Mitigating Shortcut Learning via Smart Data Augmentation based on Large Language Model

Xinyi Sun[1], Hongye Tan[1,2,3]*, Yaxin Guo[1], Pengpeng Qiang[1], Ru Li[1,2,3], Hu Zhang[1,3]

[1]School of Computer and Information Technology, Shanxi University, Taiyuan, China
[2]Institute of Intelligent Information Processing, Shanxi University, Taiyuan, China
[3]Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan, China
995633034@qq.com, tanhongye@sxu.edu.cn, guoyaxin_cong@163.com, qpp_sxu@163.com, liru@sxu.edu.cn, zhanghu@sxu.edu.cn

Abstract

Data-driven pre-trained language models typically perform shortcut learning wherein they rely on the spurious correlations between the data and the ground truth. This reliance can undermine the robustness and generalization of the model. To address this issue, data augmentation emerges as a promising solution. By integrating anti-shortcut data to the training set, the models' shortcut-induced biases can be mitigated. However, existing methods encounter three challenges: (1) Manual definition of shortcuts is tailored to particular datasets, restricting generalization. (2) The inherent confirmation bias during model training hampers the effectiveness of data augmentation. (3) Insufficient exploration of the relationship between the model performance and the augmented data quantity may result in excessive data consumption. To tackle these challenges, we propose a method of Smart Data Augmentation based on Large Language Models (SAug-LLM). It leverages the LLMs to autonomously identify shortcuts and generate their anti-shortcut counterparts. In addition, the dual validation is employed to mitigate the confirmation bias during the model retraining. Furthermore, the data augmentation process is optimized to effectively rectify model biases while minimizing data consumption. We validate the effectiveness and generalization of our method through extensive experiments across various natural language processing tasks, demonstrating an average performance improvement of 5.61%.

1 Introduction

Natural language processing (NLP) is indispensable for unlocking the full potential of numerous real-world AI applications. Data-driven pre-trained language models (PLMs) have demonstrated impressive performance in many NLP tasks. However, recent research reveal that the actual prowess of PLMs may not align with initial lofty expectations (Dogra et al., 2024) (Tang et al., 2023). PLMs are susceptible to spurious correlations between data and ground truth, and perform shortcut learning (Sun et al., 2024) (Wang et al., 2023). For example, in Figure 1, the model learns the spurious correlation between interrogative word *when* in the data and the ground truth *September 1876* (Lai et al., 2021), which is called the *interrogative word heuristic shortcut*, highlighting the model's tendency to capture superficial patterns rather than truly understand the underlying context. While PLMs may perform well on the samples aligned with the training distribution, their effectiveness diminishes when applied to diverse samples requiring nuanced inference, calling for more robust models to ensure their effectiveness across a spectrum of real-world scenarios.

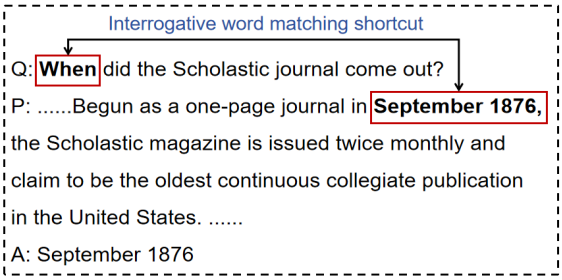


Figure 1: An example of shortcut of the interrogative word heuristic in machine reading comprehension.

Data augmentation is an effective strategy to mitigate a model's shortcut learning (Kumar et al., 2023) (Wen et al., 2022). It typically involves aggregating the anti-shortcut data into the original training set and retraining the models to rectify the prior beliefs or biases induced by shortcuts. For instance, (Jain et al., 2021) proposes a rewriting method to generate gender-balanced data, which enhances decision-making that is not influenced by gender shortcuts. Similarly, (Wen et al., 2022) proposes an automatic counterfactually data genera-

*Corresponding author

Proceedings of the 31st International Conference on Computational Linguistics, pages 8160–8172
January 19–24, 2025.

tion framework for reducing NLU models' reliance on shortcuts or spurious features.

Nevertheless, these studies only target specific datasets and depend on predefined heuristic shortcut strategies devised by humans, resulting in limited generalization. Moreover, they primarily focus on the generation methods of anti-shortcut data, with insufficient attention directed toward the effective utilization of these generated data. During the data utilization process, the model's efficacy is influenced by its inherent confirmation bias—a tendency for both humans and machines to seek or interpret information that aligns with their preexisting beliefs when processing new data (Charness and Dave, 2017). This phenomenon poses a challenge to correct the model's beliefs through the direct addition of anti-shortcut data to the training set. Furthermore, the amount of augmented examples emerges as a critical factor impacting the models' performance and training consumption. The insufficient exploration of the relationship between the quantity of augmented examples and models' performance in these studies can lead to inefficient use of data and resources.

In this paper, we introduce Smart Data Augmentation based on Large Language Models (SAug-LLM), a novel method designed to mitigate models' shortcut learning. Initially, we use large language models (LLMs) to automatically identify shortcuts in the data and generate corresponding anti-shortcut versions. To minimize confirmation bias during retraining, we implement a dual validation strategy. Furthermore, to enhance the efficiency of the data augmentation process, we incorporate specific optimization strategies. This ensures the effective utilization of anti-shortcut data, thereby improving the overall augmentation process. The contributions of this paper are as follows:

(1) We propose a novel SAug-LLM method to mitigate shortcut learning in models. This method has significant improvements in multiple NLP tasks, and is of great importance for ensuring reliable inference in future LLMs. (2) The anti-shortcut data generation method based on LLMs, introduced in this paper, can automatically discover and eliminate more explicit and implicit shortcuts, demonstrating strong usability and generalization. (3) The dual validation strategy effectively identifies confirmation bias in the model training process and mitigates it by re-weights. (4) The SAug-LLM method can achieve or even surpass the performance of traditional data augmentation using less data, realizing efficient data utilization.

2 Related works

Shortcut learning and mitigation. Shortcut learning refers to a phenomenon where models develop decision rules that rely on the *minimum effort principle* (Geirhos et al., 2020). This principle leans on spurious correlations between keywords and labels instead of leveraging semantic clues in the data for task completion. While shortcut learning may not typically impact prediction accuracy, it can undermine the robustness and generalization abilities of the model.

Scholars have explored various methods to mitigate shortcut learning, categorizing them into two main perspectives (Du et al., 2023): (1) Data-centric mitigation methods focus on alleviating shortcut learning by regulating the training data, such as data augmentation (Wen et al., 2022) and re-weighting (Utama et al., 2020). (2) Model-centric mitigation methods aim to address shortcut learning by explicitly regulating the training process of models, such as adversarial training (Chai et al., 2023), multi-task learning mitigation (Tu et al., 2020), product of experts (Sanh et al., 2020), and contrastive learning (Robinson et al., 2021).

This paper addresses shortcut learning from the perspective of data augmentation. Unlike (Wen et al., 2022), we comprehensively investigate the entirety of the data augmentation process, encompassing both data generation and utilization.

Data augmentation. Previous data augmentation methods can be broadly divided into two categories: edit-based (Xie et al., 2020), and generation-based methods (Quteineh et al., 2020). Edit-based methods use discrete operations to modify raw data, such as exchange or deletion, but their simplicity can sometimes compromise the semantic integrity of the original data. On the other hand, generation-based methods, while excelling in fluency, incur higher costs in model pre-training and decoding. Notably, the emergence of powerful LLMs has showcased proficiency in handling general instructions, demonstrating promising performance in data generation tasks (Wang et al., 2022).

Currently, the prevailing focus of data augmentation is primarily on generation processes, with insufficient attention given to subsequent data utilization. To address this gap, (Ren et al., 2021) proposes a reinforcement learning-based automatic augmentation method that autonomously explores

and refines data augmentation strategies. Furthermore, (Lemley et al., 2017) devises a network that learns how to generate augmented data during the training process of a target network, thereby minimizing network loss.

Inspired by (Wang et al., 2022), we propose a new data generation method based on LLMs. *Unlike the aforementioned methods, we use LLMs for anti-shortcut data generation and subsequently explore the impact of model confirmation bias on data augmentation during the data utilization phase.*

Confirmation bias. Confirmation bias refers to the tendency of humans or machines to seek or interpret information aligning with their prior beliefs when processing data (Lefebvre et al., 2022). Confirmation bias has been reported in various fields, such as cognitive psychology (Allakhverdov and Gershkovich, 2010), social psychology, and politics. Recent evidence suggests that scientific practices are also susceptible to various forms of confirmation bias (Talluri et al., 2018) (Austerweil and Griffiths, 2011) shows that confirmation bias in categorization decisions is similar to selective attention mechanisms, tending to acquire new evidence by *overestimating evidence consistent with the decision and underestimating inconsistent evidence*. However, research on this issue is currently limited in the field of machine learning.

Inspired by (Talluri et al., 2018), *we propose a dual verification strategy to alleviate confirmation bias during the data augmentation process.*

3 Methods

As shown in Figure 2, our proposed SAug-LLM consists of two main components: (1) Anti-shortcut data generation based on LLMs, focusing on automatically generating data that prevents shortcut learning. (2) Smart data utilization, aiming to efficiently and intelligently use anti-shortcut data.

3.1 Preliminaries

Given a NLP task with input X and output Y, a model is responsible for learning the mapping function f from input text $x \in X$ to the corresponding target label $y \in Y$. However, when the training data contains many shortcuts, the model often learns incorrect decision functions and patterns.

Data augmentation serves as an implicit regularizer, aiding the model in learning more accurate patterns by expanding more diverse data. The objective function during training can be formalized as:

$$\begin{aligned} ACC(f_{D_{aug}}(x), y) =& \frac{\sum_{(x,y)\in D_{dev}}(f_{D_{aug}}(x) = y)}{|D_{dev}|} \\ D_{aug} =& D_t \cup D_t^{'} \end{aligned} \tag{1}$$

Here, $ACC(\cdot)$ is a function to calculate the accuracy. $|\cdot|$ is used to calculate the size of a set. D_t is the original training set, $D_t^{'}$ is the anti-shortcut version of the training dataset, which is used for data augmentation. D_{dev} is the dev dataset. $f_{D_{aug}}(\cdot)$ is the model trained on the augmented dataset D_{aug}. The objective function aims to train a model on the augmented dataset to achieve maximum model performance ACC.

3.2 Anti-shortcut data generation based on LLMs

We leverage powerful LLMs, including ChatGPT or GPT-4, to generate the anti-shortcut data. The whole process is achieved through interactive prompts and responses. The detailed generation process is as follows. Step 1: We ask the LLMs for *the definition of shortcut learning* that they can exploit to answer questions. Step 2: We require LLMs to analyze the given data samples and identify as many shortcuts as possible within them. Step 3: We direct the LLMs to rephrase the data samples, rendering the identified shortcuts ineffective, based on the analysis in Step 2, while preserving the original labels. The prompt instructions and associated responses of LLMs are detailed in Appendix A, Table 4. In addition, to ensure the reliability of the data generated by the LLMs, we manually verified the generated anti-shortcut data.

Furthermore, introducing overly simple anti-shortcut data may have a negative impact on the prior knowledge or bias correction of the model. Therefore, we leverage the Dataset Cartography tool (Swayamdipta et al., 2020) to categorize anti-shortcut data generated by LLMs into three categories: easy-to-learn, hard-to-learn, and ambiguous. And then we study the impact of data augmentation with different difficulty levels on alleviating model shortcut learning. Detailed information is displayed in the Appendix B, Figure 5.

3.3 Smart data utilization

We improve the traditional data augmentation methods by proposing a novel dual validation strategy which is implemented at every evaluation point

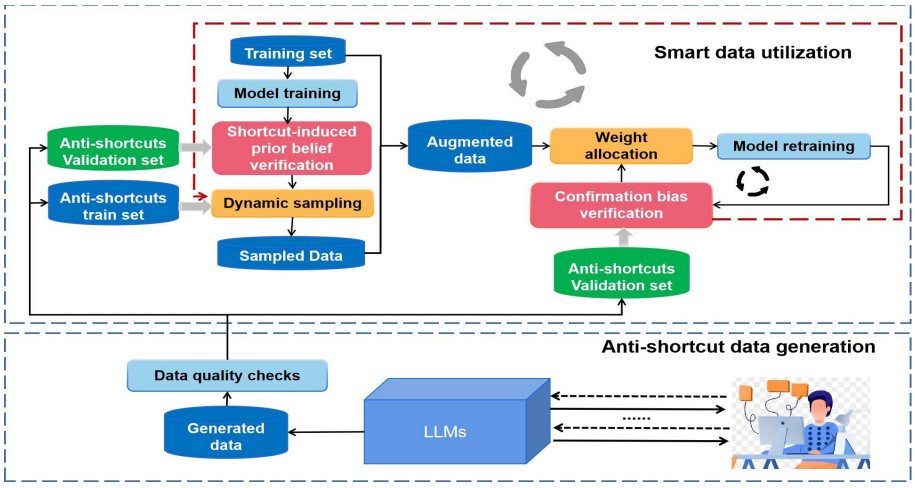


Figure 2: The framework diagram of SAug-LLM.

during training. The strategy includes two parts: shortcut-induced prior belief verification and confirmation bias verification. These components are designed to address the issues of data wastage and confirmation bias in model training.

Shortcut-induced prior belief validation. It primarily verifies the extent of the model's shortcut prior knowledge in the current evaluation period. And dynamic sampling is conducted according to the level of prior knowledge associated with the shortcuts, aiming to reduce data consumption.

The measurement of prior knowledge is reflected through the model's accuracy on the anti-shortcut dev dataset $D^{'}_{dev}$. In i^{th} evaluation point during training, the controller dynamically samples from the anti-shortcut dataset $D^{'}_{t}$, generated by LLMs, based on the degree of prior knowledge of the model, to obtain the sampled data. Specifically, a higher accuracy indicates a lower prior knowledge level of shortcut learning, and therefore fewer samples are sampled, and vice versa, more samples are sampled. The sampling strategy is as Formula 2.

$$
\begin{aligned}
&D_{sample_{i_j}} = Sample_{i_j}(D^{'}_{t}) \\
&|D_{sample_{i_j}}| = \begin{cases} \dfrac{\eta}{ACC(f_{D_{t_i}}(x), y)}|D^{'}_{t}|, j = 0 \\ \dfrac{\eta}{ACC(f_{D_{aug_{i_j}}}(x), y)}|D^{'}_{t}|, j = others \end{cases}
\end{aligned} \tag{2}
$$

where $Sample_{i_j}$ is the j^{th} sampling strategy updated in the i^{th} evaluation cycle, and $D_{sample_{i_j}}$ is the sampled data. D_t is the train set, and D_{t_i} refers to the subset of data that has been trained before the arrival of the i^{th} evaluation point. $f_{D(\cdot)}(\cdot)$ is the model trained in $D_{(\cdot)}$. The hyperparameter η is utilized to govern the scale of data sampling. In this paper, we set the value of η to 0.25 based on experience in the experiment, and the sensitivity test for η can be found in the appendix C.

Confirmation bias validation. It primarily examines the model's susceptibility to confirmation bias following data augmentation. If confirmation bias is detected, the strategy adjusts the model's training weights with respect to both the augmented and the original training data. This adjustment guides the model to focus more on the augmented data, thereby mitigating the confirmation bias.

Measurement of confirmation bias. The evaluation hinges on the premise that if the model, with data augmentation, exhibits confirmation bias, its predictions on the anti-shortcut dev set closely align with those of the model without augmentation. In other words, the accuracy and predicted probability of the model trained on $D_{aug_{i_j}}$ are comparable to those trained on D_{t_i}. We measure the confirmation bias by ACC's difference between the two models, which are trained on different amounts of data set. This evaluation can be quantified as C_{bias} as Equation 3.

$$
\begin{aligned}
C_{bias}(f_{D_{aug_{i_j}}}) = -(&ACC(f_{D_{aug_{i_j}}}(x), y) \\
&-ACC(f_{D_{t_i}}(x), y)), (x, y) \in D^{'}_{dev}
\end{aligned} \tag{3}
$$

$$
D_{aug_{i_j}} = \begin{cases} D_{t_i} \cup D_{sample_{i_j}}, j = 0 \\ D_{aug_{i_j}} \cup D_{sample_{i_j}}, j = others \end{cases} \tag{4}
$$

Mitigation of Confirmation Bias. We adjust the

weights on dataset to mitigate this bias. So, Formula 4 is transformed into Formula 5.

$$D_{aug_{i_{j\varepsilon}}} = \begin{cases} Reweight(D_{t_i}, D_{sample_{i_j}}), j = 0 \\ Reweight(D_{aug_{i_j}}, D_{sample_{i_j}}), others \end{cases} \quad (5)$$

where ε is the hyper-parameters that controls the weight ratio. $D_{aug_{i_{j\varepsilon}}}$ is the augmented dataset after adjusting data weights.

By seeking the optimal weight ratio ε, we *minimize the value* C_{bias}. We employ the particle swarm optimization algorithm (Kennedy and Eberhart, 1995) to optimize the model training process and derive the optimal parameters ε^*. It can be formulated as:

$$\begin{aligned} \varepsilon^* = & argmin(-(ACC(f_{D_{aug_{i_{j\varepsilon}}}}(x), y) \\ & - ACC(f_{D_{t_i}}(x), y))), (x, y) \in D^{'}_{dev} \end{aligned} \quad (6)$$

Then, repeat the operations of Formulas 2-6 until the C_{bias} reaches its minimum after data augmentation in this evaluation cycle. The object function for data augmentation is expressed as:

$$\begin{aligned} & ACC(f_{D_{aug_{i_{j\varepsilon}}}}(x), y) \\ & = \frac{\sum_{(x,y)\in D^{'}_{dev}} (f_{D_{aug_{i_{j\varepsilon}}}}(x) = y)}{|D^{'}_{dev}|} \end{aligned} \quad (7)$$

3.4 Implementation of SAug

The Smart Data Augmentation (SAug) is carried out in an iterative manner. At each evaluation point, the controller first dynamically samples $D_{sample_{i_j}}$ from the anti-shortcuts dataset $D^{'}_t$ based on the degree of current shortcut-induced prior knowledge to synthesize the augmented set $D_{aug_{i_j}}$, and then trains a model $f_{aug_{i_j}}$ based on it. Then, observe whether there is confirmation bias in the model $f_{aug_{i_j}}$. If so, retrain the augmented model by reallocating weights between the D_{t_i} and $D_{sample_{i_j}}$ to continuously improve its performance. This algorithm process is as the Algorithm 1.

Algorithm 1 SAug

Require: D_t; $D^{'}_t$; Number of evaluations: m
Ensure: $f_{i_{jk}}$ // Trained model

```
for each i ∈ [1, m] do
    Train f_{D_{t_i}} in D_{t_i}
    j=0; k=0; f_{i_0}= f_{D_{t_i}} // k is the number of optimizations for ε
    while j=0 or ACC_{i_j} >= ACC_{i_{j-1}} do
        Evaluate f_{i_j} in D'_{dev} → ACC_{i_j} // Shortcuts-induced prior belief validation
        Update Sample_{i_j} based on ACC_{i_j} // Dynamic sampling strategy
        sample the data from D'_t, obtain D_{sample_{i_j}}
        D_{aug_{i_j}} = D_{t_i} ∪ D_{sample_{i_j}}
        j++
        Train f_{i_j} in D_{aug_{i_j}}
        while k = 0 or C_{bias}(f_{i_k}) <= C_{bias}(f_{i_{j(k-1)}}) do
            Calculate and confirm deviation loss C_{bias}(f_{i_k}) // Confirmation bias validation
            Optimize ε_k
            Reweight(D_{aug_{i_j}}, D_{sample_{i_j}})
            k++
            Retrain f_{i_{jk}}
        end while
        Evaluate f_{i_{jk}} in D'_{dev} → ACC_{i_j}
    end while
end for
```

4 Experiments and results

4.1 Experiment setting

Baselines. We have chosen two data augmentation baselines. (1) **Single-Stage**. Single-Stage involves directly integrating augmented data into the training set for retraining, without employing any special procedures during data utilization. This method is commonly used to mitigate shortcut learning. (2) **Text AutoAugment** (**TAA**) (Ren et al., 2021). TAA is a learnable combinatorial data augmentation paradigm. The goal is to automatically learn the optimal editing-based data augmentation strategy. The accuracy of TAA on the task of low resource classification imbalance is due to the strong baseline and reaches SOTA. We only choose the data utilization strategy in TAA as the baseline for comparison.

In addition, we select three PLMs, BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and XLnet (Yang et al., 2019), as the base models for studying the applicability of the SAug-LLM. We selected these models based on the following considerations: (1) These models are smaller in scale and have a wider range of potential applications in the real world. (2) These models are more likely to perform fast learning because their training datasets are smaller, and these models can better reflect the effectiveness of our method.

Methods			C3		Dream		SNLI		MNLI	
Test set			ori	chall	ori	chall	ori	chall	ori	chall
BERT	None		62.29	56.58	60.32	51.22	82.44	53.91	72.26	52.29
	Single-Stage	Edi	62.73	57.25	60.63	53.23	82.19	61.53	72.86	55.02
		GAN	63.59	57.36	61.28	54.29	82.61	61.89	72.09	56.35
		ChatGPT	63.55	57.98	62.64	54.75	80.77	61.24	72.75	56.91
		GPT-4	64.11	58.16	62.28	55.60	82.65	69.87	72.57	59.67
	TAA	Edi	62.92	57.84	60.96	56.06	82.95	61.98	72.63	57.12
		GAN	63.02	57.90	60.95	56.72	82.37	62.03	72.91	58.75
		ChatGPT	63.84	58.03	61.85	57.32	82.65	62.87	72.07	59.12
		GPT-4	63.85	58.57	62.07	57.86	83.02	63.06	72.11	59.06
	SAug	Edi	63.24	58.37	60.99	57.36	83.01	62.99	72.29	58.99
		GAN	63.38	58.86	57.25	57.69	83.10	63.03	72.96	58.11
		ChatGPT	65.08	59.52	62.86	58.53	81.77	59.56	73.08	58.85
		GPT-4	**65.52**	**60.56**	**63.74**	**58.79**	**83.10**	**70.19**	**74.17**	**59.98**

Table 1: Performance of BERT models employing various data augmentation techniques, evaluated on the test set and its anti-shortcut version, designated as the original set (ori) and the challenge set (chall), respectively.

Datasets. We validate the effectiveness and generalization of the proposed method on two typical NLP tasks of the MRC and NLI. We conduct experiments and analysis on the dataset of the Dream, C3, MNLI, and SNLI. The dataset size used in this article is shown in Appendix C, Table 6.

(1) C3 (Sun et al., 2020) is a chinese MRC dataset derived from the Chinese Second Language Test and in a similar form to the Dream dataset. (2) Dream (Sun et al., 2019) is a challenging multiple-choice English reading comprehension dataset in which 85% of the questions require reasoning beyond a sentence. (3) SNLI (Bowman et al., 2015) is an NLI dataset developed by Stanford University, Which is a collection of hand-written pairs of English sentences that support NLI's tasks. (4) MNLI (Nangia et al., 2017) is a multi-type NLI dataset with 3 classification tasks. This dataset is widely used in NLI tasks together with the SNLI dataset.

Implementation details. The search range of the ε in Formula 6 in Section 3.3 is set as [-2.0, 2.0]. Each training iteration comprises 10 epochs, employing the Adam optimizer with a learning rate of 2e-05. Logging-steps and gradient-accumulation-steps are set as 200 and 5, respectively.

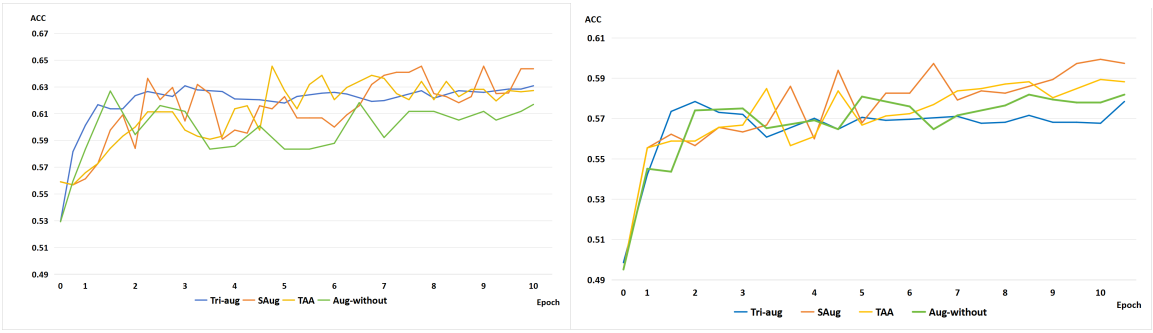
4.2 Main experimental results

Table 1 demonstrates that our proposed method exhibits an average improvement of 2.31% on the original test set and 8.88% on the challenge set (anti-shortcut version), compared to the BERT model that utilizes Single-Stage. These results affirm the effectiveness of our approach in mitigating shortcut learning.

4.2.1 The effectiveness of data generation

We employ three distinct approaches: Edit-based (Edi), GAN-based(GAN), and LLM-based (ChatGPT and GPT-4), to generate anti-shortcut data and observe their effectiveness in alleviating model shortcut learning. Table 1 illustrates that, regardless of using SAug, Single-Stage, or TAA methods for data augmentation, our proposed LLM-based method achieves superior performance. This superiority may stem from the LLM-based method's ability to identify and eliminate both explicit and implicit shortcuts more effectively than the human-based heuristic shortcuts used in Edit-based and GAN-based methods. The shortcut discovery ability of the LLM model is shown in Appendix A, Table 4. In addition, we noticed that the effectiveness of the Edit-based method is not as good as that of the GAN-based method, which may be due to the lower smoothness of the data generated by the Edit-based method, which affects model learning.

We conduct a comparative analysis of anti-shortcut data generation methods using ChatGPT and GPT-4, two popular LLMs models. Table 1 indicates that the data augmentation effectiveness of the anti-shortcut samples generated by GPT-4 exceeds that of ChatGPT. This result supports our conclusion that GPT-4 is good at identifying shortcuts, rewriting and generating data, and following instructions, which has been confirmed through manual data quality assessment.



(a) C3 dataset (b) Dream dataset

Figure 3: Variation in accuracy (ACC) on the validation set across different augmentation strategies during the training process, with each strategy using the same number of augmented samples in training.

4.2.2 The effectiveness of SAug

Table 1 shows that when comparing Single-Stage and TAA, the model's performance notably enhances with the SAug on both the original and anti-shortcut test sets. This finding remains consistent across various models and augmented data, validating the efficacy of a dual validation strategy to mitigate confirmation bias.

4.2.3 Results on various PLMs using SAug-LLM

The results from Table 1 , Table 7 and Table 8 in Appendix D demonstrate that after incorporating SAug-LLM, BERT, RoBERTa, and XLNet exhibit average improvements of 2.31%, 1.71%, and 2.13% on the original test set, and 8.88%, 7.98%, and 10.66% on the anti-shortcut version of the test set.

It can be observed that the performance of the three types of models has significantly improved on the test set, especially on the anti-shortcut version of the test set. This also verifies that the method proposed in this paper has the same applicability to different training language models.

4.3 The change of confirmation bias

To assess the confirmation bias of the model, we compute the accuracy on the dev set throughout the training process, as shown in Figure 3. Our findings indicate that compared to models without data augmentation, Single-Stage, TAA, and SAug significantly enhance the dev set accuracy during training.

Upon analyzing the trajectory of Single-Stage's transformations, a notable pattern emerges: initially, the model's accuracy experiences an upward surge before settling into a steady state. This pattern indicates that data augmentation methods can alleviate the model's shortcut learning ability and improve model performance. However, the accuracy quickly stabilized, indicating that the model established a relatively stable insight that could not be corrected through subsequent data, i.e. confirmation bias. Conversely, with our proposed SAug method, the model's accuracy exhibits frequent fluctuations within the development set, indicating a continual evolution of its prior knowledge. In addition, although TAA continuously adjusts its previous beliefs, its performance is not as good as SAug's due to the lack of the confirmation bias mitigation strategy proposed in this paper.

4.4 Data utilization costs

This study emphasizes the efficient utilization of data. We compare Single-Stage methods with our proposed SAug method and find: (1) Adding more data isn't always better. (2) The SAug method is significantly better than Single-Stage, as observed in Figure 4. In particular, Single-Stage hits its peak performance with 480 data units. In contrast, SAug reaches comparable performance with just around 200 data units. Furthermore, as data volume increases, SAug demonstrates even greater performance improvements compared to Single-Stage.

4.5 The influence of generated data quality on model performance

To further enhance model performance, we delve into the impact of data quality generated by LLMs — specifically, the difficulty level of machine-generated responses — on mitigating shortcut learning. Illustrated in Appendix B, Figure 5, we employ the Dataset Cartography tool to elucidate the challenges in generating anti-shortcut data for LLMs across the C3 and Dream training sets.

Method	C3		Dream		SNLI		MNLI	
test set	ori	chall	ori	chall	ori	chall	ori	chall
all	64.39	59.85	63.04	58.37	82.95	69.32	73.49	59.23
easy-to-learn	63.04	58.21	61.23	57.48	82.66	68.17	73.68	58.42
ambiguous	63.24	59.36	61.95	58.23	82.45	68.52	72.83	59.01
hard-to-learn	65.52	60.56	63.74	58.79	83.10	70.19	74.17	59.98

Table 2: Performance of BERT models enhanced by SAug-LLM, utilizing augmented data of varying quality.

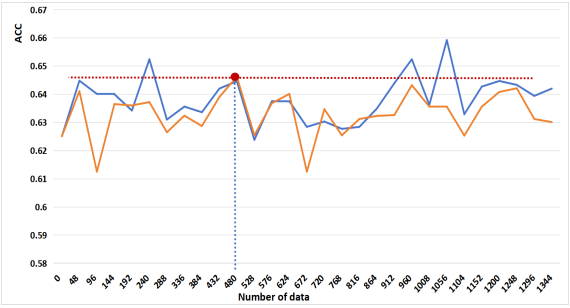


Figure 4: Performance of BERT models employing SAug-LLM method and Single-Stage with varying quantities of augmented data, where the blue line indicates SAug and the yellow line denotes Single-Stage. And the red dots represent the extreme points of Single-Stage.

gradient-accumulation-steps	C3	Dream	SNLI	MNLI
8	64.89	62.08	81.47	73.77
7	64.11	63.18	82.63	73.57
6	65.47	63.38	82.76	72.98
5	**65.52**	**63.74**	83.10	**74.17**
4	62.28	60.77	**83.25**	71.94
3	63.78	62.30	82.25	72.71
2	62.16	60.65	82.89	71.95
1	62.67	60.88	81.25	71.96

Table 3: Performance of BERT models using SAug-LLM at varying gradient-accumulation-steps.

It's apparent that while the examples generated by LLMs have no shortcuts, some are overly simplistic and may not sufficiently challenge the model's prior knowledge. Through diversifying the dataset with various difficulty levels, our investigation reveals that integrating more challenging data types results in a more pronounced correction of the model's prior cognition. Conversely, simpler data exerts a lesser impact on the model's adjustment, as detailed in Table 2.

4.6 The impact of the number of evaluations during training on model performance

We conducted an in-depth study on the impact of the number of evaluations m performed during the training process on model performance. m is influenced by two parameters: gradient-accumulation-steps and logging-steps. Here we control the logging-steps to remain unchanged and change the gradient-accumulation-steps. The larger the number of gradient-accumulation-steps, the smaller the number of evaluations m.

As shown in Table 3, our analysis reveals a trend that as the gradient-accumulation-steps decreases, the number of evaluations increases. Therefore, we use SAug-LLM to adjust the model's prior knowledge more frequently. But experiments have shown that it is not our inherent belief that the more adjustments we make, the better. Overall, when gradient-accumulation-steps is 5, the model performance reaches a good value. As the evaluation frequency continues to decrease, we observe a subsequent decline in model performance.

5 Conclusions

In addressing the challenge of shortcut learning in models, this study focuses on three key issues within data augmentation: (1) Poor generalization of anti-shortcut data generation strategies. (2) The presence of model confirmation bias, hinders correction of shortcut-induced prior knowledge. (3) Low efficiency in data utilization. To tackle these issues, we propose an intelligent augmentation strategy, SAug-LLM. This approach initially leverages LLMs to generate anti-shortcut data and ensures the quality of examples through the Dataset Cartography tool. Furthermore, we introduce a dual verification mechanism to mitigate confirmation bias and optimize the training process for efficient data utilization. Finally, we validate the effectiveness of our proposed method across two typical NLI tasks.

6 Limitations

Currently, our work utilizes the LLMs to generate data samples to alleviate shortcut learning. However, although data augmentation methods have some effectiveness in alleviating shortcut learning, their interpretability is relatively poor. Therefore, we also hope to find some more interpretable methods, such as data-centric rationale explanation (Zhao et al., 2024; Liu et al., 2023) and knowledge enhancement methods (Yan et al., 2024), etc. Additionally, we will further investigate the application potential of the method proposed in this paper in fields with extremely high-reliability demands, such as healthcare and finance, to enhance language models' reliability in real-world application scenarios.

Acknowledgements

We thank all the anonymous reviewers for their constructive comments and suggestions. This work is supported by the National Science and Technology Major Project (2020AAA0106102), the National Natural Science Foundation of China (62076155).

References

Victor M Allakhverdov and Valeria A Gershkovich. 2010. Does consciousness exist?—in what sense? *Integrative Psychological and Behavioral Science*, 44(4):340–347.

Joseph L Austerweil and Thomas L Griffiths. 2011. Seeking confirmation is rational for deterministic hypotheses. *Cognitive Science*, 35(3):499–526.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Yidong Chai, Ruicheng Liang, Sagar Samtani, Hongyi Zhu, Meng Wang, Yezheng Liu, and Yuanchun Jiang. 2023. Additive feature attribution explainable methods to craft adversarial attacks for text classification and text regression. *IEEE Transactions on Knowledge and Data Engineering*.

Gary Charness and Chetan Dave. 2017. Confirmation bias with motivated beliefs. *Games and Economic Behavior*, 104:1–23.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Varun Dogra, Sahil Verma, Kavita, Marcin Woźniak, Jana Shafi, and Muhammad Fazal Ijaz. 2024. Shortcut learning explanations for deep natural language processing: A survey on dataset biases. *IEEE Access*, 12:26183–26195.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Communications of the ACM (CACM)*.

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

Nishtha Jain, Maja Popovic, Declan Groves, and Eva Vanmassenhove. 2021. Generating gender augmented data for nlp. *arXiv preprint arXiv:2107.05987*.

J. Kennedy and R. Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4.

Abhinav Kumar, Amit Deshpande, and Amit Sharma. 2023. Causal effect regularization: Automated detection and removal of spurious correlations. In *Advances in Neural Information Processing Systems*, volume 36, pages 20942–20984. Curran Associates, Inc.

Yuxuan Lai, Chen Zhang, Yansong Feng, Quzhe Huang, and Dongyan Zhao. 2021. Why machine reading comprehension models learn shortcuts? *arXiv preprint arXiv:2106.01024*.

Germain Lefebvre, Christopher Summerfield, and Rafal Bogacz. 2022. A normative account of confirmation bias during reinforcement learning. *Neural Computation*, 34(2):307–337.

Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. 2017. Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, 5:5858–5869.

Wei Liu, Haozhao Wang, Jun Wang, Ruixuan Li, Xinyang Li, YuanKai Zhang, and Yang Qiu. 2023. MGR: Multi-generator based rationalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 12771–12787, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R. Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. *Preprint*, arXiv:1707.08172.

Husam Quteineh, Spyridon Samothrakis, and Richard Sutcliffe. 2020. Textual data augmentation for efficient active learning on tiny datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7400–7410.

Shuhuai Ren, Jinchao Zhang, Lei Li, Xu Sun, and Jie Zhou. 2021. Text autoaugment: Learning compositional augmentation policy for text classification. *arXiv preprint arXiv:2109.00523*.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. 2021. Can contrastive learning avoid shortcut solutions? *Advances in neural information processing systems*, 34:4974–4986.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2020. Learning from others' mistakes: Avoiding dataset biases without modeling them. *arXiv preprint arXiv:2012.01300*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. Dream: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2020. Investigating prior knowledge for challenging chinese machine reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:141–155.

Zechen Sun, Yisheng Xiao, Juntao Li, Yixin Ji, Wenliang Chen, and Min Zhang. 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)*, pages 6883–6893, Torino, Italia. ELRA and ICCL.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *CoRR*, abs/2009.10795.

Bharath Chandra Talluri, Anne E Urai, Konstantinos Tsetsos, Marius Usher, and Tobias H Donner. 2018. Confirmation bias through selective overweighting of choice-consistent evidence. *Current Biology*, 28(19):3128–3135.

Ruixiang Tang, Dehan Kong, Lo li Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. In *Annual Meeting of the Association for Computational Linguistics*.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards debiasing nlu models from unknown biases. *arXiv preprint arXiv:2009.12303*.

Shunxin Wang, Christoph Brune, Raymond Veldhuis, and Nicola Strisciuglio. 2023. Dfm-x: Augmentation by leveraging prior knowledge of shortcut learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 129–138.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Jiaxin Wen, Yeshuang Zhu, Jinchao Zhang, Jie Zhou, and Minlie Huang. 2022. Autocad: Automatically generating counterfactuals for mitigating shortcut learning. *Preprint*, arXiv:2211.16202.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268.

Zhichao Yan, Jiapu Wang, Jiaoyan Chen, Xiaoli Li, Ru Li, and Jeff Z. Pan. 2024. Atomic fact decomposition helps attributed question answering. *Preprint*, arXiv:2410.16708.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yunxiao Zhao, Zhiqiang Wang, Xiaoli Li, Jiye Liang, and Ru Li. 2024. AGR: Reinforced causal agent-guided self-explaining rationalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 510–518, Bangkok, Thailand. Association for Computational Linguistics.

A Prompt instructions and responses when generating data using LLMs

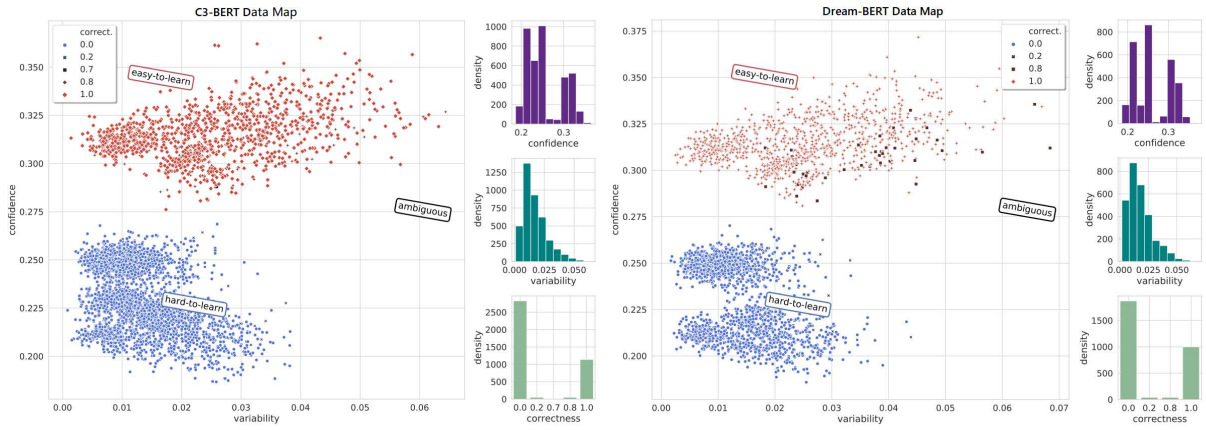
Table 4 outlines the process of prompts and responses for generating anti-shortcut data samples utilizing LLM models. It demonstrates that LLMs are capable of identifying a broader range of shortcuts, potentially surpassing human-defined shortcut limitations. This also indicates that LLMs have a better ability to detect and mitigate shortcuts in data.

B Data difficulty classification using the Dataset Cartography tool

As shown in Figure 5, we use the Dataset Cartography tool to demonstrate the difficulty level of

instructions	Responses
Provide a definition of a shortcut to answering questions	A "shortcut to answering questions" refers to a quick and efficient method or strategy used to find the solution or response to a query. In various contexts, a shortcut might involve identifying key words, phrases, or patterns that directly lead to the answer, exploiting specific linguistic or contextual cues, or utilizing pre-existing knowledge to streamline the question-answering process. The goal of employing shortcuts is to expedite the response time and enhance efficiency in obtaining accurate answers, particularly in situations where direct correspondences between the question and answer elements exist.
Here is a question for you, please try to discover as many shortcuts as possible in order to answer the question faster. PASSAGE: Begun as a one-page journal in September 1876, the Scholastic magazine is issued twice monthly and claim to be the oldest continuous collegiate publication in the United States. Question: when did the Scholastic journal come out? Option: A:September 1876 B:September 1877 C:September 1976 Answer: September 1876	Let's identify some shortcuts: Focus on Keywords: Look for keywords in the passage that directly relate to the answer. In this case, the passage mentions that the Scholastic magazine began in "September 1876," so you can quickly recognize option A as the potential answer. Eliminate Distractors: Disregard options that contain years other than 1876. In this question, options B and C can be quickly eliminated because they have different years. Use Context Clues: Consider the context of the sentence. The passage mentions that the Scholastic magazine began in September 1876, providing a clear context for the answer. By applying these shortcuts, you can confidently and quickly select option A: September 1876 as the correct answer.
Please rewrite the question based on the shortcuts found above to eliminate them, and be careful not to change the correctness of the answer.	The Scholastic magazine, asserting itself as the oldest continuous collegiate publication in the United States, started in September 1876 as a one-page journal. The Academic magazine was founded in September 1877. When was the Scholastic magazine first issued? Option: A. September 1876 B. September 1877 C. September 1976 Answer: September 1876

Table 4: An examples of instructions and LLM's responses (GPT-4) for generating the anti-shortcut data.



(a) C3 dataset (b) Dream dataset

Figure 5: Difficulty analysis of anti-shortcut data generated by GPT-4

η	C3	Dream	SNLI	MNLI
0.1	63.40	62.76	80.05	70.26
0.15	63.85	62.64	81.47	71.99
0.2	62.56	61.43	82.63	72.13
0.25	**65.52**	**63.74**	**83.10**	**74.17**
0.3	64.37	62.53	82.65	71.42
0.35	62.81	62.97	82.11	69.82

Table 5: The impact of hyper-parameter η on model performance on Bert using SAug-LLM(GPT-4).

the anti-shortcut data generated by LLMs for the BERT model. In this visualization, the blue dots in the bottom left corner represent examples that pose significant challenges to model learning, while the red dots in the top left corner represent data that is relatively easier to learn. Additionally, the presence of black dots signifies instances of data ambiguity.

C The impact of η on model performance

The hyperparameter η is utilized to govern the scale of data sampling. As shown in Table 5, we conducted sensitivity experiments on η and ultimately set it to 0.25.

D Dataset size

Table 6 presents the dataset sizes. Given the necessity to employ LLMs for generating anti-shortcut samples for the training set, the original dataset sizes for MNLI and SNLI necessitate considerable computational resources. Consequently, a subset of their data has been selected for use.

Dataset	C3	Dream	MNLI	SNLI
train	4884	3868	13090	18338
dev	1627	1097	1000	1000
test	1542	982	1000	1000

Table 6: The dataset size used in this paper.

E Results on RoBERTa and XLnet using SAug-LLM

Tables 7 and 8 depict the performance enhancements of the SAug-LLM technique on the RoBERTa and XLnet models, respectively. Notably, both models exhibit marked improvements across the test set and the non-shortcut variant of the test dataset, mirroring the improvements seen with BERT. This underscores the method's consistent applicability across various pre-trained language models.

Methods			C3		Dream		SNLI		MNLI	
Test set			ori	chall	ori	chall	ori	chall	ori	chall
RoBERTa	None		65.50	63.21	69.56	58.85	86.92	52.53	83.45	61.17
	Single-Stage	Edi	66.23	63.57	69.12	62.56	87.32	63.43	83.36	63.03
		GAN	67.04	64.21	69.65	63.17	87.27	64.15	84.20	64.94
		ChatGPT	66.98	64.76	70.00	63.24	85.92	61.94	84.63	64.68
		GPT-4	67.52	65.52	69.78	63.64	87.21	68.08	84.85	**66.66**
	TAA	Edi	67.35	63.84	69.96	64.06	86.96	61.98	83.75	64.16
		GAN	67.02	63.90	69.95	63.72	86.38	62.03	84.10	64.75
		ChatGPT	67.54	64.03	69.85	64.32	86.66	62.87	84.76	64.12
		GPT-4	68.50	65.57	69.97	64.86	87.03	68.06	84.30	65.06
	SAug	Edi	66.91	63.82	69.99	64.36	87.18	65.65	84.14	65.15
		GAN	67.38	64.52	69.25	64.69	87.13	66.24	83.33	65.43
		ChatGPT	68.32	65.35	69.86	64.53	86.32	62.04	84.04	65.49
		GPT-4	**68.92**	**65.96**	**70.77**	**65.38**	**87.43**	**70.07**	**85.15**	66.26

Table 7: Performance of RoBERTa models employing various data augmentation techniques, evaluated on the test set and its anti-shortcut version.

Methods			C3		Dream		SNLI		MNLI	
Test set			ori	chall	ori	chall	ori	chall	ori	chall
XLnet	None		60.60	56.26	62.74	55.09	84.83	45.89	80.95	58.38
	Single-Stage	Edi	60.85	56.83	64.52	60.85	83.55	66.21	81.75	62.26
		GAN	60.56	57.17	64.99	61.23	83.54	67.42	81.65	62.95
		ChatGPT	61.31	57.62	66.29	64.05	84.24	68.08	81.96	63.54
		GPT-4	62.71	58.02	66.79	64.85	84.34	68.30	82.46	63.37
	TAA	Edi	61.92	57.87	64.96	61.06	83.43	66.98	81.32	63.12
		GAN	61.02	57.92	64.95	62.72	83.85	67.03	81.60	63.75
		ChatGPT	62.84	58.13	64.85	63.32	83.13	67.87	82.16	63.12
		GPT-4	63.20	58.77	66.07	63.86	84.20	68.06	82.30	63.06
	SAug	Edi	61.23	57.89	65.24	61.39	84.16	67.07	81.45	62.96
		GAN	62.35	58.12	65.36	62.36	84.63	67.94	81.61	62.78
		ChatGPT	63.14	59.44	65.98	64.54	**84.84**	68.66	81.85	63.47
		GPT-4	**63.75**	**60.02**	**66.88**	**64.89**	84.25	**69.61**	**82.76**	**63.77**

Table 8: Performance of XLnet models employing various data augmentation techniques, evaluated on the test set and its anti-shortcut counterparts.