SALAD: Improving Robustness and Generalization through Contrastive Learning with Structure-Aware and LLM-Driven Augmented Data

Suyoung Bae[1*], **Hyojun Kim**[2*†], **YunSeok Choi**[1‡], **Jee-Hyong Lee**[1‡]
[1] Sungkyunkwan University, South Korea
[2] SK Telecom, South Korea
[1] {sybae01, ys.choi, john}@skku.edu, [2] hjkim@sk.com

Abstract

In various natural language processing (NLP) tasks, fine-tuning Pre-trained Language Models (PLMs) often leads to the issue of spurious correlations, which negatively impacts performance, particularly when dealing with out-of-distribution data. To address this problem, we propose **SALAD** (**S**tructure **A**ware and **L**LM-driven **A**ugmented **D**ata), a novel approach designed to enhance model robustness and generalization by generating structure-aware and counterfactually augmented data for contrastive learning. Our method leverages a tagging-based approach to generate structure-aware positive samples and utilizes large language models (LLMs) to generate counterfactual negative samples with diverse sentence patterns. By applying contrastive learning, *SALAD* enables the model to focus on learning the structural relationships between key sentence components while minimizing reliance on spurious correlations. We validate our approach through experiments on three tasks: Sentiment Classification, Sexism Detection, and Natural Language Inference. The results demonstrate that *SALAD* not only improves model robustness and performance across different environments but also enhances generalization to out-of-distribution datasets and cross-domain scenarios.

1 Introduction

In many natural language processing (NLP) tasks, machine learning models often suffer from the issue of spurious correlations (a.k.a shortcuts) between input text tokens and output labels (Tu et al., 2020). These shortcuts allow models to rely on irrelevant patterns in the data, leading to biased predictions. For example, a model trained mainly on positive reviews of *Spielberg* movies may incorrectly associate the word *Spielberg* with favorable sentiment, regardless of the actual content of the review (Wang and Culotta, 2020; Wang et al., 2022). This reliance on superficial patterns results in poor performance, especially when handling out-of-distribution data.

Many research has addressed this problem by exploring methods for generating counterfactually augmented data (CAD) (Kaushik et al., 2020; Samory et al., 2021), intended to disrupt these false correlations. The CAD method involves altering input data to flip its label, providing models with examples that help overcome shortcut-based learning and improve generalization across diverse datasets. However, previous approaches have focused mainly on manual generation methods and statistical techniques to automate CAD creation. Although manual methods can produce high-quality counterfactual data, they are both costly and time-consuming.

To reduce these costs and improve scalability, automated techniques have been developed, such as using sentiment dictionaries (Yang et al., 2021), statistical matching, or predefined antonyms (Wang and Culotta, 2021). However, these methods are dependent on fixed rules that restrict the quality of the generated data. For example, sentiment dictionaries and statistical techniques struggle to fully capture the complex context within the data, limiting their ability to produce diverse patterns that can enhance model performance.

Recently, there has been a shift towards generating CAD using pre-trained language models (PLMs) (Madaan et al., 2021; Wu et al., 2021; Zhou et al., 2022; Wen et al., 2022; Dixit et al., 2022; Liu et al., 2022; Chen et al., 2023). These models offer the advantage of automatically producing high-quality data by reflecting the context of the input data, thereby reducing the reliance on manual methods. However, since PLMs are trained based on the distribution of the training data, they can easily become biased toward frequently occurring contextual patterns. This causes PLMs to rely more

[*] These authors contributed equally to this work.
[†] Work was conducted during his graduate studies at Sungkyunkwan University.
[‡] Co-corresponding authors.

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), pages 12724–12738

on common patterns, rather than capturing complex sentence structures or uncovering hidden causal relationships. Moreover, most current approaches focus solely on data augmentation, neglecting the need to enhance model robustness during training.

To improve model robustness while reducing shortcuts, it is necessary not only to focus on contextual patterns but also to learn structural patterns where shortcuts occur. Models that learn these structural patterns can capture the fundamental meaning of sentences and the relationships between key components, rather than relying on word frequency or specific terms. By doing so, the model effectively avoids shortcuts and achieves stronger generalization. Structural patterns, particularly those involving the roles of subjects, verbs, and objects, remain consistent across different contexts, allowing models to perform more reliably across diverse datasets.

Therefore, the model should learn the causal relationships within sentences to effectively learn structural patterns, rather than just word frequency or simple textual features. This allows the model to focus on important words (nouns, verbs, adjectives, etc.) to grasp the core meaning of the sentence while reducing its reliance on less significant words (pronouns, conjunctions, etc.). Consequently, models that learn structural patterns can provide consistent performance across a variety of sentence structures.

In this paper, we propose an effective method, **SALAD**, for enhancing model robustness while addressing spurious correlations, called **S**tructure **A**ware and **L**LM-driven **A**ugmented **D**ata for Contrastive Learning. This method combines *structure-aware augmented data* (positive) generated by a tagging-based approach that considers structural patterns and *counterfactually augmented data* (negative) generated by large language models (LLMs) to create complex and diverse sentence patterns. Then, contrastive learning is applied to enable the model not only to classify patterns but also to capture relationships between samples. Using the original sentence as the anchor, the positive sample is trained to remain close to it while maintaining the structural patterns where shortcuts occur, whereas the negative sample, generated by the LLM, is trained to diverge from the anchor by incorporating diverse patterns. Our *SALAD* helps the model effectively learn key patterns in simple sentence structures, reducing its dependence on shortcuts. Simultaneously, the model learns negative patterns in various contexts, enhancing its generalization performance. We conducted experiments on three tasks: Sentiment Classification, Sexism, and Natural Language Inference, to verify the model's robustness across various environments. Furthermore, we demonstrated our model's generalization performance in out-of-distribution datasets and cross-domain settings.

2 Related work

Text Data Augmentation Data augmentation has traditionally been used to improve model performance and increase data diversity. EDA (Wei and Zou, 2019) applied simple heuristic transformations such as synonym replacement, word insertion, deletion, and swapping. While it was easy to implement and low-cost, it has the limitation of not considering context. PLM-based methods, such as SSMBA (Ng et al., 2020), used pre-trained models like BERT (Devlin et al., 2019) to corrupt and reconstruct data, improving performance, particularly on out-of-domain datasets. However, this approach requires significant computational resources and could introduce noise. LLM-based methods, like AugGPT (Dai et al., 2023), leveraged GPT-3 to rephrase sentences and enhance performance in few-shot learning settings, but they relied heavily on large language models, which were resource-intensive. While these methods contributed to increased data diversity, they had the drawback of lacking fine-grained control over the quality and relevance of the generated data.

Counterfactual Data Augmentation Counterfactual augmented data has been generated using various approaches. Early methods relied heavily on manual annotation, where human annotators made minimal changes to flip the label of the original text (Kaushik et al., 2020; Samory et al., 2021). This manual approach was effective in improving the robustness and generalization of text classification models but was also time-consuming and costly. To address the limitations of manual annotation, rule-based methods were introduced. These methods employed fixed rules such as sentiment dictionaries (Yang et al., 2021) or named-entity tags, semantic role labels, and sentiment information (Madaan et al., 2021) to automatically generate CAD. However, these approaches were restricted by the rigid nature of predefined rules, limiting the quality and flexibility of the generated data. More recently, LLMs have been explored for CAD gener-

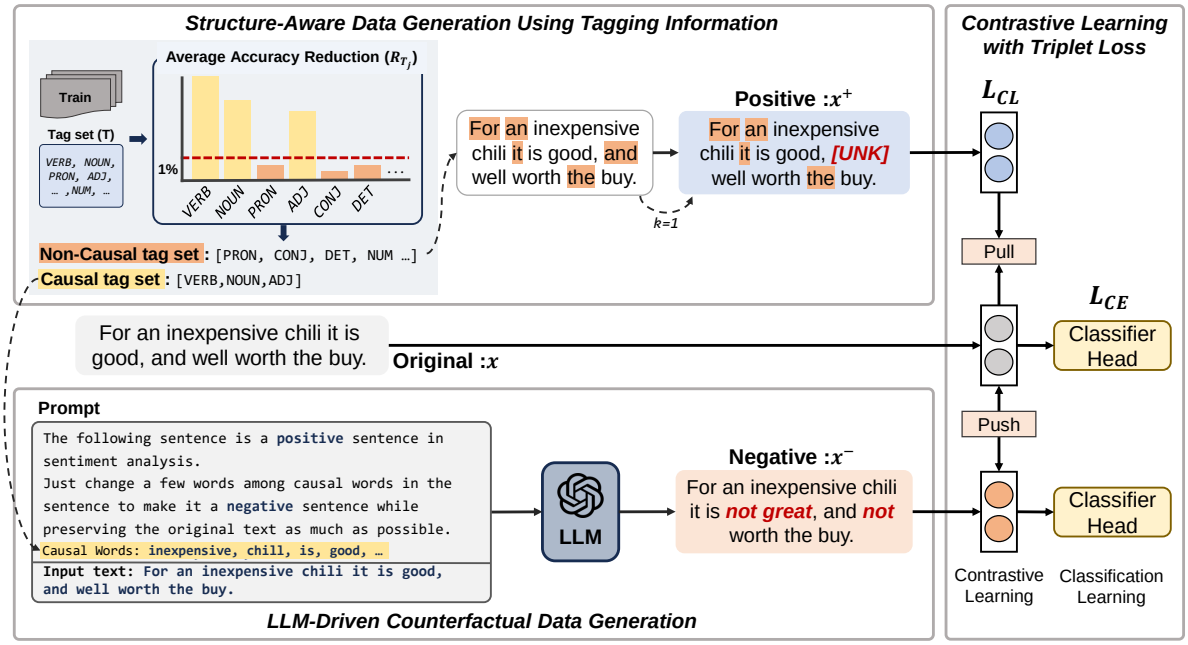

Figure 1: **Overview of *SALAD*.** Our proposed method consists of three steps. First, we use a tagging-based method to generate positive data based on the structure where shortcuts occur (Sec. 3.1). Next, we use an LLM to generate counterfactual data to capture complex and diverse sentence patterns (Sec. 3.2). Finally, contrastive learning is applied to effectively capture key sentence structural patterns between our augmented data, minimizing spurious correlations and enhancing generalization performance (Sec. 3.3).

ation. For example, GPT-3 has been combined with counterfactual retrievers to automatically produce CAD (Dixit et al., 2022), while other frameworks have utilized collaboration between human workers and LLMs to create effective datasets (Liu et al., 2022). Additionally, GPT-3 has been employed to generate high-quality data for natural language inference (NLI) tasks (Chen et al., 2023). Despite these advancements, many LLM-based methods still require human validation or the use of additional models for data verification. Furthermore, the majority of these studies have focused on data augmentation without addressing practical training strategies aimed at improving model robustness.

Robust Learning Recent methods for improving robustness in text classification have explored a range of strategies. One approach combines cross-entropy loss with SupCon loss (Khosla et al., 2020) which has shown improvements in general performance and robustness but does not fully address spurious correlations (Gunel et al., 2021). To tackle the spurious correlation problem, some methods focus on identifying and removing shortcut-related features. For example, matched sample techniques have been used to distinguish between shortcuts and genuine patterns, improving robustness by filtering out shortcut-related words (Wang and Culotta, 2020). Further methods include using cross-domain analysis and knowledge-aware perturbations to differentiate spurious tokens from important ones (Wang et al., 2022). Another approach is causally contrastive learning, which trains models to identify causal features, improving their robustness against spurious correlations (Choi et al., 2022). Despite these advancements, many of these methods still rely on gradient-based techniques and fine-tuned classifiers, which can be biased themselves, limiting their effectiveness in fully overcoming spurious correlations.

3 Proposed Method

Figure 1 illustrates an overview of *SALAD*, consisting of the following three processes: Structure-aware data generation, LLM-driven counterfactual data generation, and contrastive learning with triplet loss.

3.1 Structure-Aware Data Generation Using Tagging Information

We propose a method to construct positive data that reduces bias from non-causal words and spurious correlations, enhancing model robustness and generalization. To address these issues, it is crucial that the model learns structural patterns where shortcuts

occur by focusing on understanding the causal relationships within sentences, rather than relying on word frequency or simple textual features. Therefore, we leverage the Part-of-Speech (POS) (Petrov et al., 2012) tag set to identify structural information where shortcuts are likely to occur. Then, we construct a non-causal POS tag set G.

Given a collection of training data $D = \{x_i\}_{i=1}^m$ and the universal POS tag set $T = \{VERB, NOUN, \ldots, DET\}$, we remove all tokens corresponding to each POS tag T_j. For example, by removing all words corresponding to *VERB* from the data, we construct $D_{\backslash VERB} = \{o_i\}_{i=1}^m$. We then calculate the average accuracy reduction from the standard fine-tuned model f to determine which tags are irrelevant to the label.

$$R_{T_j} = \frac{1}{m}\sum_{i=1}^{m}(f(x_i) - f(o_i)) \quad (1)$$

If the average accuracy reduction R_{T_j} exceeds a *threshold*, we consider that T_j is a causal POS tag, meaning it is directly associated with the label. Conversely, if the reduction is smaller than the *threshold*, we assume that T_j belongs to the non-causal POS tag set, indicating it does not affect the label.

After defining the non-causal POS tag set G, we randomly select k tokens from each sentence $\{x_i\}$ that belong to G and replace these tokens with the `[UNK]` token, constructing a set of structure-aware positive data, denoted as $D_{pos} = \{x_i^+\}_{i=1}^m$. This allows us to focus on the genuine tokens that influence the label while ignoring the tokens involved in the structural pattern where the shortcut occurs. Here, k is determined by multiplying a scaling factor α that reflects the average number of non-casual words in the training data.

As shown in Figure 1, words such as 'For', 'an', and 'the' can be identified as non-causal words. The key point is that our method considers different non-causal words at each epoch while maintaining structural information that does not affect the label. This helps mitigate the tendency for the model to become biased towards spurious correlations as training progresses.

3.2 LLM-Driven Counterfactual Data Generation

To enhance model robustness and improve generalization, it is essential to incorporate not only structure-aware data but also diverse sentence patterns. A key aspect of our proposed method is leveraging LLMs to generate counterfactually negative data. Although the LLM-generated counterfactual sentences involve only minor changes prompted by simple instructions, they provide sufficient contextual variation and offer a range of sentence patterns.

Given a collection of training data $\{x_i\}_{i=1}^m$, we construct counterfactual data, denoted as $D_{neg} = \{x_i^-\}_{i=1}^m$, using LLM. In contrast to recent studies using LLMs to generate counterfactually augmented data (Dixit et al., 2022; Liu et al., 2022; Chen et al., 2023), we focus on generating counterfactual data using a simple prompt. To achieve this, we provide additional word information corresponding to the causal POS tag set in the prompt, minimally altering genuine tokens that directly impact the label. The causal POS tag set is obtained through the same process as deriving the non-causal tag set in Section 3.1, defining the causal POS tag set if their accuracy reduction exceeds the *threshold*.

Appendix C provides instructions used in our method. For the sentiment and sexism task, negative samples are created by changing tokens such that the label flips from *Positive (or sexist)* to *Negative (or non-sexist)* or from *Negative (or non-sexist)* to *Positive (or sexist)*. For the NLI task, we only consider two cases for generating negative samples: *Entailment* to *Contradiction* and *Contradiction* to *Entailment*, excluding the *Neutral*.

3.3 Contrastive Learning with Triplet Loss

Finally, we use contrastive learning for effective training of models in the generated counterfactual and positive data. First, the counterfactual data generated by altering only genuine tokens using LLM are considered not only to be a loss for direct label prediction but also to be a loss that encourages them to move further away from the original sentence in the latent space. Next, by bringing the positive data (structure-aware augmented data) closer to the original samples in the representation space, we effectively mitigate bias towards non-causal words and enhance the model's generalization ability. In summary, we aim to emphasize important features and eliminate unnecessary shortcuts through the generated triplets. In conventional fine-tuning models, the `[CLS]` hidden representations from PLM are passed through a classifier head to produce the probability distribution over the label set y. As a result, the model parameters θ are trained to minimize the cross-entropy loss between the predicted

label $\hat{y}$ and the ground-truth label y:

$$L_{CE} = \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \cdot \log \hat{y}_{i,c} \quad (2)$$

where N denotes a batch of training examples of size and C denotes classes.

We utilize a loss function similar to the training approach in C2L (Choi et al., 2022), which applied a margin-based ranking loss. The specific calculation of the triplet loss is as follows:

$$L_{CL} = \max(0, \frac{1}{m} \sum_{i=1}^{m} d(x_i, x_i^+) - d(x_i, x_i^-) + \alpha) \quad (3)$$

where m is the number of sentences, x_i represents the i-th original sentence, x_i^- and x_i^+ are the negative and positive sentence generated by our method, respectively. α is a margin value enforced between positive and negative pairs, and $d(\cdot)$ computes the distance between the hidden states of `[CLS]` tokens as the representations of two sentences.

The final loss function, which combines both the classification objective and the contrastive learning objective, is as follows:

$$L = (1 - \lambda) L_{CE} + \lambda L_{CL} \quad (4)$$

where λ is a scalar weighting hyperparameter that balances the two loss components and is tuned separately for each downstream task.

4 Experiment Setups

4.1 Datasets

Source Datasets To validate the ability of our method to address the phenomenon of being biased by spurious correlation in training data in various tasks, we conduct experiments on three tasks: sentiment classification, sexism classification, and natural language inference.

For sentiment classification and natural language inference tasks, we use the original dataset from Kaushik et al. (2020) where a counterfactually-revised dataset (CF) is paired with the original dataset (O). For the sexism classification task, we use one origin dataset from Samory et al. (2021), where the dataset contains pairs annotated by crowd workers, where sexist sentences are revised to non-sexist counterparts. We use these original-counterfactual pairs and ensure label balance by constructing an additional non-sexist dataset sampled from non-pairs in the dataset. Further, the original dataset is split in a 9:1 ratio for training and testing, with 10% of the training dataset aside for validation.

Evaluation Datasets In all tasks, we evaluate the robustness and generalization ability of our method using three in-domain datasets (IDD) and out-of-distribution datasets (ODD). For IDD evaluation, we use the test set from the original dataset (O-test) and the test set from the counterfactual dataset (CF-test). In the NLI task, the CF-Test datasets utilize a revised counterfactual dataset that combines both premise-revised and hypothesis-revised data.

For ODD evaluation, we use YELP (Asghar, 2016), SST2 (Socher et al., 2013), FineFood (McAuley and Leskovec, 2013), and the Tweet [1] for sentiment analysis. For the sexism task, we use another Tweet [2]. For the NLI task, we use the two types of MNLI (Williams et al., 2018) dataset.

Cross-Domain Datasets To further demonstrate the generalization ability of our method, we conduct cross-domain experiments. We use three sentiment datasets on SST-2 (Socher et al., 2013), IMDb (Maas et al., 2011), FineFood (McAuley and Leskovec, 2013). We utilize official train, validation, and test sets if available. In cases where such datasets are not provided, we randomly split the data into training and validation sets with an 8:2 ratio for each seed.

4.2 Baselines

We compare our proposed method with various data augmentation techniques, text data augmentation methods (Wei and Zou, 2019; Ng et al., 2020; Dai et al., 2023), counterfactual data augmentation methods (Kaushik et al., 2020; Dixit et al., 2022; Chen et al., 2023), and robust learning methods (Gunel et al., 2021; Choi et al., 2022). Detailed explanations are provided in the Appendix A.

4.3 Implementation Details

For all experiments, we use RoBERTa-large (Liu et al., 2019) as our PLM backbone, the batch size is 16, and the learning rate is 1e-05. For sentiment and sexism classification tasks, the maximum sequence length is 256, while it is 128 for NLI tasks.

[1] https://www.kaggle.com/c/tweet-sentiment-extraction

[2] https://www.kaggle.com/datasets/dgrosz/sexist-workplace-statements

Methods	In-Domain Dataset		Out-of-Distribution Dataset				Overall
	O-Test	CF-Test	YELP	SST2	FindFood	Tweet	
Standard Fine-Tuning (full-data)							
RoBERTa-large (Liu et al., 2019)	**94.13**	92.28	94.85	79.41	95.24	73.04	88.16
Robust Learning							
SupCon (Gunel et al., 2021)	93.85	88.11	95.26	86.20	95.32	74.90	88.94
C2L (Choi et al., 2022)	93.37	93.03	93.19	79.90	94.26	68.85	87.10
Text Data Augmentation							
EDA (Wei and Zou, 2019)	93.58	93.72	95.28	89.73	95.40	81.24	91.49
SSMBA (Ng et al., 2020)	93.60	92.69	**95.90**	89.40	**96.12**	78.75	91.08
AugGPT (Dai et al., 2023)	93.37	91.46	95.32	90.21	94.18	78.66	90.53
Counterfactual Data Augmentation							
Human-CAD (Kaushik et al., 2020)	93.17	95.47	92.16	88.65	94.26	80.66	90.73
CORE-CAD (Dixit et al., 2022)	91.73	95.15	89.70	90.10	93.06	**86.77**	91.09
SALAD	93.78	**95.90**	94.99	**92.68**	95.58	85.35	**93.05**

Table 1: **Accuracy of various approaches in sentiment classification task:** For the in-domain dataset, we use the original test set (O-Test) and counterfactual test set (CF-Test). The best performance is highlighted in **boldface**, and the second-best is marked as underlined.

Methods	IDD		ODD	Overall
	O-Test	CF-Test	Tweet	
RoBERTa-large	92.69	49.23	81.00	72.49
SupCon	91.79	22.56	76.28	60.84
C2L	**93.21**	37.69	77.92	67.18
EDA	91.67	37.69	81.59	67.74
SSMBA	92.82	25.64	79.36	63.02
AugGPT	92.31	29.23	78.83	64.08
Human-CAD	91.79	**91.80**	83.11	**89.47**
SALAD	93.07	88.47	**83.38**	88.31

Table 2: **Accuracy of various approaches in sexism classification task:** We evaluate on in-domain (IDD) and out-of-distribution (ODD) dataset.

Also, we run all experiments three times with three different random seeds and report the average performances. For each experiment that includes a contrastive objective, we employ different scalar weighting hyperparameters λ for each dataset that achieves the best performance. For counterfactual sample generation, we use GPT-4o-mini from OpenAI with a temperature of 0.1 and a Top-p value of 1. The *threshold*, used for determining the causal and non-causal tag set G, 0.1 across all datasets. The parameter α is defined as `0.18`, and based on the value of α, k is defined differently for each dataset. The dataset statistics, hyperparameters for each dataset, and the non-causal tag sets used in constructing positive samples are presented in Appendix B.

5 Results

In this section, we demonstrate the outperformance of *SALAD* for the robustness and generalization abilities of the model in three tasks. We also conduct comprehensive ablation studies to demonstrate its superiority.

5.1 Main Result

Robustness and Generalization As shown in Table 1, 2, and 3, *SALAD* demonstrates improved overall accuracy compared to *standard fine-tuning baseline* across all tasks. While it shows a slight decrease of about 1.16% compared to *Human-CAD* in the sexism classification task, making it the second-best, *SALAD* achieves the best performance in all other tasks. This indicates that our method is robust and performs similarly to human-generated high-quality CAD, even surpassing it in the sentiment classification task.

In particular, the results of the CF-Test for the sexism classification task show that *SALAD*'s performance drop is relatively small at just 3.33%, while other baselines show a significant performance drop compared to the CF-Test, indicating a lack of robustness. This suggests that fine-tuned PLMs using our *SALAD* are less sensitive to spurious patterns.

Furthermore, *SALAD* is particularly effective in ODD scenarios. While *Human-CAD* achieves the highest performance in IDD due to the use of CF-Train during training, its performance on ODD is

Methods	In-Domain		Out-of-Distribution		Overall
	O-test	CF-test	MNLI[1]	MNLI[2]	
Standard Fine-Tuning (full-data)					
RoBERTa-large (Liu et al., 2019)	87.50	69.90	73.27	73.97	76.16
Robust Learning					
SupCon (Gunel et al., 2021)	86.42	60.03	64.70	64.39	68.89
C2L (Choi et al., 2022)	87.96	68.49	72.18	72.74	75.34
Text Data Augmentation					
EDA (Wei and Zou, 2019)	86.59	67.58	70.93	71.12	74.06
SSMBA (Ng et al., 2020)	87.16	63.54	72.03	72.95	73.92
AugGPT (Dai et al., 2023)	86.92	69.61	73.62	74.38	76.13
Counterfactual Data Augmentation					
Human-CAD (Kaushik et al., 2020)	88.25	71.60	71.74	71.47	75.76
CORE-CAD (Dixit et al., 2022)	64.65	57.26	62.60	62.98	61.88
DISCO (Chen et al., 2023)	79.84	78.66	68.42	67.60	73.63
SALAD	**88.40**	**80.91**	**74.06**	**74.93**	**79.57**

Table 3: **Accuracy of various approaches in natural language inference task:** For out-of-distribution dataset, MNLI[1] refers to MNLI-hard-match, and MNLI[2] refers to MNLI-hard-mismatch. The best performance is highlighted in **boldface**, and the second-best is marked as underlined.

Methods	S → I	S → F	I → S	I → F	F → S	F → I	Overall
Standard Fine-Tuning (full-data)							
RoBERTa-large (Liu et al., 2019)	91.67	93.08	89.16	91.13	82.48	90.22	89.62
Robust Learning							
SupCon (Gunel et al., 2021)	90.82	89.64	91.21	94.95	73.40	89.68	88.28
C2L (Choi et al., 2022)	90.52	91.61	89.90	94.64	81.18	90.50	89.72
Text Data Augmentation							
EDA (Wei and Zou, 2019)	91.64	93.51	90.76	94.12	80.18	89.29	89.92
SSMBA (Ng et al., 2020)	90.71	90.78	**94.21**	93.96	78.75	89.31	89.62
SALAD	**92.41**	**94.19**	90.88	**94.96**	**86.00**	**91.25**	**91.61**

Table 4: **Accuracy of cross-domain task:** We evaluate across three datasets, SST-2 (S), IMDB (I), and FineFood (F) to evaluate the model's generalization ability of our method.

consistently lower compared to *SALAD*. This underscores that the proposed method significantly enhances generalization capabilities and ensures model robustness, leading to a dramatic improvement in overall performance.

Cross-Domain Generalization We additionally experiment with the performance of the domain generalization task to demonstrate that our proposed method is effective in securing robustness and enhancing generalization capabilities. As shown in Table 4, there is a substantial increase in performance, with the overall accuracy reaching 91.61% in six cross-domain settings.

Specifically, except in IMDB (I) → SST2 (S), all accuracy achieves the best performance. This indicates that the efforts to address spurious correlations in *SALAD* can potentially contribute to improving generalization abilities, even when the domain undergoes a shift.

5.2 Effectiveness of Our Data Augmentation

We conduct ablation studies on *SALAD* for sentiment classification and NLI tasks, focusing on two key aspects: whether our simple negative sample generation method using LLM outperforms using other complex methods and whether our tagging-based positive data augmentation method helps the model effectively learn structural patterns during contrastive learning, thus addressing spurious correlation issues and improving generalization in ODD.

Methods	Data Augmentation		Loss		Sentiment Task		NLI Task	
	Neg	Pos	CE	CL	IDD	ODD	IDD	ODD
Human-CAD	Human	X	X	X	94.32	88.93	79.93	71.60
CORE-CAD	GPT	X	X	X	93.44	89.91	60.96	62.79
GPT-CAD	GPT (ours)	X	X	X	92.42	89.59	71.35	68.78
SALAD-EDA	GPT (ours)	EDA	O	O	**94.98**	91.45	79.05	73.84
SALAD-GPT	GPT (ours)	AugGPT	O	O	94.88	91.86	78.70	72.51
SALAD	GPT (ours)	PosTag	O	O	94.84	**92.15**	**84.65**	**74.49**

Table 5: **Accuracy based on variations in SALAD:** GPT (ours) refers to counterfactually augmented data generated by GPT-3.5 using a simple prompt, and PosTag is generated structure-aware augmented data using pos-tagging information. IDD represents the in-domain dataset, and ODD represents the out-of-distribution dataset.

	Diversity	**Overlap (%)**	**BERTScore**
Human-CAD	1,392	92.68	0.969
CORE-CAD	498	60.15	0.914
GPT-CAD	1,490	86.77	0.969

Table 6: Analysis of the method of CAD on sentiment analysis. GPT-CAD is a counterfactually augmented dataset proposed in *SALAD*.

Original Sentence Long, boring, blasphemous. Never have I been so glad to see ending credits roll.
Human-CAD Long, fascinating, soulful. Never have I been so sad to see ending credits roll.
CORE-CAD I don't know why I hate this movie so much, now I am tired of watching it.
GPT-CAD Short, exciting, delightful. Always have I been so happy to see the beginning credits roll.

Table 7: Example of counterfactual data of our method (*GPT-CAD*) and other baselines. The purple indicates where the tokens are changed to flip the label.

In Table 5, the first three rows present the experimental results for training the model using only negative sample augmentation. Here, *GPT-CAD* shows effective performance in both tasks, particularly similar performance with the *Human-CAD*.

The last three rows demonstrate the results of comparing our tagging-based positive data augmentation method with other methods during contrastive learning. *SALAD* outperforms other baselines in both tasks, achieving notably higher performance in ODD. This indicates that our approach effectively enables the model to learn structural patterns, reduce bias, and make more accurate predictions in ODD scenarios, free from spurious correlations.

5.3 Quality of LLM-Driven Augmented Data

We evaluate our generated LLM-based counterfactual augmented data (GPT-CAD) in three metrics, as shown in Table 6. First, we measure the number of new corpora that did not appear in the original train dataset (**Diversity**). Second, we calculate the ratio of corpora that overlap with the original train dataset's (**Overlap**). Lastly, to examine how well the generated counterfactual sentences maintain the existing context, we use BERTScore (Zhang* et al., 2020), which computes cosine similarity between the original sentences and the generated counterfactual sentences using BERT encodings (**BERTScore**). As a result of our evaluation, *GPT-CAD* demonstrates performance close to *Human-CAD*, particularly with a BERTScore and diversity. This highlights the effectiveness of our LLM-driven counterfactual data generation compared to manually generated data. Additionally, as shown in Table 7, *GPT-CAD* effectively preserves the original sentence structure while changing only a minimal number of genuine tokens, outperforming previous methods in terms of both structure preservation and context. This suggests its ability to preserve the original context while altering keywords.

6 Conclusion

In this paper, we introduced **SALAD**, a method that enhances model robustness and generalization through the use of structure-aware and counterfactually augmented data for contrastive learning. By combining a tagging-based method for generating structure-preserving positive samples and LLM-generated counterfactual negative samples, *SALAD* enables models to learn meaningful structural relationships while reducing their reliance on spurious correlations. Our experiments, conducted on

three tasks demonstrate that *SALAD* significantly improves model performance across diverse environments, particularly in out-of-distribution and cross-domain settings. These results highlight the potential of our approach to address the challenges of spurious correlations in natural language processing, providing a more robust and generalizable solution.

7 Limitation

In this work, we utilized the GPT-4o-mini model to generate the dataset. GPT-CAD for *SALAD* consists of data in which sentence labels are flipped without requiring human intervention or additional models. While CADs that are re-labeled or generated by humans may yield better performance, our focus is not on meticulously generating CADs. Instead, we aim to verify and analyze the effectiveness of learning with CADs. Therefore, in future work, we believe that as higher-quality CADs become available, our proposed framework can be effectively utilized to further enhance model performance.

Acknowledgments

This work was partly supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant and the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (RS-2019-II190421 (10%), No.2022-0-01045 (45%), No.RS-2024-00360227 (45%))

References

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *CoRR*, abs/1605.05362.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. DISCO: Distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528, Toronto, Canada. Association for Computational Linguistics.

Seungtaek Choi, Myeongho Jeong, Hojae Han, and Seung-won Hwang. 2022. C2l: Causally contrastive learning for robust text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10526–10534.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *Preprint*, arXiv:2302.13007.

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

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. CORE: A retrieve-then-edit framework for counterfactual data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2964–2984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. 2021. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. 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.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13516–13524.

Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pages 897–908.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey. European Language Resources Association (ELRA).

Mattia Samory, Indira Sen, Julian Kohne, Fabian Flöck, and Claudia Wagner. 2021. "call me sexist, but...": Revisiting sexism detection using psychological scales and adversarial samples. In *Proceedings of the international AAAI conference on web and social media*, volume 15, pages 573–584.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. 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.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022. Identifying and mitigating spurious correlations for improving robustness in NLP models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1719–1729, Seattle, United States. Association for Computational Linguistics.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics.

Zhao Wang and Aron Culotta. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14024–14031.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Jiaxin Wen, Yeshuang Zhu, Jinchao Zhang, Jie Zhou, and Minlie Huang. 2022. AutoCAD: Automatically

generate counterfactuals for mitigating shortcut learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2302–2317, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

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

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 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)*, pages 306–316, Online. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022. FlipDA: Effective and robust data augmentation for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8646–8665, Dublin, Ireland. Association for Computational Linguistics.

A Baselines

Supcon (Gunel et al., 2021) SupCon is a joint optimization method combining cross-entropy loss and contrastive loss, demonstrating enhanced robustness and improved generalization performance in text classification tasks

C2L (Choi et al., 2022) C2L relies on the classifier model to identify causal words that significantly influence the label to enhance robustness. They treat the masking of causal words as negative examples, and the masking of less significant words as regular positive examples, thereby jointly optimizing triplet loss and cross-entropy.

EDA (Wei and Zou, 2019) This method proposed augmenting sentences by randomly applying four heuristic techniques: synonym replacement, word insertion, word deletion, and word swapping. We employed this method to augment our dataset by applying one augmentation per sentence.

SSMBA (Ng et al., 2020) SSMBA proposed a corrupt-and-reconstruct data augmentation technique using the BERT model, showing performance improvements on out-of-distribution datasets. In our experiments, we adopted the approach of augmenting data while keeping the labels unchanged. We also employed this method to augment our dataset by applying one augmentation per sentence.

AugGPT (Dai et al., 2023) This method used GPT-3 to augment data, enhancing the performance of text classification in a few-shot setting. In our experiments, we augment data using single-turn dialogues with the prompt "Please rephrase the following sentence."

Human-CAD (Kaushik et al., 2020) This method, which predominantly explores the automated generation of CAD, involves augmenting CAD by human annotators and training it with the original train dataset.

CORE-CAD (Dixit et al., 2022) CORE proposed a retrieval-augmented generation framework for generating CAD using a combination of a retrieval model and GPT-3. In our approach, we use the publicly available dataset on our experimental setup.

DISCO (Chen et al., 2023) For NLI tasks, we additionally compared DISCO, automatically generating high-quality counterfactual data at scale using the GPT-3 model.

Sentiment classification task

	Positive	Negative	Total
O-Train	856	851	1,707
O-Test	245	243	488
CF-Test	243	245	488

Sexism classification task

	Sexist	Non-sexist	Total
O-Train	1,036	1,036	2,072
O-Test	130	130	260
CF-Test	132	130	262

Natural language inference task

	Entail	Neutral	Contradict	Total
O-Test	146	123	131	400
O-Train	562	554	550	1,666
CF-Test	508	554	538	1,600

Table 8: Statistics of counterfactual task datasets.

B Implementation Details

B.1 Experimental Environment

For all experiments, our experiments are implemented with Pytorch framework (Paszke et al., 2019), Huggingface trasnformers (Wolf et al., 2020), NLTK library (Bird and Loper, 2004), OpenPrompt toolkit (Ding et al., 2021). We set the environment for all experiments using four NVIDIA 3090 GPUs with 24GB graphic memory, Ubuntu 22.04, Python 3.8, and CUDA 11.7 version.

B.2 Statistics of Counterfactual Task Dataset

Table 8 shows the statistics of the dataset used in the counterfactual task.

B.3 Hyper-parameters

As mentioned in the paper, we employ different hyperparameters, denoted as k and λ, for each dataset. In the structure-aware data generation using tagging information, the parameter k is used to determine the number of word tokens where randomly selected from each sentence that belongs to G and replace these tokens with the `[UNK]` token. According to our experimental results, defining k as 8 showed significant performance improvement for the CF-IMDB dataset, particularly on the out-of-distribution dataset (ODD) (Analysis results are displayed in Figure 2). Therefore, using CF-IMDB

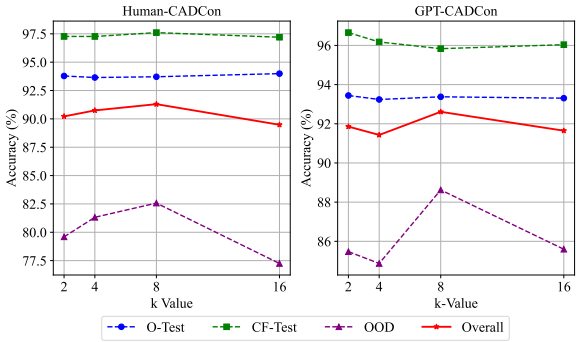

Figure 2: **Experiments on defining k:** The value of 8 shows significant performance improvement for the CF-IMDB dataset, particularly on the out-of-distribution dataset (ODD).

Dataset	k	λ
CF-IMDB (Kaushik et al., 2020)	8	0.9
CF-NLI (Kaushik et al., 2020)	1	0.9
Sexism (Samory et al., 2021)	1	0.3
SST2 (Socher et al., 2013)	1	0.1
IMDB (Maas et al., 2011)	8	0.9
FineFood (McAuley and Leskovec, 2013)	5	0.1

Table 9: **Hyperparameters of *SALAD*:** k represents the number of randomly selected tokens from the non-causal POS tag set (G), and λ is a scalar weighting hyperparameter used to define the final loss function.

as a reference, the scaling factor α was calculated. This calculation is determined by dividing the average number of non-causal tokens, which is 45 for CF-IMDB, resulting in a value of 0.18. Consequently, we calculate the value of k for each dataset by multiplying its respective average non-causal token count with the scaling factor. Summarizing the relevant hyperparameters, they are presented in Table 9.

B.4 Non-Causal Tag Sets Across Datasets

In the Structure-aware data generation methods, we define the non-causal tag set G by iteratively removing each POS tag set for each dataset and calculating the accuracy reduction. The following Figure 3 is an ablation study on the results of calculating accuracy reductions in sentiment and sexism classification datasets. We estimate θ to be 1%, defining the non-causal tag set as the POS information for which the score is less than 1%. The Causal tag sets calculated for each dataset used in our experiment are listed in Table 10.

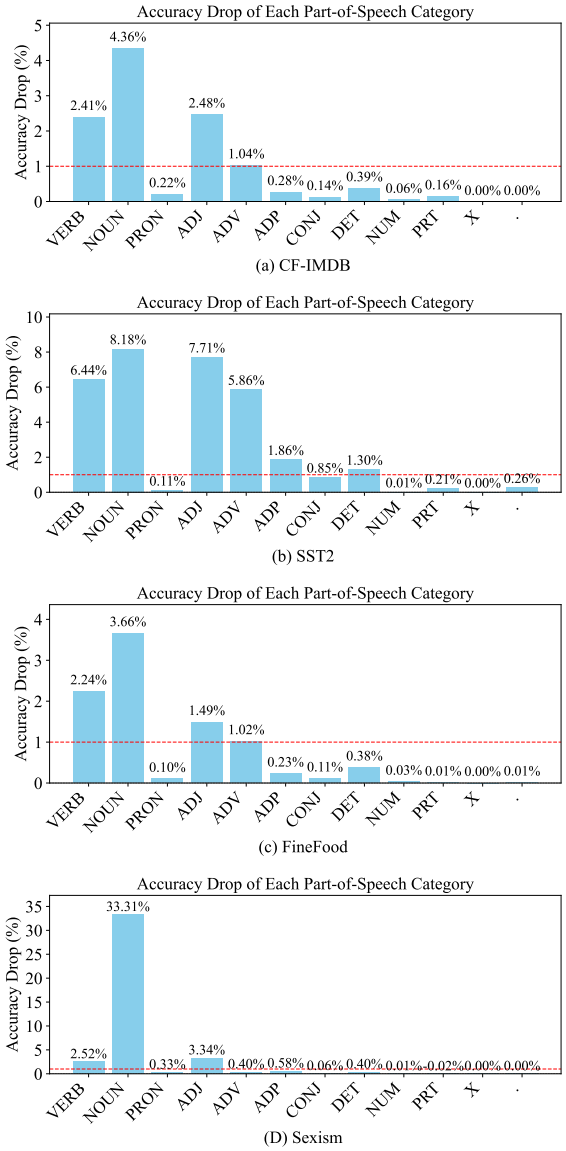

Figure 3: **Accuracy reduction of each POS category across datasets:** The x-axis represents each POS category, and the y-axis represents the average accuracy reduction. We define POS tags with an average accuracy reduction of less than 1% as the non-causal tag set G.

C Analysis of Prompt Instructions

As mentioned in Section 3.2, we utilize GPT-4o-mini to generate counterfactual sentences from the original ones using a simple prompt. Since zero-shot LLMs are sensitive to the template used, we conducted experiments with four variations of prompt instructions and selected the most effective one. *Instruction 1* includes the phrase, "*Please make it a negative sentence*", which directly indicates the intended behavior of the model. *Instruction 2* provides the task and label information for the sentence. In *Instruction 3*, we offer more spe-

Dataset	Causal tag set G
CF-IMDB	*VERB, NOUN, ADJ, ADV*
CF-NLI	*VERB, NOUN, ADJ, ADV*
Sexism	*VERB, NOUN, ADJ*
SST2	*VERB, NOUN, ADJ, ADV, ADP, DET*
IMDB	*VERB, NOUN, ADJ, ADV*
FineFood	*VERB, NOUN, ADJ, ADV*

Table 10: Causal tag sets in each training data.

Idx	Instructions
1	Please make it a negative sentence.
2	The following sentence is a positive sentence in sentiment analysis. Please make it a negative sentence.
3	The following sentence is a positive sentence in sentiment analysis. Just change a few words to make it a negative sentence while preserving the original text as much as possible.
4	The following sentence is a positive sentence in sentiment analysis. Just change a few words among causal words in the sentence to make it a negative sentence while preserving the original text as much as possible. Causal Words: ***causal words***

Table 11: Examples of instructions for generating negative samples in a sentiment analysis task. We use *Instruction 4* as our prompt to construct counterfactual data in *SALAD*

	Diversity	Overlap (%)	BERTScore
Human	1,392	92.68	0.969
Instruction 1	758	55.26	0.895
Instruction 2	1,183	76.91	0.934
Instruction 3	1,218	83.28	0.955
Instruction 4	1,490	86.77	0.969

Table 12: Analysis of our generated CAD (GPT-CAD) with different prompt instructions on sentiment analysis.

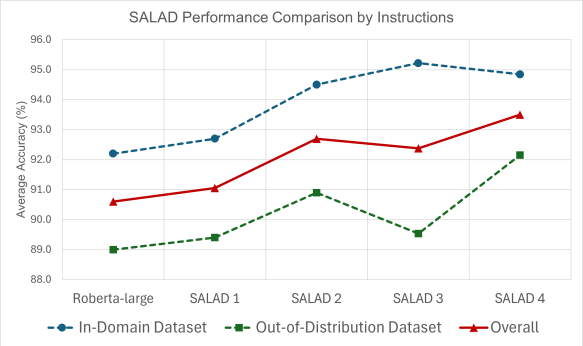

Figure 4: Performance variations of *SALAD* on datasets generated for each instruction. The number following "*SALAD*" corresponds to the instructions associated with each number used in Table 11.

cific guidance with phrases like, "*Just change a few words*" and "*while preserving the original text as much as possible*". We use a similarly designed prompt, following the format of *Instruction 3*, with adding causal word information, for each task and label. A specific example is shown in Table 11.

We aim to compare and analyze the performance and quality associated with each prompt instruction. We evaluate the generated CAD using three metrics. Additionally, we assess the performance of our CAD based on three prompt instructions. *Instruction 1*, which simply flips labels, shows a very low word overlap of 55.26% with the original sentence. Particularly in *instruction 4*, by incorporating the phrase "while preserving the original text as much as possible" and adding causal words information, we identify preservation of up to 86.77% of the original sentence while flipping the label. Moreover, with a diversity count of 1,490, indicating the number of corpora not used in the original sentence, it can be considered the most superior CAD among the four instructions. The CAD generated with *instruction 4* exhibits similarity to *Human-CAD*, as indicated by the BERTScore.

Also, we conduct an ablation study on datasets generated by four different prompts. Table 4 reports the performance of applying *SALAD* to the datasets generated through instructions for the four different scenarios. We find that even in instructions where task-related information is limited, such as in *SALAD_1*, there is a significant improvement in the ability to generalize to ODD data compared to the baseline model Roberta-large. Furthermore, the addition of task-related information in *SALAD_2* and the inclusion of the instruction "while preserving the original text as much as possible" in *SALAD_3* gradually lead to performance improvements. Particularly, *SALAD_4*, which generates CAD with the aim of minimally flipping the label by changing only genuine tokens, proves to be the most effective in achieving robustness through representation learning. Consequently, we utilized the *Instruction 4* in all final experiments.

D Prompt-based Fine-tuning Settings

Recently, in order to narrow the gap between pre-training and downstream tasks prompt-based Fine-tuning models are attracting attention and few-shot setting (Brown et al., 2020; Gao et al., 2021). Most prompt-based learning approaches (Shin et al., 2020; Schick and Schütze, 2021; Gao et al., 2021) utilize task-specific templates consisting of discrete prompts alongside input sentences. These prompts

Methods (8-shot)	In-Domain Dataset		Out-of-Distribtuion Dataset				Overall
	O-Test	CF-Test	YELP	SST2	Food	Tweet	
Prompt-based Fine-Tuning							
RoBERTa-large (Liu et al., 2019)	<u>92.21</u>	90.33	93.54	82.61	94.85	72.41	88.13
Robust Learning							
SupCon (Gunel et al., 2021)	91.52	90.45	**95.31**	84.16	<u>95.28</u>	73.51	88.80
Data Augmentation							
EDA (Wei and Zou, 2019)	91.02	91.64	94.18	84.34	94.79	71.00	88.53
SSMBA (Ng et al., 2020)	**92.25**	92.13	93.91	84.70	<u>95.28</u>	74.63	89.35
AugGPT (Dai et al., 2023)	92.13	92.30	92.68	81.55	94.64	70.53	88.07
Counterfactually Augmented Dataset							
Human-CAD	91.19	**93.16**	94.01	85.13	94.96	78.45	90.07
CORE-CAD	91.76	92.95	93.36	<u>88.30</u>	93.72	<u>81.50</u>	<u>90.67</u>
SALAD	91.11	<u>91.93</u>	<u>95.28</u>	**89.59**	**95.37**	**82.23**	**91.23**

Table 13: Accuracy of various approaches in sentiment classification task under the prompt-based fine-tuning setting. For the in-domain dataset, we use the original test set (O-Test) and counterfactual test set (CF-Test). The best performance is highlighted in **boldface**, and the second-best is marked as <u>underlined</u>.

contain a `[MASK]` token and are designed to construct an objective that is similar to MLM training, where the goal is to map the `[MASK]` token to the right label (a specific word) with a pre-defined verbalizer. The probability distribution over the label is shown below:

$$P_M(\texttt{[MASK]} = v|T(x))|v \in V_y) \quad (5)$$

where $T(\cdot)$ is a task-specific template and V_y is the label words of y. We conduct an 8-shot experiment with extremely low data volume for sentiment classification tasks to illustrate the enhancement of robustness.

Table 13 presents the results of experiments conducted in prompt-based fine-tuning settings. The results show that *SALAD* achieved state-of-the-art performance with an overall accuracy of 91.23%. It also delivered the best performance on three out-of-distribution datasets and demonstrated considerable performance in in-domain datasets, achieving the second-best result on the CF-Test. In conclusion, *SALAD* outperforms in terms of robustness and generalization abilities under the prompt-based fine-tuning setting.