute of Intelligent Information Processing, Shanxi University, Taiyuan 030006, Shanxi, China*
[c]*Institute for Infocomm Research, A*STAR, Singapore*

Abstract

Data-driven language models often perform poorly on out-of-distribution data due to the internalization of spurious correlations. Existing mitigation strategies focus on data preprocessing or model regularization, which do not sufficiently correct the spurious correlation and require high costs. We propose a novel plug-and-play method, MacU, a machine unlearning strategy for mitigating the spurious correlation. Firstly, we accurately locate neurons encoding correlation information based on the gradient responses of the proximity. Then, we analyze the model's neuron storage mechanism of causal or spurious correlation information, and identify overlap between these two types of neurons. Based on the above findings, we develop a neuron editing method based on Particle Swarm Optimization, ensuring that the model forgets spurious correlations while retaining the best memory of causal correlations. Experiments on 5 language models and 11 datasets show that MacU significantly improves generalization, surpassing the fine-tuning baseline by 3.13% and 2.05% in sentiment analysis and natural language inference tasks, respectively. To the best of our knowledge, this is the first work to design a machine unlearning technique to mitigate spurious correlation, which has achieved superior correction capability without requiring retraining the model.

Keywords: Spurious correlation, Generalization, Machine unlearning, Neuron localization, Neuron editing

1. Introduction

Data-driven language models (LM), especially large language models (LLM), have achieved outstanding performance in various natural language processing (NLP) tasks. However, recent studies show that these models lack out-of-distribution (OOD) generalization ability [9, 26]. A major contributing factor is that these LMs often *learn spurious correlations between the training data and corresponding labels* [16, 20, 25]. These spurious correlations can lead to unreliable predictions made by the model in practical applications, which may result in serious consequences [29].

At present, there are three main approaches to alleviate spurious correlations: pre-processing, in-processing, and post-processing [16]. Pre-processing approaches aim to retrain the models by

[†]Corresponding author: Hongye Tan (Email: tanhongye@sxu.edu.cn; ORCID:0000-0002-5858-899X)

removing or reducing the proportion of spurious correlation data in the training set. In-processing approaches adjust the training strategy and mitigate the spurious correlations during training. Post-processing approaches correct the model's outputs by removing the effects of spurious correlations at the prediction stage. Although these approaches have achieved certain results, they still face two major bottlenecks: (1) Insufficient correction caused by training mechanism defects. These mitigation approaches, based on data augmentation or training strategy optimization, fine-tune the model parameters by retraining the model to achieve spurious correlation correction. However, due to the inherent confirmation bias [25] in the model and deficiencies in the training mechanism (such as optimization mechanisms) [21], this correction process is often hindered. (2) High costs. Many of these approaches require the construction of spurious correlation-free datasets (the datasets without spurious correlations) or retraining models, which incur significant computational and resource costs.

Methods	Parameter fine-tuning	Parameter editing	No adjustment
Pre-processing	×	✓	×
In-processing	×	✓	×
Post-processing	×	×	✓
Machine unlearning based on locate-then-edit	✓	✓	×

(a) Parameter adjustment method of different mitigation approaches.

Methods	Additional data	Retraining the model
Pre-processing	✓	✓
In-processing	✓	✓
Post-processing	×	×
Machine unlearning based on locate-then-edit	×	×

(b) Dataset and training requirements of different approaches.

Table 1: Comparative analysis of various mitigation approaches. In the tables, ✓ indicates that the approaches need to meet the corresponding conditions, while × indicates that the approaches do not meet the corresponding conditions.

Motivated by these limitations, our research goal is to design a spurious correlations mitigation approach with strong correction and low-cost characteristics, which can better improve the generalization of language models.

Machine unlearning [17] [36] is a widely used technology in fields such as privacy protection [35], fairness [4], etc. It deletes or modifies the model's memory of specific data or information through particular methods. Especially, machine unlearning based on the locate-then-edit method can accurately remove target information without retraining the model. This method achieves forgetting by locating and editing neurons relevant to the content to be forgotten. Inspired by [35] and [4], which suggest that privacy information or specific knowledge is stored in specific privacy neurons or knowledge neurons, we hypothesize that spurious correlation information may also exist in specific neurons. This suggests that we could alter the model's memorization of spurious correlation information by detecting and editing these neurons, which we term *spurious correlation neurons*.

This study proposes MacU, an innovative methodology based on machine unlearning technology for spurious correlation mitigation. It achieves efficient calibration by locating and editing these "spurious correlation neurons" without the need to retrain the model, combining the advantages of low cost and strong calibration, as follows the Tables 1a and 1b. The core technological innovation of MacU includes:

(1) **Neuron localization guided by the gradient of proximity.** Typically, neuron localiza-

tion methods rely on the gradient of the model's performance on the specific dataset to neuron changes. However, for the spurious correlation mitigation task, constructing such datasets (spurious correlation-free datasets) is challenging due to the limitations of spurious correlation cognition. *Regarding this problem, we innovatively propose a specific-dataset-free method that locates spurious correlation neurons by analyzing the gradient of the proximity metric to neuron changes*. The proximity metric is defined as the alignment degree between the correlation tokens identified by the model and the golden correlation tokens. Among these, correlation tokens are markers automatically identified by the model during prediction that exhibit strong associations with the output. In contrast, golden correlation tokens represent pre-defined standard tokens that have either causal or spurious correlations with the ground-truth labels. Significant fluctuations in the proximity metric resulting from adjustments to specific neuronal parameters suggest that the corresponding neuron encodes certain causal or spurious correlations.

(2) **Spurious correlation unlearning with causal preservation**. Our localization results reveal that neurons storing spurious correlations often overlap with those storing causal correlations. Adjusting these neurons' parameters can inadvertently affect *both* causal and spurious correlation memories. *To mitigate this problem, we propose a new neuron parameter editing method based on the Particle Swarm Optimization (PSO) algorithm [31]. This new method aims to optimize the model parameters, ensuring the model forgets spurious correlations while retaining causal correlations optimally*.

In the experiment section, we perform a wide range of tests on two different tasks (Sentiment Analysis (SA) and Natural Language Inference (NLI)) using multiple datasets, demonstrating that MacU substantially improves the model's performance and generalization. Specifically, we first validate the dual advantages of the MacU method on the BERT model: 1) The MacU demonstrates good generalization ability, surpassing the Fine-tuning baseline by 3.13% and 2.05% on SA and NLI, respectively. 2) The MacU exhibits better spurious correlation correction characteristics than all other spurious correlation mitigation strategies. Then, to verify the universality of the MacU, we extend the MacU to LLMs with different parameter scales (such as GPT-2, LLaMA-2-7B , Qwen-2-1.5B, and GLM-4-9B), and the experimental results show that the MacU maintains stable effectiveness. Besides, we verify the core hypothesis of "spurious correlation and causal correlation information are encoded in specific neurons" through analysis of neuron activation patterns. Furthermore, we analyze the results of neuron localization, and find that there is a significant overlap between the neurons encoding spurious correlations and causal correlations. This discovery provides a theoretical basis for neuron parameter adjustments to achieve spurious correlation forgetting.

The key contributions of our work are outlined below:

(1) We propose a new plug-and-play method, MacU, which is the first work to utilize machine unlearning techniques to alleviate spurious correlations in models and improve their generalization without the need to build external datasets and retrain the model, offering both low costs and substantial correction capability.

(2) We implement the MacU method through an efficient neuron localization module guided by the proximity, and a neuron editing module based on the PSO algorithm. This method allows that the model forgets spurious correlations while retaining the best memory of causal correlations as much as possible.

(3) We discover the overlap phenomenon between neurons encoding spurious correlations and causal correlations, which is the theoretical basis of the neurons editing method proposed in this paper and a supplement to the current research on how LMs encode the knowledge of correlation type, contributing to a understanding of knowledge representation in neural networks.

2. Related works

Spurious correlation mitigation methods. Many studies have been conducted to alleviate spurious correlations in existing research [16]. **Some researchers** attempt to reduce spurious correlations at the data level through counterfactual or adversarial data augmentation. For example, [33] proposes a counterfactual data augmentation method to reduce the model's reliance on potentially spurious correlations present in the original data. [25] proposes a smart data augmentation method that generates adversarial examples without spurious correlations using LLMs and employs a two-phase learning strategy to train the model for spurious correlation mitigation. *However, these approaches are costly due to the need to construct external datasets.* **Other researchers** reduce spurious correlations by intervening in the model's training process. For example, [24] uses an ensemble of multiple adversaries to avoid the hypothesis-only bias, significantly reducing the spurious correlation or bias stored within a model's representations. [6] propose a causal contrastive learning approach, which first generates the counterfactual pairs and the factual pairs by masking the identified causal and non-causal terms separately, and then employs contrastive learning on these constructed samples to reduce the model's sensitivity to spurious correlations. [5] proposes a regularization method, NFL, which constrains changes in language model parameters or outputs to prevent capturing spurious correlations from misalignment. *These approaches typically require detailed algorithm design and model retraining.* **Still other researchers** propose a post-processing method to alleviate spurious correlations by adjusting model outputs, which overcomes the aforementioned shortcomings. For example, [14] proposes an end-to-end unbiased method, PoE, that adjusts the cross-entropy loss based on predictions from a hypothesis-only biases model, reducing spurious correlations learned during training.

Although these strategies offer some benefits, they are insufficient in correcting spurious correlations and often result in limited generalization. This paper introduces a spurious correlation mitigation method based on machine unlearning, which effectively removes spurious correlations without retraining the model or using external datasets.

Machine unlearning. Machine unlearning allows models to forget specific data, information, or learned knowledge [17] [36]. It includes exact unlearning and approximate unlearning [4]. The goal of exact unlearning is to ensure that the forgotten data has no impact on the final model, as if it had never been used for training the model [2]. This method typically requires high computational costs as it involves retraining the model. Approximate unlearning does not aim to completely eliminate the impact of forgotten data, but rather reduces this impact as much as possible by adjusting model parameters, which is usually more efficient [13].

Model parameter adjustment based on the locate-then-edit is an important approximate unlearning method for achieving machine unlearning, which achieves forgetting by locating and editing neurons that are relevant to the content to be forgotten. This method can accurately remove target information without retraining the model, and has been widely applied in privacy protection [35], fairness [38], and more. For example, [35] proposes the DEPN framework, which identifies neurons linked to private information based on the performance gradient responses, and edits them by setting their activations to zero to achieve privacy forgetting. [38] proposes an interpretable neuron editing method that combines logit-based and causal-based strategies to target biased neurons selectively. Inspired by this, we hypothesize that spurious correlation information may be encoded in specific neurons, similar to factual or privacy neurons. And we propose a spurious correlations mitigation method based on machine unlearning.

In contrast to [35], (1) we use a neuron localization method driven by the *proximity gradient*

response, instead of performance gradient responses on the specific dataset, to achieve more efficient localization. (2) Instead of simply setting the recognized neurons to zero, we introduce a neuron parameter editing method based on PSO, which *can edit neuron parameters to an optimal state, in which the model forgets spurious correlations while preserving causal correlations*.

3. Methods

We introduce our proposed MacU method, as shown in Figure 1. The method consists of two modules: (1) **The neuron localization module**, which identifies neurons involved in encoding causal (or spurious) correlation information, and (2) **The neuron parameter editing module**, which forgets spurious correlation by adjusting the parameters of the identified neurons.

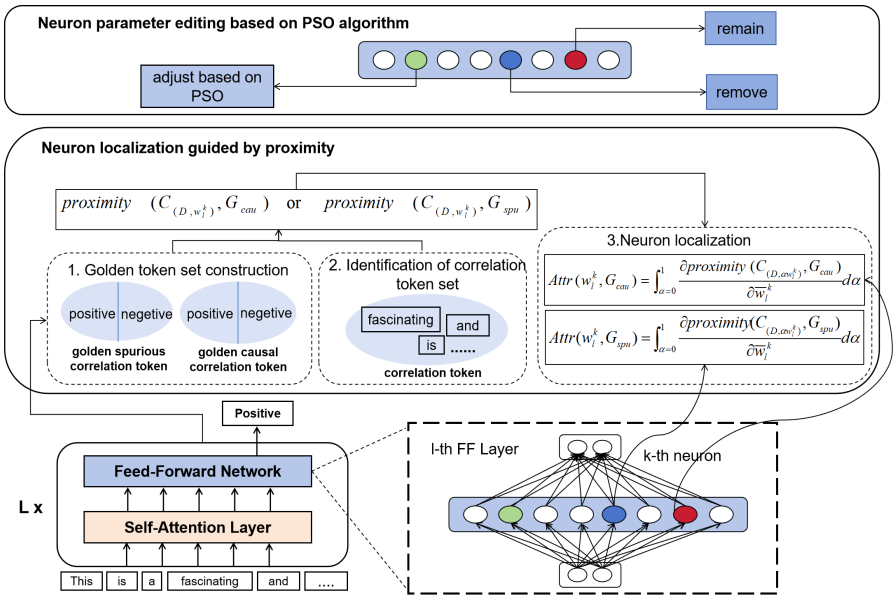


Figure 1: Framework diagram of the MacU. Red circles represent neurons encoding *causal correlations*, blue circles represent neurons encoding *spurious correlations*, and green circles represent neurons encoding *both causal and spurious correlation* information simultaneously.

3.1. Neuron localization guided by the gradient responses of proximity

We propose a novel neuron localization strategy guided by the gradient responses of proximity between the correlation tokens identified by the model and golden spurious correlation tokens. Specifically, for spurious correlation neuron localization, this strategy first identifies tokens that play a critical role in model prediction, which we refer to as "correlation tokens", reflecting the behavioral patterns of the model during decision-making. Then we compute the proximity between these correlation tokens and golden spurious correlation tokens that conform to human prior knowledge, and monitor how this proximity changes with the variation of neurons, i.e., the gradient responses of proximity. The higher the gradient response is, the higher the degree of internalization of these spurious correlations by neurons, indicating that the neurons encode the

corresponding spurious correlations. *The localization method of causal correlation neurons is similar to spurious correlation neurons.*

The main steps of this module include constructing the golden correlation token set, identifying the correlation token set, and performing neuron localization.

3.1.1. Golden correlation token set construction

The golden correlation token set is divided into two subsets: the *golden causal correlation token set* and the *golden spurious correlation token set*, used to localize causal or spurious correlation neurons, respectively.

Golden causal correlation token set. We construct the *golden causal correlation token set*, G_{cau}, by extracting tokens with *strong causal relationships to specific labels*, across multiple training sets for the same task. Specifically, we leverage the LLM (ChatGPT) to aid in constructing the golden causal correlation token sets.

1) Automatic discovery: LLM is employed to analyze the given datasets and identify the golden *causal* tokens which are causally correlated to the specific label. Specifically, we input the following prompt to the ChatGPT: "*This is an XXX task with the labels of XXX. Extract tokens that are causally correlated to the different labels from the text in the data*".

2) Manual verification: We manually review the identified tokens and remove any erroneous tokens that do not match human understanding of causality. This process is carried out by two master's students and one doctoral student, using double-blind annotation and third-party verification methods. Firstly, two individuals independently annotate the incorrect tokens. If the annotations are consistent, these tokens are deleted. If the annotations are inconsistent, a third party shall conduct an inspection [1]. Finally, the golden causal correlation token set G_{cau} is obtained.

$$G_{cau} = (G_{cau}^{label_1}, G_{cau}^{label_2}, ..., G_{cau}^{label_j}) \tag{1}$$

where $label_j$ is the j^{th} label in this task, and $G_{cau}^{label_j}$ is the token set causally related to the $label_j$.

Golden spurious correlation token set. According to experience, tokens that are not classified as having a causal correlation with the label are spurious correlation tokens. However, due to the limitations of human cognition of causal correlation, treating all non-causal tokens as spurious correlation tokens and deleting them may damage the model's performance. In this paper, the *golden spurious correlation token set* is defined to include both *misclassified* and *irrelevant* tokens.

1) Misclassified tokens: For a given label, misclassified tokens are those that have causal relationships with *other labels*. For example, the token "*fail*" has a stronger causal correlation with the label "*negative*", so it is considered a misclassified token when associated with the label "*positive*". The misclassified token set, G_{spu_M}, can be derived from the causal token set G_{cau} as follows:

$$G_{spu_M}^{label_j} = G_{cau}^{label_{\_j}} \tag{2}$$

[1] It is worth pointing out that due to the involvement of LLMs, the labor cost required to construct the golden causal correlation token set is not high. The annotators are only responsible for verification work, which is not difficult. In the end, we built approximately 8000 golden correlation tokens on the SA task, and nearly 7000 golden correlation tokens on the NLI task.

Golden correlation token set				
Golden causal correlation token set		Golden spurious correlation token set		
-		Misclassified tokens		Irrelevant tokens
Positive	Negative	Positive	Negative	-
Yes, Happy, Charm, Award, Beautiful...	Abnormal, Abolish, Fail, Absurdly, Paradoxical...	Abnormal, Abolish, Fail, Absurdly, Paradoxical...	Yes, Happy, Charm, Award, Beautiful...	Where, A, Is, When, He, She, Boy, Men, Woman, Girl...

Table 2: The examples of golden correlation token set in SA.

Golden correlation token set						
Golden causal correlation token set			Golden spurious correlation token set			
-			Misclassified tokens			Irrelevant tokens
En-	Con-	Neu-	En-	Con-	Neu-	-
Necessity, Consistency, Indispensable, Sufficient, Resultant, ...	No, Not, Inconsistency, Dispute, Confliction, ...	Balance, Objectivity, Impartiality, Moderation, Equilibrium, Neutrality, Nonalignment ...	Balance, Objectivity, Impartiality, Moderation, Equilibrium, Neutrality,..., No, Not, Inconsistency, Dispute, Confliction, ...	Necessity, Consistency, Indispensable, Sufficient, Resultant, ... , Balance, Objectivity, Impartiality, Moderation, Equilibrium, Neutrality, Nonalignment ...	Necessity, Consistency, Indispensable, Sufficient, Resultant, ... , No, Not, Inconsistency, Dispute, Confliction, ...	Where, When, What, How, Whether, A, Is, When, He, She, Boy, Men, Woman, Girl...

Table 3: The examples of golden correlation token set in NLI. En- is the label of Entailment, Con- is the label of Contradiction, and Neu- is the label of Neutral.

where $\text{label}_{\_j}$ represents all labels except the j^{th} label. $G_{spu_M}^{label_j}$ is the set of misclassified tokens associated with the j^{th} label, while $G_{cau}^{label_{\_j}}$ refers to the *causal* token set related to the $\text{label}_{\_j}$, i.e., the tokens that are causally related to labels *other than* j^{th} label.

2) Irrelevant tokens. Tokens that are maximally or entirely unrelated to the labels are considered irrelevant, such as common words like *where*, *is*, *a*, etc. These tokens are collected to form the irrelevant token set, G_{spu_I}. The entire golden spurious correlation token set, G_{spu}, is represented as:

$$\begin{aligned} G_{spu} &= (G_{spu}^{label_1}, G_{spu}^{label_2}, ..., G_{spu}^{label_j}) \\ G_{spu}^{label_j} &= G_{spu_M}^{label_j} \cup G_{spu_I} \end{aligned} \tag{3}$$

Tables 2 and 3 are examples of G_{cau} and G_{spu} in SA and NLI tasks.

3.1.2. Identification of correlation token set

We obtain the correlation tokens through attention attribution analysis [7], which provides an important observation perspective for the correlation analysis between the model's internal representations and external decisions. Specifically, we analyze the behavior of LMs at the token level by computing and outputting the attention scores. We require our model to output attention scores for each token when predicting labels. Tokens with higher scores contribute more to the predicted label. From these, we extract the Top-K tokens to form the correlation token set associated with the label. This process is formalized as follows.

Define f as a model trained on specific task. For each input example e_i in the corpus D, we obtain the attention scores a_i^1, a_i^2,..., a_i^m corresponding to each token t_i^1, t_i^2,..., t_i^m in e_i, where m represents the token quantity in e_i. We identify the Top-K tokens associated with the prediction labels by analyzing the attention scores, forming a correlation set $\mathrm{C} = (\mathrm{C}^{\mathrm{label}_1}, \mathrm{C}^{\mathrm{label}_2}, ..., \mathrm{C}^{\mathrm{label}_j})$, where $\mathrm{C}^{\mathrm{label}_j}$ is the correlation tokens set related to the j^{th} label.

3.1.3. Neuron localization

We utilize an integrated gradient attribution strategy for neuron localization. We further define the C, which represents the correlation set extracted by the language model f trained on the corpus D.

$$\mathrm{C}_{(\mathrm{D},w_l^k)} = \mathrm{C}_{(\mathrm{D},w_l^k=\hat{w}_l^k)} \tag{4}$$

Here, w_l^k represents the k^{th} neuron in the l^{th} layer of the model f, and $\hat{w}_l^k$ is its value. To compute the gradient attribution score $\mathrm{Attr}(w_l^k)$ for the neuron, we gradually change w_l^k from 0 to its original value $\bar{w}_l^k$ obtained from the language model, while integrating the gradient:

$$\mathrm{Attr}(w_l^k, predict) = \int_{\alpha=0}^{1} \frac{\partial \mathrm{proximity}(\mathrm{C}_{(\mathrm{D},w_l^k=\alpha\bar{w}_l^k)}, \mathrm{G}^{\mathrm{label=predict}})}{\partial w_l^k} d\alpha \tag{5}$$

$$\mathrm{proximity}(\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)}, \mathrm{G}^{\mathrm{label=predict}}) = \frac{|\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)} \cap \mathrm{G}^{\mathrm{label=predict}}|}{|\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)}|} + \frac{\sum_{\mathrm{token}\in(\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)} \cap \mathrm{G}^{\mathrm{label=predict}})} \mathrm{attention}_{\mathrm{token}}}{|\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)} \cap \mathrm{G}^{\mathrm{label=predict}}|} \tag{6}$$

Here, the "predict" refers to the output of the model with parameters w_l^k. $\frac{\partial \mathrm{proximity}(\mathrm{C}_{(\mathrm{D},w_l^k=\alpha\bar{w}_l^k)}, \mathrm{G}^{\mathrm{label=predict}})}{\partial w_l^k}$ computes the gradient of proximity with respect to w_l^k. $\mathrm{G}^{\mathrm{label=predict}} \in \{\mathrm{G}_{\mathrm{cau}}^{\mathrm{label=predict}}, \mathrm{G}_{\mathrm{spu}}^{\mathrm{label=predict}}\}$. If $\mathrm{G}^{\mathrm{label=predict}} = \mathrm{G}_{\mathrm{cau}}^{\mathrm{label=predict}}$, Formula 5 is used to locate *causal correlation neurons*, otherwise, it is used to locate *spurious correlation neurons*. In Formula 6, the term $\mathrm{proximity}(\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)}, \mathrm{G}^{\mathrm{label=predict}})$ represents the alignment degree between $\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)}$ and $\mathrm{G}^{\mathrm{label=predict}}$. Specifically, it consists of two parts. The first part $\frac{|\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)} \cap \mathrm{G}^{\mathrm{label=predict}}|}{|\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)}|}$ is used to calculate token-level alignment degree, which measures the extent to which the elements in the set $\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)}$ also belong to the set $\mathrm{G}^{\mathrm{label=predict}}$. And the second part $\frac{\sum_{\mathrm{token}\in(\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)} \cap \mathrm{G}^{\mathrm{label=predict}})} \mathrm{attention}_{\mathrm{token}}}{|\mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)} \cap \mathrm{G}^{\mathrm{label=predict}}|}$ is used to calculate the attention-level alignment degree, which measures the average attention score of tokens belonging to the $\mathrm{G}^{\mathrm{label=predict}} \cap \mathrm{C}_{(\mathrm{D},\alpha\bar{w}_l^k)}$. An example of the proximity calculation process is shown in Figure 2.

As α varies from 0 to 1, the integration of gradients accumulates the changes in proximity due to the shift in w_l^k. If a neuron significantly affects proximity, Attr(w_l^k, *predict*) will be large, *indicating that the neuron encodes relevant causal or spurious correlation information.*

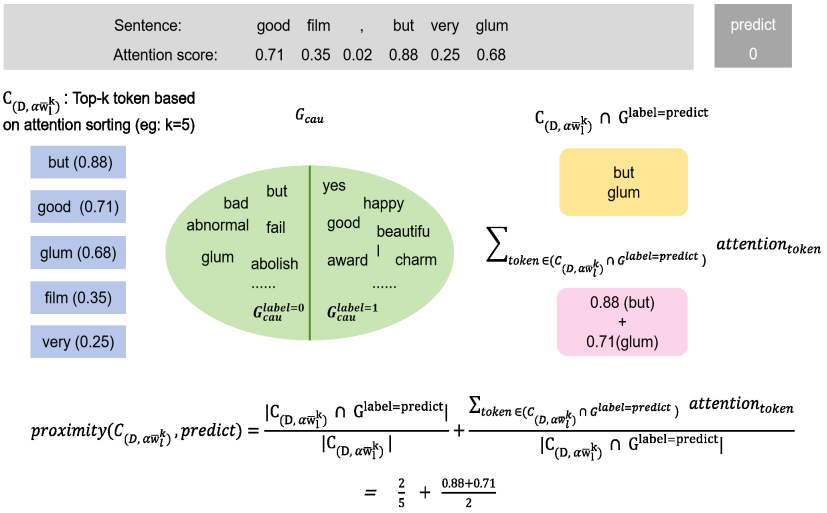


Figure 2: An example of the proximity calculation process. The light gray part contains the tokens along with their corresponding attention scores assigned during the model's prediction. The dark gray part is the model's predicted output. The blue part lists the tokens ranked by their attention scores. The green part represents the golden correlation tokens set $G^{label=predict}$. In this figure, the $G^{label=predict}$=$G_{cau}^{label=predict}$. The yellow part indicates the intersection between $C_{(D,\alpha\bar{w}_l^k)}$ and $G_{cau}^{label=predict}$, and the pink part is the sum of attention values for the elements in the intersection.

3.2. Neuron parameter editing

After locating the neurons, we observe an overlap between neurons storing spurious correlations and those encoding causal correlations. Adjusting the parameters of spurious correlation neurons would inadvertently erase both types of information. *To improve forgetting, we first set the parameters of neurons encoding only spurious correlation information to zero. At the same time, for neurons encoding both spurious and causal correlations, we adopt a neuron parameter editing method based on the PSO*, which can dynamically adjust the parameters of neurons to an optimal combination based on the loss function, making the model forget spurious correlations while maintaining the memory of causal correlations as much as possible.

The PSO algorithm is widely applied in parameter optimization, which is capable of iteratively searching for the optimal solution in a specified search space. The PSO algorithm has the characteristics of low algorithm complexity, fast convergence speed, and strong robustness, and has strong applicability in large-scale optimization problems. These characteristics meet our low computational cost requirements for mitigation strategies. Its core components include the particles, populations, fitness function, iterative process, and stopping criteria. In the context of neuron editing, we formalize the problem as follows:

• Particles and populations: Assuming there are n neurons that need to be edited by PSO. We define $x = (x_1, x_2, \ldots, x_i, \ldots, x_n)$, where x_i is the i^{th} neuron, and x is the value set of a group of neurons, treated as a particle in the optimization process. Multiple particles together form a population, denoted as X.

Each particle consists of two elements: position ($\mathbf{x}$) and velocity ($\mathbf{v}$). The position represents the parameter values of the particle, while the velocity indicates the magnitude and direction of changes in these values within the search space. Together, these components determine the particle's update during each iteration.

• Fitness function: The fitness function f evaluates the quality of a particle's position. The higher the fitness value, the higher the particle position quality, and the closer the particle is to the optimal solution. The fitness function is:

$$f(x) = \text{proximity}(\mathrm{C}_{(\mathrm{D},x)}, \mathrm{G}_{\text{cau}}) - \text{proximity}(\mathrm{C}_{(\mathrm{D},x)}, \mathrm{G}_{\text{spu}}) \tag{7}$$

where C(D, x) is the correlation set derived from dataset D when the model's neuron parameters are set to x.

• Iteration mechanism: The iterative mechanism combines individual learning and group learning. Each particle updates its position and velocity based on its own best position, p_i, and the global best position, g. The particle's best position is the position corresponding to the highest fitness it finds during the search process, while the global best position is the position corresponding to the highest fitness found by the entire population. The iterative mechanism is as follows:

$$\begin{aligned} \mathbf{v}_{i,t+1} &= w\mathbf{v}_{i,t} + c_1 r_1 \left(\mathbf{p}_{i,t} - \mathbf{x}_{i,t}\right) + c_2 r_2 \left(\mathbf{g}_t - \mathbf{x}_{i,t}\right) \\ \mathbf{x}_{i,t+1} &= \mathbf{x}_{i,t} + \mathbf{v}_{i,t+1} \end{aligned} \tag{8}$$

where $\mathbf{v}_{i,t}$ is the velocity of the i^{th} particle at the t^{th} iteration, and w is the inertia weight balancing the current and past velocities. c_1 and c_2 are learning factors adjusting the learning speed, while r_1 and r_2 are random numbers introducing variability. $\mathbf{x}_{i,t}$ represents the position of the i^{th} particle at the t^{th} iteration. Through continuous iterations, the particle position x^* that maximizes f is obtained, which corresponds to the optimal combination of neuron parameters.

$$x^* = \text{argmax}_x f(x) \tag{9}$$

• Stop criteria: We set the maximum iteration number, with the optimization algorithm halting once this limit is reached.

4. Experiments and Analysis

In our experiments, we aim to answer the following research questions: **RQ1**: Can MacU effectively enhance the model's generalization capability? **RQ2**: Can MacU effectively correct spurious correlation? **RQ3**: Is MacU scalable on models with different sizes? **RQ4**: Is spurious correlation information encoded in specific neurons? **RQ5**: Are neurons encoding causal correlations and spurious correlations independent?

4.1. Experimental setup

4.1.1. Tasks and benchmark datasets

In this paper, we conduct experiments on SA and NLI tasks. This is because they are fundamental tasks that can effectively evaluate the linguistic comprehension and reasoning abilities of models. And many complex tasks, such as question answering and machine reading comprehension, can often be reformulated into forms similar to NLI or SA for resolution. Furthermore, these tasks are widely used in research on spurious correlations [33, 1, 32, 5], and experiments conducted on them facilitate fair comparisons among different methods. Specifically, we select the following datasets.

(1) For the SA task, we use SST-2 [23] as the training and validation dataset, and then evaluate the model's generalization on different test datasets. Specifically, the test datasets are divided into in-domain, out-of-domain, and challenge test sets. In this paper, we use the SST-2 as the in-domain test set, and Yelp and IMDB [18] as the out-of-domain test sets. Human-CAD [15] and Contrast [10] are employed as the challenging test sets. The dataset size is shown in Table 4, and the dataset information is as follows.

SST-2 [23]. It is an important benchmark dataset in the sentiment analysis task proposed by Stanford University. This dataset includes approximately 67000 samples, and this paper uses a subset of this dataset for training and validation.

IMDB [18]. It is a commonly used dataset in the field of NLP, mainly used for sentiment analysis tasks. IMDB data mainly comes from information related to movies and TV programs.

Yelp. It is a dataset that contains rich user reviews and merchant information and is widely used for sentiment analysis tasks. It is one of the important datasets for evaluating the performance of NLP models.

Human-CAD [15]. It is a counterfactual dataset obtained by manually modifying existing NLP datasets such as the IMDB and SNLI datasets. This dataset has been used by many studies as a challenging benchmark for spurious correlation mitigation research due to its unique construction method.

Contrast [10]. It is created by Allen AI to evaluate whether a model truly understands semantic logic, rather than relying on spurious correlations.

(2) For the NLI, we use the subset of SNLI [3] as the training and validation datasets, and then evaluate the model's generalization on different test sets. In this paper, we use the SNLI as the in-domain test set, while MNLI [34] is used for the out-of-domain test set. The challenging test sets include Human-CAD, Diagnostic [30], Stress [19], and Break [12]. The dataset size is shown in Table 5, and the dataset information is as follows.

SNLI [3]. It is a large-scale NLI dataset developed by Stanford University, widely used for evaluating models' performance in understanding and inferring language abilities, and is one of the important benchmark datasets in NLP.

MNLI [34]. It is a natural language inference dataset released by Stanford University and New York University. Compared to SNLI, this dataset provides a wider range of contexts and styles, making model training more challenging.

Diagnostic [30]. It is a manually constructed test set designed to evaluate the performance of models on challenging language phenomena such as lexical semantics and logical abilities.

Stress [19]. It is constructed by conducting error analysis on the MNLI and designing adversarial examples, providing a more comprehensive and rigorous standard for model evaluation. And it is used to evaluate the antonyms and number reasoning ability of the model, as well as its dependence on false vocabulary features.

Split	Train	Validation	Test				
Dataset	SST-2	SST-2	SST-2	IMDB	Yelp	Human-CAD	Contrast
Number of samples	8544	1101	1818	487	5166	487	487

Table 4: Datasets size on SA task.

Split	Train	Validation	Test					
Dataset	SNLI	SNLI	SNLI	MNLI	Human-CAD	Diagnostic	Stress	Break
Number of samples	20000	2400	4801	9909	1000	1104	9822	8193

Table 5: Datasets size on NLI task.

Break [12]. It is an adversarial test set used to evaluate the model's vocabulary reasoning ability, which is constructed by replacing a single word in the premise of the SNLI dataset to generate hypotheses.

4.1.2. Baseline methods

In addition to comparing with the **Fine-tuning** method that did not use any spurious correlation mitigation operation, we also compare MacU with three types of approaches.

(1) Pre-processing: - **AutoCAD** [33] generates counterfactual data by modifying causal features, and reduces the language model's reliance on spurious features through retraining the model, and achieves state-of-the-art (SOTA) performance. - **Sentiment-CAD** [37] automatically generates counterfactual SA data and retrains the model to mitigate spurious correlations.

(2) In-processing: - **NFL** [5] is a regularization method that constrains changes in language model parameters or outputs to prevent capturing spurious correlations from misalignment. - **C^{2}L** [6] makes the model more focused on robust features in the data through contrastive learning method.

(3) Post-processing: - **PoE** [14] is an end-to-end unbiased method that adjusts the cross-entropy loss based on predictions from a hypothesis-only biases model, reducing spurious correlations learned during training.

4.1.3. Baseline models

In this paper, we select the baseline models based on a comprehensive consideration of their popularity, architectural diversity, parameter scale, and open-source availability. The baseline models are as follows:

BERT [8]: It is a typical model of the auto-encoding architecture, which provides powerful contextual understanding capabilities through its bidirectional encoding mechanism. Evaluating this model helps assess the applicability of our method on small models.

GPT-2[22]: It is a representative model of auto-regressive generative models. Evaluating it helps us analyze the applicability of our method to the generative tasks.

LLaMA-2-7B[28]: It is a medium-sized model with roughly 7B parameters. It enjoys considerable influence and widespread adoption within the open-source community.

Qwen-2-1.5B[27]: It is a small-scale open-source large language model with approximately 1B parameters. It holds significant influence in the open-source community and allows us to validate the performance of our method on LLM with relatively limited parameters.

Accuracy on the SA task						
Methods	**In-Domain**	**Out-of-Domain**		**Challenge**		**Avg.**
	SST-2	IMDB	Yelp	Human-CAD	Contrast	
BERT-base						
Fine-tuning	90.30	84.01	86.51	74.13	70.94	81.17
AutoCAD	89.29	82.93	85.11	78.44	73.50	81.85
Sentiment-CAD	88.73	83.93	84.73	78.16	73.09	81.73
NFL	88.24	84.02	87.10	**81.81**	73.86	83.01
MacU (Ours)	**90.52**	**87.68**	**87.71**	80.17	**75.42**	**84.30**
AutoCAD +MacU	89.74	86.60	87.28	82.75	79.05	85.08
Sentiment-CAD +MacU	87.73	86.93	83.73	81.16	78.09	83.53
NFL+MacU	88.51	84.62	88.53	81.94	77.53	84.23

Table 6: The performance of BERT models trained using different methods on the SA task.

GLM-4-9B[11]: It is a large-scale model with around 9B parameters. Along with LLaMA-2, it is recognized as a strong baseline in both Chinese and English research communities. Its inclusion allows a comprehensive validation of the effectiveness and competitiveness of our proposed method.

All baseline models are open-source, ensuring the reproducibility of the experiments and the fairness of comparisons. Specifically, we first train the BERT models with various spurious correlation mitigation methods on both SST-2 and SNLI datasets, and subsequently evaluate the generalizability of these methods on different test data. Additionally, we also fine-tune the GPT-2, LLaMA-2-7B, Qwen-2-1.5B, and GLM-4-9B models using the MauU method to verify the applicability and effectiveness of MauU on larger-scale models.

4.1.4. Parameter settings

We train the Fine-tuning model using a batch size of 32, a training epoch of 10, and the initial learning rate of 1e-5. In the PSO optimization algorithm, we set the maximum number of iterations to 40, the particle optimization interval to [-1,1], c_1 and c_2 to 1.5, and w to 0.5.

4.1.5. Metrics

The method's effectiveness is assessed based on the following metrics.

Accuracy [33]. *Accuracy* is used to assess the performance of the spurious correlation mitigation method on various test sets.

Average accuracy (*Avg*) [33]. *Avg* is the average performance of spurious correlation mitigation method on different test sets, used to evaluate the generalization.

Average proximity (η). η proposed in this paper is used to quantify the correction degree of spurious correlations by mitigation method.

4.2. RQ1: Performence and generalization comparison

Tables 6 and 7 show the results of the BERT model (trained with various spurious correlation mitigation methods on the SST-2 and SNLI datasets) on different test sets. We evaluate the generalization of the model using the *Avg*. in the tables. The result analyses are as follows.

In-Domain: From the Tables 6 and 7, we observe that on the BERT model, compared to the Fine-tuning, the performance of other methods *decreases*, whereas MacU shows a *slight*

Methods	Accuracy on the NLI task						
	In-Domain	Out-of-Domain	Challenge				*Avg.*
	SNLI	MNLI	Human-CAD	Diagnostic	Stress	Break	
BERT-base							
Fine-tuning	84.37	61.85	56.35	49.18	59.42	60.45	61.93
AutoCAD	82.85	59.58	58.25	48.01	57.15	53.66	59.92
NFL	83.21	62.33	58.62	51.26	60.00	**65.07**	63.41
C^2L	84.12	64.98	58.11	51.65	56.33	58.47	62.27
PoE	83.40	64.10	**66.20**	50.36	37.95	63.13	60.86
MacU (Ours)	**84.76**	**65.27**	59.43	**52.26**	**60.03**	62.14	**63.98**
AutoCAD + MacU	82.25	60.27	62.43	53.26	59.13	57.59	62.49
NFL + MacU	83.29	71.07	59.24	53.24	48.07	68.06	63.82
C^2L + MacU	83.85	68.43	58.11	52.32	56.02	59.01	62.95
PoE + MacU	83.79	65.34	67.15	51.23	42.98	64.60	62.51

Table 7: The performance of BERT models trained using different methods on the NLI task.

improvement on in-domain datasets. This suggests that the MacU method, based on machine unlearning, is more effective than the other methods.

Out-of-Domain and Challenge: From the model's results on the out-of-domain datasets, we observe that our method performs better. In Table 6, MacU outperform the Fine-tuning by 3.67% *(87.68%-84.01%)* and 1.20% *(87.71%-86.51%)* on the IMDB and Yelp, respectively. In Table 7, MacU surpasses the Fine-tuning by 3.42% *(65.27%-61.85%)* on MNLI. Additionally, MacU performs well on the challenge datasets. Notably, MacU achieves the highest *Avg*. score, surpassing the Fine-tuning by 3.13% *(84.30%-81.17%)* on the SA task and 2.05% *(63.98%-61.93%)* on the NLI task, demonstrating our machine unlearning-based method has superior generalization ability.

In the tables, we find that although MacU outperformed other methods overall, it does not achieve optimal performance on all datasets. This may be due to the characteristics of the datasets. We discuss this issue in section 4.8.

Method+MacU. Furthermore, we enhance models trained using other methods with MacU, observing a noticeable performance improvement. For instance, the *Avg*. accuracy of Auto-CAD + MacU surpasses that of Auto-CAD by 3.23% *(85.08%-81.85%)* and 2.57% *(62.49%-59.92%)* on the SA and NLI tasks, respectively. This suggests that models trained with other mitigation strategies still retain some spurious correlations, which can be further eliminated by MacU. Moreover, this demonstrates our approach is both *plug-and-play and effective.*

Overall, it can be seen that MacU can effectively alleviate the spurious correlation of the model and improve its generalization ability.

4.3. RQ2: Spurious correlation correction capability comparison

In this section, we explore to what extent various mitigation methods corrected the spurious correlations internalized by the model.

The average proximity η between C and G_{spu} quantifies the extent to which the model's behavior reflects spurious correlations. A lower value of η indicates that the model internalizes less spurious correlation information, demonstrating stronger correction ability. Specifically, the

Average proximity (η)		
Methods	SNLI	SST-2
Fine-tuning	0.98	0.91
AutoCAD	1.01	1.05
Sentiment-CAD	-	1.02
C^2L	0.88	-
NFL	1.29	0.97
PoE	1.01	-
MacU (Ours)	0.84	0.87

Table 8: Average proximity η of models using different mitigation strategies. "-" represents no experimental results, as these methods are not applicable on SST-2. To facilitate comparison, we will normalize this value.

metric η for average accuracy is defined as:

$$\eta = \text{proximity}(\text{C}, \text{C}_{\text{spu}})/len(\text{D}) \tag{10}$$

Table 8 shows the η values of models trained with different strategies. *It can be observed that the model trained using MacU has the smallest η, indicating superior correction capability.* In contrast, other strategies exhibit weaker correction abilities. For instance, the η of Auto-CAD is higher than that of Fine-tuning, suggesting that the model's spurious correlation information has not been effectively removed. This could be due to the fact that when Auto-CAD introduces counterfactual data, it may also introduce additional spurious correlations.

4.4. RQ3: MacU's Scalability on models with different sizes

We further extend our method to the LLMs. Tables 9 and 10 present the performance of GPT-2, LLaMA-2-7B , Qwen-2-1.5B, and GLM-4-9B models fine-tuned using MacU on the SST-2 and SNLI datasets, respectively. *As observed, similar to the results with BERT, the LLMs fine-tuned using the MacU method also showed better generalization ability.*

It should be pointed out that due to the high computational complexity of the C^2L strategy, even with four A100, it is difficult to run on LLMs. Therefore, the relevant results are not presented in the table. In contrast, the MacU strategy only requires a single A100. This comparison not only highlights the advantages of our method in computational efficiency, but also demonstrates its deployment convenience in practical application scenarios.

From Tables 9 and 10, we find that some methods exhibit different trends in performance on LLM models compared to the BERT models, such as the Auto-CAD and Sentiment-CAD, which significantly reduce generalization on larger models. This may be due to the mismatch between the generated counterfactual data size and model requirements, as the LLMs often rely more on spurious correlation patterns established in the pre-training stage rather than learning effective causal representations from limited new samples.

4.5. RQ4: Neural encoding mechanism for spurious correlation information

We design the following experiments to confirm that spurious correlations or causal correlations are indeed encoded in specific neurons.

Accuracy on the SA task						
Methods	**In-Domain**	**Out-of-Domain**		**Challenge**		*Avg.*
	SST-2	IMDB	Yelp	Human-CAD	Contrast	
GPT − 2						
Fine-tuning	**90.3**	67.82	74.16	65.95	61.75	71.99
AutoCAD	87.85	67.39	66.71	64.22	63.03	69.84
Sentiment-CAD	80.46	72.54	71.12	51.33	60.78	67.24
NFL	89.79	68.36	72.43	66.48	62.78	71.93
MacU (Ours)	90.20	**69.67**	**75.97**	**69.23**	**65.26**	**74.06**
LLaMA − 2 − 7*B*						
Fine-tuning	95.99	93.23	95.49	93.95	**92.62**	94.25
AutoCAD	95.82	93.03	94.81	92.52	90.37	93.31
Sentiment-CAD	95.44	90.78	91.37	87.09	78.28	88.59
NFL	95.76	93.86	94.98	93.15	91.47	93.84
MacU (Ours)	**96.32**	**94.88**	**96.64**	**94.56**	91.98	**94.87**
Qwen − 2 − 1.5*B*						
Fine-tuning	94.28	95.49	97.13	92.82	91.39	94.22
AutoCAD	93.82	91.59	94.03	87.50	88.72	91.53
Sentiment-CAD	93.34	90.57	94.46	81.55	81.14	88.21
NFL	93.67	95.48	95.97	**93.57**	91.54	94.04
MacU (Ours)	**94.35**	**96.73**	**97.94**	93.13	**93.26**	**95.08**
GLM − 4 − 9*B*						
Fine-tuning	**97.69**	95.49	97.81	95.29	94.06	96.06
AutoCAD	92.10	**97.38**	**98.16**	96.77	93.28	95.53
Sentiment-CAD	90.83	96.47	94.19	92.00	84.40	91.57
NFL	97.50	95.55	97.70	94.82	93.95	96.20
MacU (Ours)	97.59	96.67	98.07	**97.03**	**94.86**	**96.84**

Table 9: The performance of GPT-2, LLaMA-2-7B, Qwen-2-1.5B, and GLM-4-9B models trained using different methods on the SA task.

Accuracy on the NLI task							
Methods	**In-Domain**	**Out-of-Domain**		**Challenge**			*Avg.*
	SNLI	MNLI	Human-CAD	Diagnostic	Stress	Break	
			GPT − 2				
Fine-tuning	72.20	48.83	74.93	44.50	32.87	25.3	49.77
AutoCAD	66.30	43.66	65.75	38.49	34.30	22.96	45.24
NFL	68.49	49.26	73.28	46.58	35.56	26.68	50.02
PoE	69.57	50.21	**76.89**	43.87	28.96	27.89	49.56
MacU (Ours)	**72.86**	**50.58**	75.85	**47.41**	**36.15**	**29.61**	**52.07**
			LLaMA − 2 − 7*B*				
Fine-tuning	91.21	83.71	77.10	63.04	81.92	93.81	81.79
AutoCAD	91.05	83.95	77.35	63.28	82.15	**93.67**	82.01
NFL	90.87	83.52	76.89	62.75	81.63	91.45	81.18
PoE	90.42	83.10	76.50	62.30	81.25	90.20	80.62
MacU (Ours)	**91.78**	**85.62**	**78.05**	**64.12**	**83.89**	93.15	**82.76**
			Qwen − 2 − 1.5*B*				
Fine-tuning	90.56	80.78	71.27	59.87	78.46	87.75	78.11
AutoCAD	**90.89**	81.19	71.67	61.95	79.13	86.63	78.57
NFL	90.61	81.02	71.55	60.05	78.95	87.80	78.33
PoE	89.78	80.28	70.88	59.15	78.70	87.95	77.79
MacU (Ours)	90.48	**82.54**	**72.94**	**62.27**	**79.89**	**88.26**	**79.56**
			GLM − 4 − 9*B*				
Fine-tuning	92.29	85.76	75.48	65.85	83.20	89.88	82.07
AutoCAD	77.49	88.97	70.50	81.71	85.41	83.08	81.19
NFL	91.85	86.05	76.20	65.10	83.75	89.95	82.15
PoE	91.68	85.25	75.85	64.50	81.90	**90.10**	81.56
MacU (Ours)	**92.34**	**87.98**	**77.62**	**67.35**	**84.69**	89.71	**83.28**

Table 10: The performance of GPT-2, LLaMA-2-7B, Qwen-2-1.5B, and GLM-4-9B models trained using different methods on the NLI task.

Average proximity		
Methods	SNLI	SST-2
Ori	1.06	1.01
Zero	0.90	0.95
Amplified	1.03	1.03

Table 11: Average proximity between the correlation token extracted by the model and the golden spurious correlation when the neuron parameters change. To facilitate comparison, we will normalize this value.

Activation values				
Samples	golden label	prediction	Activation values of causal neurons	Activation values of spurious neurons
unflinchingly bleak and desperate.	0	1	0.067	0.1240
What is 100% missing here is a script of the most elemental literacy, an inkling of genuine wit...	0	1	0.070	0.100
but taken as a stylish and energetic one-shot , the queen of the damned can not be said to suck.	1	0	0.062	0.086
good film , but very glum.	1	0	0.103	0.155

Table 12: The activation values of causal or spurious correlation neurons under different inputs.

(1) **From neuron changes to model behaviors.** We discuss the impact of changes in neuron parameters on the extracted correlation tokens by the model. Specifically, we manipulate the parameters of the identified spurious correlation neurons by setting them to 0 or amplifying their values. Then, we observe the changes in the correlation tokens extracted by the model. Table 11 shows the average proximity between the identified correlation tokens and the golden spurious correlation tokens during prediction after the identified spurious correlation neurons have been set to 0 or amplified. (The smaller the average proximity value, the weaker the memory of spurious correlation information). It can be seen that after the neurons are set to 0, the average proximity decreases, indicating that the model has forgotten the spurious correlation information. When the neurons are amplified, the average proximity increases, indicating that the model's memory of the spurious correlation information is also enhanced. These phenomena indicate that these neurons do indeed encode spurious correlation information. (2) **From predictive behaviors to neuron activation state.** We discuss the activation state of neurons when the model predicts incorrectly. Specifically, we observe the model's output and analyze the activation effect of the samples on the identified causal (or spurious correlation) neurons when the model makes incorrect predictions. We analyze the activation values of several samples on the identified causal correlation neurons and spurious correlation neurons. The samples and their

corresponding activation values are presented in Table 12.

In Table 12, when the sentence "*unflinchingly bleak and desperate*" is input into the model, we observe that the model incorrectly predicted a positive emotion. Through a detailed analysis of the neuron activation patterns induced by this sentence, we find that the activation level of spurious correlation neurons is significantly higher than that of causal neurons. This result suggests that the model may not have fully understood the true emotional meaning of the sentence but instead relied on the spurious correlations between the token "*unflinchingly*" in the sample and the labels "*positive*" to make inferences, leading to an incorrect answer.

Based on the above analysis, we find that suppressing and amplifying located spurious correlation neurons notably affects the extraction of the spurious tokens. Besides, we have observed that when the model predicts incorrectly, neurons associated with spurious correlations tend to be activated more readily than those related to causal correlations. *These findings support our hypothesis in section 1 that spurious and causal correlation information can be encoded in particular neuron groups.*

4.6. RQ5: Distribution characteristics of causal neurons and spurious correlation neurons

We visualize the distribution of neurons encoding causal correlations and spurious correlations in Figures 3 and 4. Figure 3 shows the independent distribution of spurious correlations neurons and causal correlations neurons in BERT trained on SST-2 and SNLI datasets. Overall, both types of neurons exhibit a pyramid-like distribution structure, with the majority concentrated in layers 8 to 12. Further comparison between Figure 3a and Figure 3c, as well as between Figure 3b and Figure 3d, we can observe that causal correlation neurons are predominantly distributed in the deeper layers of the FFN, whereas spurious correlation neurons are mainly located in the shallower layers of the network.

Figure 4 shows the joint distribution of causal correlation neurons and spurious correlation neurons in the BERT model trained on the SST-2 and SNLI datasets. *We observe an overlap between neurons encoding causal correlation and spurious correlation information, indicating that modifying spurious correlation neurons inevitably influences the model's memory of causal correlations.* This suggests that a simple approach of setting all neurons to "zero" to eliminate spurious correlations is not suitable.

4.7. Other comparative experiments

We conduct comparative experiments on neuron localization and neuron parameter editing methods.

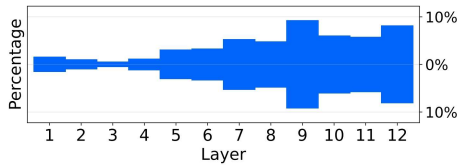
4.7.1. Comparison of localization methods

We compare the neuron localization method guided by the proximity between correlation tokens identified by the model and golden correlation tokens with the method guided by the model's performance on the spurious-free dataset (The data are from the Human-CAD dataset).

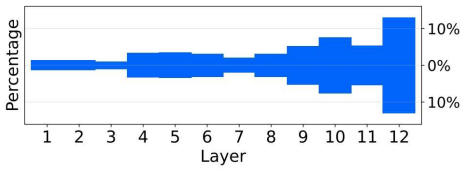
Table 13 indicate that the localization method guided by the proximity outperforms the performance-guided method, demonstrating its superior accuracy and effectiveness.

4.7.2. Comparison of neuron parameter editing methods

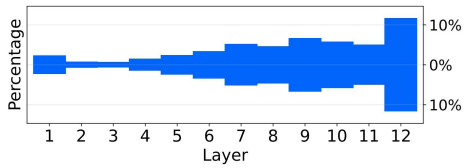
We compare the neuron editing method based on the PSO algorithm to the common "Zero" method used in other machine unlearning studies, where neuron parameters are simply set to zero. The results in Table 14 show that when the parameters of all identified spurious correlation



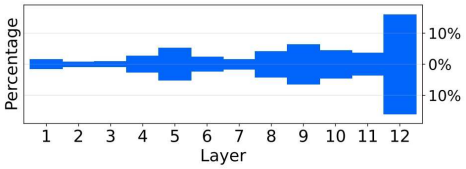
(a) Spurious correlation neuron distribution map in BERT trained on SST-2 dataset



(b) Spurious correlation neuron distribution map in BERT trained on SNLI dataset

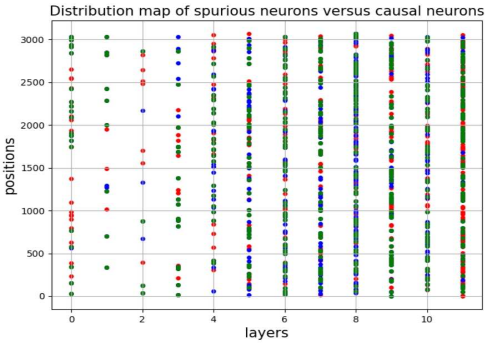


(c) Causal correlation neuron distribution map in BERT trained on SST-2 dataset

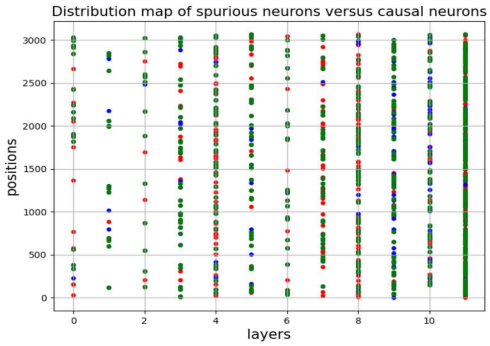


(d) Causal correlation neuron distribution map in BERT trained on SNLI dataset

Figure 3: Distribution of spurious and causal correlation neurons in BERT, showing a pyramid-like structure. Causal correlation neurons are predominantly in deeper FFN layers, while spurious correlation neurons are clustered in shallower layers.



(a) The joint distribution map of neurons in the BERT model trained on the SST-2 dataset



(b) The joint distribution map of neurons in the BERT model trained on the SNLI dataset

Figure 4: The joint distribution of causal correlation neurons and spurious correlation neurons in BERT. The red dots represent causal correlation neurons, blue dots represent spurious correlation neurons, and green dots represent neurons storing both causal and spurious correlations simultaneously. It can be observed that causal correlation neurons (red dots) are predominantly distributed in deeper FFN layers, while those spurious correlation neurons (blue dots) are located in shallower layers. Additionally, there is an overlap between these two types of neurons (green dots).

Accuracy on in-domain dataset		
Methods	SNLI	SST-2
Localization guided by performance on samples	83.16	88.63
Localization guided by proximity	84.76	90.52

Table 13: Bert model performance under different neuron localization strategies.

Accuracy on in-domain dataset		
Methods	SNLI	SST-2
Ori	84.37	90.30
Zero	83.20	89.46
PSO(Ours)	84.76	90.52

Table 14: Performance of BERT model with different neuron parameter editing strategies. In the Table, "Ori" refers to the scenario where no adjustments are made to the neuron parameters. "Zero" is to set the located spurious correlation neurons to 0.

neurons are set to zero, model performance decreases compared to the neuron parameter editing strategy based on the PSO algorithm. This suggests that setting the parameters to zero not only eliminates spurious correlations but also inadvertently disrupts certain causal correlations, leading to a decline in performance. *In contrast, the PSO-based parameter editing strategy fine-tunes the parameters, optimizing them to achieve a balanced state that allows the model to effectively forget spurious correlations while retaining crucial causal information.*

4.8. How do dataset characteristics affect the performance of spurious correlation mitigation methods?

We explore the impact of dataset characteristics on the performance of spurious correlation mitigation strategies.

In Table 7, we find that the PoE outperforms MacU on the Human-CAD and Break dataset. To investigate the reasons for this phenomenon, we quantify the degree of hypothesis-only bias contained in the nli test datasets used in this paper. Table 15 shows the performance of the Fine-tuning model in different NLI datasets when the input is the original dataset versus when the input contains only the hypothetical dataset. Observing the the performence on the hypothesis-only input, a higher performence indicates that the BERT model contains more hypothesis-only bias. It can be seen that compared to other datasets, the Human-CAD and Break datasets contain more such hypothesis-only biases. *The PoE method focuses on reducing hypothesis-only biases, making it more effective for these two datasets.* Furthermore, we conduct the experiment on Human-CAD to evaluate the capability of the MacU in handling the "hypothesis-only bias" . We first select the samples that the model predicts incorrectly under the hypothesis-only input setting. Then, we use the corresponding versions of these error samples from the original Human-CAD dataset and construct a subset. On this subset, a correct prediction indicates that the model remains unaffected by the hypothesis-only bias, whereas an incorrect prediction suggests that the model may either rely on this bias or lack sufficient capability to make correct predictions. The results of the model's performance on this subset are summarized in Table 16, and it shows that the MacU demonstrates a certain ability to mitigate hypothesis-only bias.

Methods	Accuracy					
	In-Domain	Out-of-Domain	Challenge			
	SNLI	MNLI	Human-CAD	Diagnostic	Stress	Break
Original input	84.37	61.85	56.35	49.18	59.42	60.45
hypothesis-only input	45.34	40.61	46.64	34.51	39.98	76.49

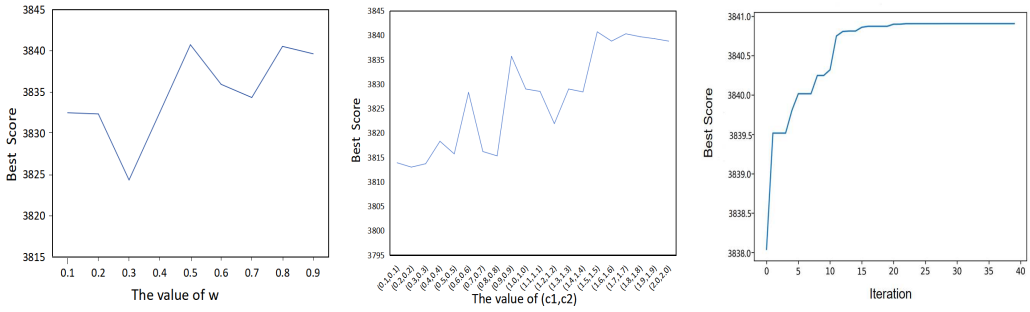
Table 15: Fine-tuned BERT model performance (Accuracy) under original input and hypothesis-only input.

Methods	Accuracy
Fine-tuning	52.15
MacU	58.97
POE	69.54

Table 16: Performance of BERT models using different mitigation methods on the constructed Human-CAD subset. Higher performance indicates a greater capability of the model to mitigate hypothesis-only bias.

Besides, in Table 7, the performance of the NFL outperforms the MacU on the Break dataset, which may be related to the dataset construction method. The Break is an NLI dataset constructed through the automatic replacement of words, where the replacement words are collected from online resources for English learning. This automatic replacement may involve incorrect clustering of many tokens. NFL is an effective regularization method that reduces spurious correlations by preventing erroneous clustering. *Therefore, NFL may achieve better performance on the Break dataset.*

4.9. Sensitivity experiments of the parameters in PSO



(a) The impact of changes in the w on fitness score.

(b) The impact of changes in (c_1, c_2) value on fitness scores.

(c) The impact of changes in iteration number value on fitness scores.

Figure 5: The impact of parameter values (w, c_1, c_2, and iteration number) in the PSO algorithm on fitness. The horizontal axis represents the changing values of the parameter, and the vertical axis represents the corresponding best fitness score.

Methods	data processing time	model training/editing time	overall time
AutoCAD	6.6 h	0.52 h	7.12 h
NFL	-	2.50 h	2.50 h
C^2L	1.54 h	1.13h	2.67 h
POE	0.35 h	0.78 h	1.13 h
MacU	-	1.91 h	1.91 h

Table 17: Comparison of time consumption for adapting BERT using different spurious correlation mitigation methods on NLI.

In this section, we conduct parameter sensitivity experiments to analyze how the best fitness score varies with the parameters w, c_1, c_2, and iteration number. The experimental results on Table 5 show that the fitness score reaches its maximum when the iteration number is 40, the inertia weight w is 0.5, and both learning factors c_1 and c_2 are set to 1.5. Accordingly, this parameter combination is adopted as the final configuration for the PSO algorithm in this paper.

4.10. Comparison of the time consumption

Under the experimental setting employing the BERT model and the NVIDIA GeForce RTX 3090 GPU on the SNLI dataset, we compare the time consumption of various spurious correlation mitigation methods. As shown in Table 17, our method achieves the second-lowest time consumption among all compared methods, superseded solely by the POE method. The primary time consumption of our method is concentrated on the editing and optimization of model parameters, which fundamentally enhances the core capabilities of the model. In comparison, the POE method only adjusts the model's output without changing its parameters, which limits its ability to generalize to different scenarios.

5. Discussion and Limitations

The MacU strategy effectively mitigates the model's spurious correlations and demonstrates broad applicability. To further elucidate the technical boundaries and limitations of this study, we will conduct in-depth discussions around the following issues.

Human-Centric golden correlation token set construction. This paper adopts a research method centered on human cognition in neuron localization, which combines LLM-assisted discovery and manual verification to construct a set of golden correlation tokens aimed at accurately locating neurons encoding relevant information. The design of this method is mainly based on the following considerations: 1) The existing model-driven causal token extraction methods are often limited by endogenous biases in the model, which may lead to incorrect token extraction. 2) Causal correlations have dynamic evolutionary characteristics, and their semantic boundaries are constantly reconstructed with social context and technological evolution. Taking the evolution of the semantic network of "apple" as an example, before the electronic age, this concept was mainly related to the stable category of "fruit". However, with the rise of electronic technology companies, their semantic focus has shifted towards the dimension of "electronic products", and traditional methods are difficult to effectively capture this "concept drift" phenomenon. *In the future, we will explore the human-machine collaboration method to achieve efficient and*

accurate discovery of causal correlation tokens by controlling the timing and degree of human intervention.

Applicability of spurious correlation in individual semantic units. Our spurious correlation mitigation method focuses on the spurious correlations between individual semantic units (such as "failure" and "negative"), without fully considering the spurious correlations issues of compound semantic units (such as "not fail" and "positive"). The reason is that spurious correlations are common between two individual semantic units, and we'd like to investigate whether the MacU method can effectively solve this kind of spurious correlation. *In the future, we will continue to explore the MacU method on the spurious correlations in compound semantic units.*

Analysis of model behavior based on attention attribution. In section 3.1.2, we use the attention attribution method to analyze model behavior. This method is widely used in academic research and industrial applications due to its high computational efficiency and ease of integration. However, there may be a lack of stable causal correlation between attention weights and model predictions. *In the future, we will develop more reliable model interpretation methods, striving to improve the reliability and scientificity of interpretation results while maintaining the practicality of the methods.*

6. Conclusions and Future Work

For the issue of spurious correlations between training data and labels that language models may internalize, we propose an innovative strategy, MacU, which effectively mitigates these spurious correlations. The MacU strategy incorporates a novel neuron localization method, guided by the proximity between correlation tokens identified by the model and golden correlation tokens, alongside a neuron editing approach leveraging the PSO algorithm. A large number of experiments have demonstrated the superiority of our strategy in generalization and demonstrated strong correction capabilities. Moreover, MacU does not require retraining the model or constructing counterfactual datasets, making it both efficient and effective. In the future, we will develop a strategy with more reliable model interpretation methods and solve the problem of spurious correlation in compound semantic units.

Author Contribution

Xinyi Sun is responsible for designing the study, implementing the methods, conducting experiments, analyzing the case data, and drafting the manuscript. Tan Hongye is responsible for designing the study and providing guidance to the project. Dongzhi Han and Zhichao Yan contribute to the data collection and pre-processing. Xiaoli Li, Ru Li and Hu Zhang make contributions to the manuscript review and Editing.

Acknowledgements

We thank all the anonymous reviewers for their constructive comments and suggestions. This work is supported by the National Natural Science Foundation of China (62076155), Natural Language Processing Innovation Team (Sanjin Talents) Project of Shanxi Province, the Science and Technology Cooperation and Exchange Special Project of ShanXi Province (202204041101016).

References

[1] Bae, S., Choi, Y., Kim, H., Lee, J.-H., 2025. Salad: Improving robustness and generalization through contrastive learning with structure-aware and llm-driven augmented data. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 12724–12738.

[2] Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N., 2021. Machine unlearning. In: 2021 IEEE symposium on security and privacy (SP). IEEE, pp. 141–159.

[3] Bowman, S. R., Angeli, G., Potts, C., Manning, C. D., 2015. A large annotated corpus for learning natural language inference. In: Conference on Empirical Methods in Natural Language Processing, EMNLP 2015. Association for Computational Linguistics (ACL), pp. 632–642.

[4] Chen, R., Yang, J., Xiong, H., Bai, J., Hu, T., Hao, J., Feng, Y., Zhou, J. T., Wu, J., Liu, Z., 2023. Fast model debias with machine unlearning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. pp. 14516–14539.

[5] Chew, O., Lin, H.-T., Chang, K.-W., Huang, K.-H., 2024. Understanding and mitigating spurious correlations in text classification with neighborhood analysis. In: Findings of the Association for Computational Linguistics: EACL 2024. pp. 1013–1025.

[6] Choi, S., Jeong, M., Han, H., Hwang, S.-w., 2022. C2l: Causally contrastive learning for robust text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. pp. 10526–10534.

[7] Clark, K., Khandelwal, U., Levy, O., Manning, C. D., 2019. What does bert look at? an analysis of bert's attention. In: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. Association for Computational Linguistics, p. 276.

[8] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 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). pp. 4171–4186.

[9] Du, M., He, F., Zou, N., Tao, D., Hu, X., 2023. Shortcut learning of large language models in natural language understanding. Communications of the ACM 67 (1), 110–120.

[10] Gardner, M., Artzi, Y., Basmov, V., Berant, J., Bogin, B., Chen, S., Dasigi, P., Dua, D., Elazar, Y., Gottumukkala, A., et al., 2020. Evaluating models' local decision boundaries via contrast sets. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 1307–1323.

[11] GLM, T., Zeng, A., Xu, B., Wang, B., Zhang, C., Yin, D., Zhang, D., Rojas, D., Feng, G., Zhao, H., et al., 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. arXiv preprint arXiv:2406.12793.

[12] Glockner, M., Shwartz, V., Goldberg, Y., 2018. Breaking nli systems with sentences that require simple lexical inferences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 650–655.

[13] Golatkar, A., Achille, A., Soatto, S., 2020. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 9301–9309.

[14] Karimi Mahabadi, R., Belinkov, Y., Henderson, J., 2020. End-to-end bias mitigation by modelling biases in corpora. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 8706–8716.

[15] Kaushik, D., Hovy, E. H., Lipton, Z. C., 2019. Learning the difference that makes a difference with counterfactually-augmented data. CoRR abs/1909.12434.

[16] Kim, J. M., Koepke, A., Schmid, C., Akata, Z., 2023. Exposing and mitigating spurious correlations for cross-modal retrieval. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2585–2595.

[17] Liu, S., Yao, Y., Jia, J., Casper, S., Baracaldo, N., Hase, P., Yao, Y., Liu, C. Y., Xu, X., Li, H., et al., 2025. Rethinking machine unlearning for large language models. Nature Machine Intelligence, 1–14.

[18] Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., Potts, C., 2011. Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. pp. 142–150.

[19] Naik, A., Ravichander, A., Sadeh, N., Rose, C., Neubig, G., 2018. Stress test evaluation for natural language inference. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 2340–2353.

[20] Pang, J., Yang, X., Qiu, X., Wang, Z., Huang, T., 2024. Mmaf: Masked multi-modal attention fusion to reduce bias of visual features for named entity recognition. Data Intelligence 6 (4), 1114–1133.

[21] Puli, A., Zhang, L., Wald, Y., Ranganath, R., 2023. Don't blame dataset shift! shortcut learning due to gradients and cross entropy. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. pp. 71874–71910.

[22] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI blog 1 (8), 9.

[23] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C., 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing. pp. 1631–1642.
[24] Stacey, J., Minervini, P., Dubossarsky, H., Riedel, S., Rocktäschel, T., 2020. Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8281–8291.
[25] Sun, X., Tan, H., Guo, Y., Qiang, P., Li, R., Zhang, H., 2025. Mitigating shortcut learning via smart data augmentation based on large language model. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 8160–8172.
[26] Sun, Z., Xiao, Y., Li, J., Ji, Y., Chen, W., Zhang, M., 2024. Exploring and mitigating shortcut learning for generative large language models. In: Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024). pp. 6883–6893.
[27] Team, Q., 2024. Qwen2 technical report. arXiv preprint arXiv:2407.10671.
[28] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
[29] Unnikrishnan, B., Tadic, T., Patel, T., Duhamel, J., Kandel, S., Moayedi, Y., Brudno, M., Hope, A., Ross, H., McIntosh, C., et al., 2024. Shortcut learning in medical ai hinders generalization: method for estimating ai model generalization without external data. NPJ Digital Medicine 7 (1), 124–124.
[30] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S., 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355.
[31] Wang, D., Tan, D., Liu, L., 2018. Particle swarm optimization algorithm: an overview. Soft computing 22 (2), 387–408.
[32] Wang, T., Sridhar, R., Yang, D., Wang, X., 2021. Identifying and mitigating spurious correlations for improving robustness in nlp models. arXiv preprint arXiv:2110.07736.
[33] Wen, J., Zhu, Y., Zhang, J., Zhou, J., Huang, M., 2022. Autocad: Automatically generate counterfactuals for mitigating shortcut learning. In: Findings of the Association for Computational Linguistics: EMNLP 2022. pp. 2302–2317.
[34] Williams, A., Nangia, N., Bowman, S. R., 2018. A broad-coverage challenge corpus for sentence understanding through inference. In: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018. Association for Computational Linguistics (ACL), pp. 1112–1122.
[35] Wu, X., Li, J., Xu, M., Dong, W., Wu, S., Bian, C., Xiong, D., 2023. Depn: Detecting and editing privacy neurons in pretrained language models. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 2875–2886.
[36] Xu, J., Wu, Z., Wang, C., Jia, X., 2024. Machine unlearning: Solutions and challenges. IEEE Transactions on Emerging Topics in Computational Intelligence 8 (3), 2150–2168.
[37] Yang, L., Li, J., Cunningham, P., Zhang, Y., Smyth, B., Dong, R., 2021. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. 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). pp. 306–316.
[38] Yu, Z., Ananiadou, S., 2025. Understanding and mitigating gender bias in llms via interpretable neuron editing. arXiv preprint arXiv:2501.14457.

Author Biography

Xinyi Sun is currently a PH.D. student at Shanxi University and received the M.S. degree from Shanxi Agricultural University, China, in 2019. She has participated in the project of the Science and Technology Cooperation and Exchange Special Project of ShanXi Province. Her research interests include natural language processing and language model faithfulness.

Hongye Tan is currently a professor at Shanxi University, China, and she received her Ph.D. degree from Harbin Institute of Technology in 2008. She has published more than 60 papers in top international academic conferences and journals. Her research interests include natural language processing, language model faithfulness, and multimodal hallucination.

Dongzhi Han is currently a postgraduate at Shanxi University. Her research interests include natural language processing and language model faithfulness.

Zhichao Yan is currently working toward the PhD degree with the School of Computer and Information Technology, Shanxi University, Shanxi. His research interests include question answering and frame semantic parsing.

Xiaoli Li is currently an IEEE fellow and holds the positions of Director and Chief Scientist at the Institute of Machine Intelligence in the Institute for InfoComm Research at A*STAR, Singapore. He received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 2001. His research interests include AI, data mining, machine learning, and bioinformatics.

Ru Li is currently a professor at Shanxi University, China, and she received the Ph.D. degree from Shanxi University, China, in 2012. She has published more than 100 papers in top international academic conferences and journals and received two prizes for scientific and technological progress in Shanxi. Her research interests include natural language processing, knowledge graphs, and information extraction.

Hu Zhang is currently a professor at Shanxi University, China. He has published more than 80 papers in top international academic conferences and journals. His research interests include natural language processing, big data mining, and analysis.