# Similarizing the Influence of Words with Contrastive Learning to Defend Word-level Adversarial Text Attack

**Pengwei Zhan**[§‡], **Jing Yang**[§*], **He Wang**[§], **Chao Zheng**[§], **Xiao Huang**[§], **Liming Wang**[§]
[§]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[‡]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{zhanpengwei,yangjing,wanghe6029}@iie.ac.cn
{zhengchao1135,huangxiao,wangliming}@iie.ac.cn

## Abstract

Neural language models are vulnerable to word-level adversarial text attacks, which generate adversarial examples by directly substituting discrete input words. Previous search methods for word-level attacks assume that the information in the important words is more influential on prediction than unimportant words. In this paper, motivated by this assumption, we propose a self-supervised regularization method for **S**imilarizing the **I**nfluence of **W**ords with **Con**trastive Learning (SIWCon) that encourages the model to learn sentence representations in which words of varying importance have a more uniform influence on prediction. Experiments show that SIWCon is compatible with various training methods and effectively improves model robustness against various unforeseen adversarial attacks. The effectiveness of SIWCon is also intuitively shown through qualitative analysis and visualization of the loss landscape, sentence representation, and changes in model confidence.

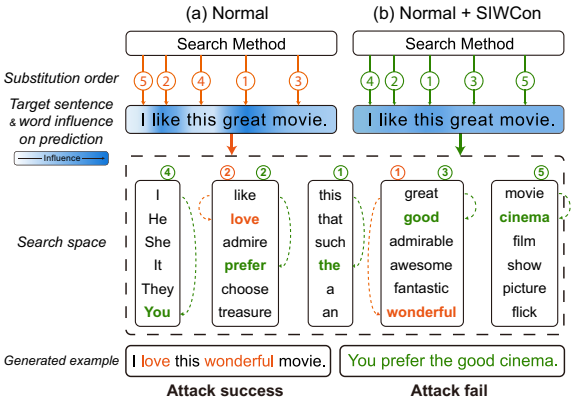



Figure 1: The motivation of SIWCon. The normally trained model considers the information in important words to have a significant impact on prediction, and as a result, search methods that prioritize substituting important words are more likely to find adversarial examples. SIWCon, on the other hand, considers the information in both important and unimportant words to have a similar degree of influence on prediction, thus making it less possible for search methods that focus on important words to find optimal substitutions.

## 1 Introduction

Neural language models have achieved impressive performance in various natural language processing (NLP) tasks, but they are also proven vulnerable to adversarial examples, which induce incorrect model output by adding small perturbations to natural inputs (Szegedy et al., 2014; Jia and Liang, 2017). Unlike attacks on images, which are performed by directly adding imperceptible continuous noise to the input, adversarial text attacks are commonly performed by substituting input text due to the discrete and non-differentiable nature of text (Gao et al., 2018; Alzantot et al., 2018; Li et al., 2019; Zhan et al., 2022b; Garg and Ramakrishnan, 2020). Among the various granularities of adversarial text attacks, word-level attacks have been more focused on by recent works for their effectiveness in maintaining semantic similarity and grammatical correctness. Unlike character-level and sentence-level attacks, word-level attacks are less likely to be detected by spell checkers or to undermine the overall coherence of a sentence (Ebrahimi et al., 2018; Iyyer et al., 2018; Liang et al., 2018).

Under a unified framework, word-level attacks can always be formulated as a combinatorial optimization problem (Yoo et al., 2020; Morris et al., 2020a,b), and various attack methods can be decomposed into *Search Space* and *Search Method*. The search space contains the possible substitutions for each word, while the search method determines the substitution order and strategy for selecting the optimal substitution from the search space. Since the search space may be model-agnostic, we should focus on the search method for the potential of improving the robustness against word-level attacks.

Previous search methods for word-level attacks are based on the assumption: *different words in*

*Corresponding Author.



*Findings of the Association for Computational Linguistics: ACL 2023*, pages 7891–7906
July 9-14, 2023 

*a sentence contribute differently to model prediction, with the information in important words being more influential than the information in unimportant words.* Therefore, following the word importance scores obtained through attribution methods, the attack can be seen as a process of *iteratively* substituting words in a sentence, with important words substituted first, followed by unimportant words. For example, the search methods Word Importance Ranking (WIR) (Gao et al., 2018; Jin et al., 2020; Li et al., 2020) and PWWS (Ren et al., 2019) obtain word importance using Occlusion (Zeiler and Fergus, 2014), then WIR performs substitution in descending order of word importance and PWWS formulates token scores that use word importance as weights to guide the attack.

Following this assumption, the success of word-level attacks can be explained. The words in a sentence can be classified as important words, which contain more influential information for prediction, or unimportant words, which contain less influential information. Search methods that substitute important words first can perturb more influential information in each attack step, making the model more likely to be deceived. Therefore, it is natural to wonder: *will the model be more robust when the information in both important and unimportant words has a similar degree of influence on prediction?* Motivated by this question, we propose a self-supervised regularization method for **S**imilarizing the **I**nfluence of **W**ords with **Con**trastive Learning (SIWCon) that improves the model robustness against word-level attacks. The motivation of our method is illustrated in Figure 1. We summarize our main contributions as follows:

1. We discuss the relationship between model robustness and the influence of information in words of different importance.

2. We propose SIWCon, a contrastive learning method that improves the robustness of language models by encouraging models to learn sentence representations that consider the information in words of different importance to have a more similar influence on prediction.

3. We evaluate SIWCon against several attack methods on three models of different architectures and on Movie Review (MR), SST2, and IMDB datasets. Results show that SIWCon improves the model robustness against unforeseen adversarial attacks *without learning from any adversarial perturbation*.

4. We provide qualitative analysis and visualization on loss landscape, sentence representation, and model confidence change, intuitively showing the effectiveness of SIWCon.

## 2 Related Works

**Robustness of Language Models.** The current methods for improving the robustness of language models ignore the assumption discussed in §1. While some works attempt to detect or transform potential adversarial examples before the model (Zhou et al., 2019; Mozes et al., 2021), this does not actually improve the model's robustness. Other methods, such as performing certifiably robust training through interval bound propagation (IBP), can be computationally costly and difficult to scale to large models like BERT (Jia et al., 2019; Huang et al., 2019). Additionally, it has been reported that while IBP improves adversarial accuracy, it comes at the huge cost of reduced clean accuracy (Wang et al., 2021). Some works try to perform adversarial training by incorporating adversarial examples in the training set (Jin et al., 2020; Li et al., 2021), while this method can only improve the robustness against the adversarial perturbations that the model has seen. Moreover, generating adversarial examples is time-consuming, thus adversarial training is difficult to scale to a large dataset. In this paper, based on the ignored assumption, we discuss the model robustness from new perspectives, focusing on attribution and sentence representation.

**Contrastive Learning.** Contrastive learning is first proposed in computer vision tasks to help models learn better image representation (Chen et al., 2020a,b; He et al., 2020; Pan et al., 2021). This self-supervised learning method alleviates the dependence on the costly labeled data. Recently, encouraged by the superior performance, various contrastive learning methods have been proposed for NLP tasks. Following the discrete nature of text, some previous works construct the pair examples by augmenting the input sentence (Giorgi et al., 2021; Wu et al., 2020; Fang and Xie, 2020; Zhan et al., 2022a; Gao et al., 2021), e.g., by word deleting, reordering, substituting, and back-translating, or by augmenting the word embedding (Yan et al., 2021), e.g., by shuffling, cutting off, dropping out the embedding matrix. Unlike the previous works

that aim to improve the downstream performance, we focus on improving the model robustness.

## 3 Methodology

### 3.1 Preliminaries

Suppose we have the input text $\boldsymbol{X} \in \mathcal{X}$ and the output labels $Y \in \mathcal{Y} = \{1, \ldots, C\}$ that follow the data distribution $\mathcal{D}$. A model $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$ that maps the input text to the output probability space is trained by minimizing $\mathcal{L}_{ce}(\boldsymbol{X}, Y; \boldsymbol{\theta})$:

$$\mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}[-\log \frac{\exp(w_Y^T r_{\boldsymbol{\theta}}(\boldsymbol{X}))}{\sum_{k=1}^{C} \exp(w_k^T r_{\boldsymbol{\theta}}(\boldsymbol{X}))}] \,, \quad (1)$$

where $w_Y \in \mathcal{W}$ denotes the model classification parameters toward class $Y$, $\mathcal{W}$ is the overall classification parameters, and $r_{\boldsymbol{\theta}}(\cdot)$ denotes the latent sentence representation encoded by the model $f$ with parameters $\boldsymbol{\theta}$. The well-trained model can learn the distribution of data and predict the input sentence based on the posterior probability:

$$P(Y_{true}|\boldsymbol{X}) = \frac{\exp(w_{true}^T r_{\boldsymbol{\theta}}(\boldsymbol{X}))}{\sum_{k=1}^{C} \exp(w_k^T r_{\boldsymbol{\theta}}(\boldsymbol{X}))} \,, \quad (2)$$

where $w_{true}$ denotes the classification parameters toward the ground-truth class $Y_{true}$. To attribute the prediction $P(Y_{true}|\boldsymbol{X})$, i.e., identifying the words that are most influential on the prediction (Li et al., 2016b; Ross et al., 2017; Sundararajan et al., 2017; Kim et al., 2020), we use the gradient-based attribution method (Feng et al., 2018; Li et al., 2016a; Arras et al., 2016; Situ et al., 2021). The influence score of word $x_i \in \boldsymbol{X}$ can be formally defined as:

$$Score(x_i) = \left\| \frac{\partial\, w_{true}^T r_{\boldsymbol{\theta}}(\boldsymbol{X})}{\partial\, emb(x_i)} \right\|_2 \,, \quad (3)$$

where $emb(\cdot)$ denotes embedding, and $\|\cdot\|_2$ denotes $L^2$ norm. The influence of a word is the norm of the influence score of every embedding dimension.

### 3.2 Word-level Adversarial Attack

Following the analysis of word-level adversarial attacks in §1, an adversarial example $\boldsymbol{X}^{adv}$ generated by search methods from a normal example $\boldsymbol{X} = (x_n)_{n\in\{1,\ldots,N\}}$ can be formulated as:

$$\begin{aligned} &\boldsymbol{X}^{adv} = \mathcal{O}(\boldsymbol{X}) = o(x_n)_{n\in\{1,\ldots,N\}}, \\ &\text{s.t.} \quad \forall n \in \{1,\ldots,N\}, \;\; \Delta x_n < \delta, \\ &\text{and} \quad \Delta \boldsymbol{X} < \varepsilon, \\ &\text{and} \quad \operatorname*{argmax}_{Y\in\mathcal{Y}} \mathcal{P}(Y|\boldsymbol{X}^{adv}) \neq \operatorname*{argmax}_{Y\in\mathcal{Y}} \mathcal{P}(Y|\boldsymbol{X}), \end{aligned} \quad (4)$$

where $\mathcal{O}(\boldsymbol{X})$ denotes performing word-level substitution on sentence $\boldsymbol{X}$, $o(x_n)$ denotes substituting the word $x_n$ with a new word from a finite search space that contains all qualified substitutions, if possible. $\Delta x_n$ and $\delta$ respectively denote the difference and the maximum allowed difference between $x_n$ and $o(x_n)$, $\Delta \boldsymbol{X}$ and $\varepsilon$ respectively denote the difference and the maximum allowed difference between $\boldsymbol{X}$ and $\mathcal{O}(\boldsymbol{X})$. $\delta$ and $\varepsilon$ are used to filter qualified substitutions in the search space, which may mainly focus on the semantics and the $L^p$ norm of the embedding distance of each word and the entire sentence, ensuring the adversarial example is imperceptible to humans.

To generate adversarial examples more effectively, the search methods of current attacks, i.e., the strategies to perform $o(\cdot)$, follow the assumption that *the information in important words is more influential than the information in unimportant words*, and heavily rely on attribution results like (3). These methods attempt to substitute important words first to perturb more influential information in each attack step. Therefore, if different words in a sentence have a similar slight influence on prediction, the attacks should only slightly impact the model prediction in each attack step. To this end, we detail the SIWCon regularization method next.

### 3.3 The SIWCon Regularization

Recall that the goal of SIWCon is to *similarize the influence of words*. After regularization, the influence of different words on prediction should be similarly slight. To formally define this goal, we first define the 40% of words in a sentence with the highest and lowest influence scores as the important and unimportant words, respectively, following the attribution results of (3). We then propose two efficient non-deterministic data augmentation operations, $t^{imp}(\cdot)$ and $t^{ump}(\cdot)$, which respectively means randomly removing important and unimportant words in a sentence. Therefore, under the training scenario of (1), the primary goal of SIWCon can now be formulated as:

$$\begin{aligned} &\min_{\theta} \quad ||\mathcal{Q}_{imp.} - \mathcal{Q}_{ump.}|| \;: \quad (5) \\ &\mathcal{Q}_{imp.} = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{X},Y)\sim\mathcal{D} \\ \boldsymbol{X}^{imp}\sim t^{imp}(\boldsymbol{X})}} [\mathcal{P}(Y_{true}|\boldsymbol{X}) - \mathcal{P}(Y_{true}|\boldsymbol{X}^{imp})] \,, \\ &\mathcal{Q}_{ump.} = \mathop{\mathbb{E}}_{\substack{(\boldsymbol{X},Y)\sim\mathcal{D} \\ \boldsymbol{X}^{ump}\sim t^{ump}(\boldsymbol{X})}} [\mathcal{P}(Y_{true}|\boldsymbol{X}) - \mathcal{P}(Y_{true}|\boldsymbol{X}^{ump})] \,, \end{aligned}$$

where $\boldsymbol{X}^{imp}$ is an augmentation sampled from $t^{imp}(\boldsymbol{X})$, and $\boldsymbol{X}^{ump}$ is an augmentation sampled

from $t^{ump}(\boldsymbol{X})$. $\mathcal{Q}_{imp}$ and $\mathcal{Q}_{ump}$ measure the extent of model confidence decrease when a random part of information in the important and unimportant words is lost, indicating the *overall* influence of the information in words of different importance on prediction. The complete objective of SIWCon can be further decomposed into two perspectives:

**Objective 1:** The influence of different words should be *similar*, thus the model should treat the sentences with information in words of different importance lost ($\boldsymbol{X}^{imp}$ and $\boldsymbol{X}^{ump}$) similarly.

**Objective 2:** The influence of different words should be *slight*, thus the model should treat the sentences with different information lost ($\boldsymbol{X}^{imp}$ and $\boldsymbol{X}^{ump}$) similarly to the original sentence that contains complete information ($\boldsymbol{X}$).

To achieve **Objective 1** and **Objective 2**, and further the goal of SIWCon, we use a contrastive loss objective from the perspective of sentence representation. To define the contrastive loss objective, for convenience, we first define the calculation $\mathcal{S}$:

$$\mathcal{S}_{(i,j)}^{(k,l)} = \exp(\text{sim}[r_{\boldsymbol{\theta}}(\boldsymbol{X}_i^k), r_{\boldsymbol{\theta}}(\boldsymbol{X}_j^l)]/\tau) \,, \quad (6)$$

where $k, l \in \{imp, ump, \cdot\}$, respectively indicate the augmentation sampled from $t^{imp}(\cdot)$, the augmentation sampled from $t^{ump}(\cdot)$, and the normal example, $i, j$ are the example indices, $\text{sim}[\boldsymbol{r_i}, \boldsymbol{r_j}] = \boldsymbol{r_i}^T \boldsymbol{r_j} / \|\boldsymbol{r_i}\| \|\boldsymbol{r_j}\|$ is the cosine similarity, $\tau$ is a temperature parameter similar to the NT-Xent loss (Chen et al., 2020a; van den Oord et al., 2018). Then the contrastive loss function for an example in a mini-batch $\boldsymbol{X}_i \in \{\boldsymbol{X}_i\}_{i=1}^B$ is defined as:

$$\mathcal{L}_{SIWCon}(\boldsymbol{X}_i; \boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\substack{\{\boldsymbol{X}_i\}_{i=1}^B \sim \mathcal{D} \\ \boldsymbol{X}_i^{ump} \sim t^{ump}(\boldsymbol{X}_i) \\ \boldsymbol{X}_i^{imp} \sim t^{imp}(\boldsymbol{X}_i)}} [-\log \frac{\mathcal{S}_{positive}}{\sum_{j=1}^B (\mathcal{S}_{negative})}] \,, \quad (7)$$

where

$$\mathcal{S}_{positive} = \mathcal{S}_{(i,i)}^{(imp,ump)} + \mathcal{S}_{(i,i)}^{(\cdot,ump)} + \mathcal{S}_{(i,i)}^{(\cdot,imp)} \,,$$
$$\mathcal{S}_{negative} = \mathcal{S}_{(i,j)}^{(\cdot,\cdot)} + \mathbb{1}_{[i \neq j]} [\mathcal{S}_{(i,j)}^{(\cdot,ump)} + \mathcal{S}_{(i,j)}^{(\cdot,imp)}] \,,$$

$B$ is the batch size, $\mathbb{1}_{[\cdot]}$ is an indicator function that equals 1 if the condition $[\cdot]$ is true; otherwise, it equals 0. Specifically, to calculate the loss for each mini-batch, we first randomly sample the augmentations $\boldsymbol{X}_i^{ump}$ from $t^{ump}(\boldsymbol{X}_i)$ and the augmentations $\boldsymbol{X}_i^{imp}$ from $t^{imp}(\boldsymbol{X}_i)$ for each example in the mini-batch. The general framework of SIWCon is shown in Figure 2.

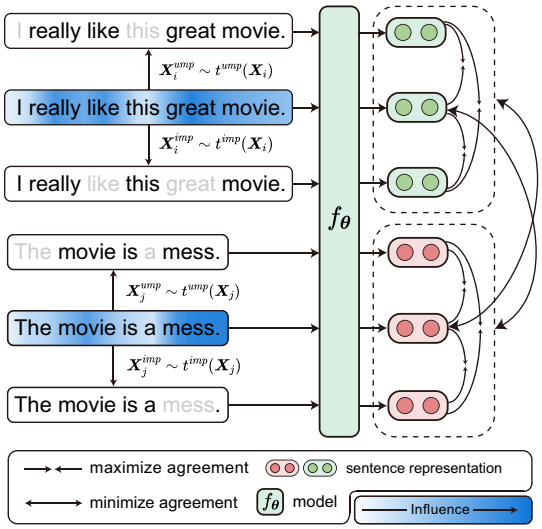



Figure 2: The framework of SIWCon. The word importance for each sentence is obtained through attribution (3), and the augmentations of each sentence are sampled from the non-deterministic transformations $t^{imp}$ and $t^{ump}$. The contrastive objective is calculated on the sentence representations learned by the same model $f_{\boldsymbol{\theta}}$.

To achieve Objective 1, we use the term $\mathcal{S}_{(i,i)}^{(imp,ump)}$ in the numerator. This constraint maximizes the similarity between the representations of the augmentations with important and unimportant words removed, making the different degrees of incomplete information in the augmentations have a similar impact on the prediction.

To achieve Objective 2, we use the term $\mathcal{S}_{(i,i)}^{(\cdot,ump)}$ and $\mathcal{S}_{(i,i)}^{(\cdot,imp)}$ in the numerator. These constraints maximize the similarity between the original sentence and the two augmentations, making the incomplete information in the remaining words of the augmentations have a similar influence as the complete information in the normal sentence.

Intuitively, the semantics of different examples should be different, and following the constraints in $\mathcal{S}_{positive}$, the semantics of the augmentations of different examples should also be different. Therefore, the three terms in $\mathcal{S}_{negative}$ denote that, given an example within a mini-batch, we treat both the other examples and the augmentations derived from other examples as negative examples.

The final loss of SIWCon regularization is computed across all examples in a mini-batch. When SIWCon is used in the normal training scenario (1), the overall objective is:

$$\min_\theta \; \mathcal{L}_{ce}(\boldsymbol{X}, Y) + \alpha \, \mathcal{L}_{SIWCon}(\boldsymbol{X}) \,, \quad (8)$$

where $\alpha$ is a parameter balancing the supervised part and the contrastive regularization part.

| Model | Method | MR | | | | SST2 | | | | IMDB | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | DeepWordBug | | TextFooler | | DeepWordBug | | TextFooler | | DeepWordBug | | TextFooler | |
| | | *ACC.* ↑ | *AUA.* ↑ | *ACC.* ↑ | *AUA.* ↑ | *ACC.* ↑ | *AUA.* ↑ | *ACC.* ↑ | *AUA.* ↑ | *ACC.* ↑ | *AUA.* ↑ | *ACC.* ↑ | *AUA.* ↑ |
| LSTM | Normal | **77.01** | 3.66 | **77.01** | 0.33 | 80.96 | 4.67 | **80.96** | 0.33 | **77.38** | 0.30 | 77.38 | 0.00 |
| | +SIWCon | 76.84 | **23.00** | 76.74 | **1.67** | **81.19** | **12.00** | 80.39 | **4.67** | 76.32 | **15.67** | **78.55** | **8.33** |
| | AT | **76.45** | 40.00 | **75.79** | 2.00 | **78.78** | 46.33 | **80.05** | 1.67 | **74.41** | 47.67 | **77.32** | 0.33 |
| | +SIWCon | 76.08 | **54.00** | 75.04 | **6.00** | 78.56 | **55.33** | 79.59 | **3.68** | 74.07 | **56.67** | 76.03 | **3.67** |
| TextCNN | Normal | 77.58 | 9.66 | **77.58** | 3.33 | **79.47** | 15.67 | 79.47 | 4.33 | **76.60** | 2.33 | **76.60** | 5.67 |
| | +SIWCon | **77.67** | **15.67** | 76.64 | **6.33** | 78.73 | **19.33** | **80.73** | **8.00** | 76.01 | **22.00** | 75.24 | **9.67** |
| | AT | **75.23** | 43.00 | 73.73 | 10.33 | **73.74** | 68.67 | 75.11 | 10.67 | **74.34** | 33.00 | **76.72** | 9.00 |
| | +SIWCon | 74.26 | **53.00** | **74.48** | **14.33** | 73.28 | **71.33** | **75.22** | **16.00** | 73.73 | **45.67** | 75.58 | **22.67** |
| BERT | Normal | **86.12** | 9.67 | **86.12** | 8.33 | **91.74** | 24.67 | **91.74** | 12.33 | 83.44 | 12.33 | 83.44 | 5.67 |
| | +SIWCon | 85.46 | **60.33** | 84.31 | **30.67** | 90.94 | **32.00** | 90.83 | **19.33** | **83.93** | **22.33** | **83.92** | **10.33** |
| | AT | **86.68** | 68.33 | 84.80 | 34.67 | **91.63** | 72.00 | 91.51 | 34.67 | **83.46** | 51.67 | 83.24 | 31.33 |
| | +SIWCon | 86.49 | **77.33** | **84.90** | **40.00** | 90.85 | **76.33** | **91.63** | **41.67** | 83.16 | **64.67** | **83.62** | **37.33** |

Table 1: The comparisons of model accuracy (*ACC.*) and accuracy under attack (*AUA.*). The **bold** values of *ACC.*, *AUA.* indicate the best performance and best robustness, respectively. *Normal* and *Normal+SIWCon* are under the *unforeseen* scenario, indicating the model do not learn from any adversarial perturbation. Conversely, *AT* and *AT+SIWCon* are under the *foreseen* scenario, indicating the model learn from adversarial perturbation in training.

# 4 Experiment

## 4.1 Metrics

We measure the model performance with *Accuracy (ACC.)*, the model robustness with *Accuracy Under Attack (AUA.)*, and the influence of words with three *Area Over the Perturbation Curve (AOPC)* metrics (DeYoung et al., 2020; Samek et al., 2017; Nguyen, 2018). $AOPC_{Comp.}$ and $AOPC_{Suff.}$ respectively measure the overall influence of the information in important and unimportant words on prediction. $AOPC_{Comp.}$ is formulated as:

$$\frac{1}{K+1}\sum_{k=1}^{K}\mathcal{P}(Y_{true}|\boldsymbol{X}) - \mathcal{P}(Y_{true}|t^{imp}_{/k}(\boldsymbol{X})), \quad (9)$$

and $AOPC_{Suff.}$ is formulated as:

$$\frac{1}{K+1}\sum_{k=1}^{K}\mathcal{P}(Y_{true}|\boldsymbol{X}) - \mathcal{P}(Y_{true}|t^{ump}_{/k}(\boldsymbol{X})), \quad (10)$$

where $t^{imp}_{/k}$ and $t^{ump}_{/k}$ are deterministic transformations that remove the $k$ most and least important words in a sentence, respectively. We also use $AOPC_{Diff.}$ to indicate the difference between $AOPC_{Comp.}$ and $AOPC_{Suff.}$, measuring how the goal of SIWCon is achieved.

## 4.2 Experiment Setup

**Setup.** We conduct experiments on MR (Pang and Lee, 2005), SST2 (Socher et al., 2013), and IMDB (Maas et al., 2011) datasets. We use LSTM (Hochreiter and Schmidhuber, 1997), TextCNN (Kim, 2014), and the base version of BERT (Devlin et al., 2019) as models. More details of the datasets and models can be found in Appendix A.1 and A.2. We use Normal training (1) and Adversarial training (AT, detailed in Appendix A.3) as basic training methods. In the main experiment, we use DeepWordBug (Gao et al., 2018) and TextFooler (Jin et al., 2020) as attack methods. We also use BAE (Garg and Ramakrishnan, 2020), TextBugger (Li et al., 2019), and PWWS (Ren et al., 2019) in the analysis.

**Implementation Details.** The $K$ in (9) and (10) are set as 40% of each sentence's length. We use Adam (Kingma and Ba, 2015) as the optimizer. For LSTM and TextCNN, we use the average token embedding before the last dense layer as the sentence representation. For BERT, we use the `[CLS]` token embedding as the sentence representation. Unless otherwise specified, the batch size is set as 32, the learning rate/$\alpha$/$\tau$ for LSTM, TextCNN, and BERT is 1e-3/1.2/0.01, 1e-3/1.2/0.05, and 3e-5/0.005/1.5. The reported results are the average of five individual runs with randomly picked seeds.

## 4.3 Main Results

In the main experiment, we train the models on three datasets with different training methods and then measure their robustness by attacking 600 examples randomly picked from the testing set. Following Jin et al. (2020) and Li et al. (2021), for adversarial training, we incorporate the adversarial examples of 10% randomly picked training data into the new training set, which are generated by the same attack method for measuring robustness.

| Model | Method | DeepWordBug | | | TextFooler | | |
|---|---|---|---|---|---|---|---|
| | | $A_{Comp.}$ | $A_{Suff.}$ | $A_{Diff.}\downarrow$ | $A_{Comp.}$ | $A_{Suff.}$ | $A_{Diff.}\downarrow$ |
| *MR* | | | | | | | |
| LSTM | Normal | 0.096 | 0.046 | 0.050 | 0.096 | 0.046 | 0.050 |
| | +SIWCon | 0.070 | 0.035 | **0.035** | 0.070 | 0.032 | **0.038** |
| | AT | 0.084 | 0.037 | 0.047 | 0.072 | 0.032 | 0.040 |
| | +SIWCon | 0.066 | 0.040 | **0.026** | 0.048 | 0.021 | **0.027** |
| TextCNN | Normal | 0.094 | 0.024 | 0.070 | 0.094 | 0.024 | 0.070 |
| | +SIWCon | 0.090 | 0.028 | **0.062** | 0.058 | 0.004 | **0.054** |
| | AT | 0.096 | 0.037 | 0.059 | 0.087 | 0.031 | 0.056 |
| | +SIWCon | 0.114 | 0.063 | **0.051** | 0.076 | 0.024 | **0.052** |
| BERT | Normal | 0.064 | 0.018 | 0.046 | 0.064 | 0.018 | 0.046 |
| | +SIWCon | 0.030 | 0.015 | **0.015** | 0.038 | 0.022 | **0.016** |
| | AT | 0.054 | 0.029 | 0.025 | 0.042 | 0.016 | 0.026 |
| | +SIWCon | 0.050 | 0.035 | **0.015** | 0.036 | 0.029 | **0.007** |
| *SST2* | | | | | | | |
| LSTM | Normal | 0.083 | 0.022 | 0.061 | 0.083 | 0.022 | 0.061 |
| | +SIWCon | 0.071 | 0.017 | **0.054** | 0.055 | –0.004 | **0.059** |
| | AT | 0.099 | 0.027 | 0.072 | 0.075 | 0.006 | 0.069 |
| | +SIWCon | 0.078 | 0.026 | **0.052** | 0.066 | 0.010 | **0.056** |
| TextCNN | Normal | 0.094 | 0.028 | 0.066 | 0.094 | 0.028 | 0.066 |
| | +SIWCon | 0.078 | 0.016 | **0.062** | 0.087 | 0.026 | **0.061** |
| | AT | 0.031 | –0.007 | 0.038 | 0.046 | –0.006 | 0.052 |
| | +SIWCon | 0.040 | 0.010 | **0.030** | 0.303 | –0.018 | **0.048** |
| BERT | Normal | 0.042 | 0.013 | 0.029 | 0.042 | 0.013 | 0.029 |
| | +SIWCon | 0.038 | 0.020 | **0.018** | 0.045 | 0.025 | **0.020** |
| | AT | 0.041 | 0.015 | 0.026 | 0.032 | 0.017 | 0.015 |
| | +SIWCon | 0.047 | 0.031 | **0.016** | 0.028 | 0.020 | **0.008** |
| *IMDB* | | | | | | | |
| LSTM | Normal | 0.070 | 0.006 | 0.064 | 0.070 | 0.006 | 0.064 |
| | +SIWCon | 0.048 | 0.041 | **0.007** | 0.064 | 0.045 | **0.019** |
| | AT | 0.083 | 0.024 | 0.059 | 0.033 | 0.002 | 0.031 |
| | +SIWCon | 0.012 | 0.008 | **0.004** | 0.113 | 0.088 | **0.026** |
| TextCNN | Normal | 0.124 | 0.041 | 0.083 | 0.124 | 0.041 | 0.083 |
| | +SIWCon | 0.077 | 0.023 | **0.054** | 0.065 | 0.018 | **0.047** |
| | AT | 0.108 | 0.040 | 0.068 | 0.078 | 0.024 | 0.054 |
| | +SIWCon | 0.112 | 0.088 | **0.024** | 0.114 | 0.096 | **0.018** |
| BERT | Normal | 0.059 | 0.023 | 0.036 | 0.059 | 0.023 | 0.036 |
| | +SIWCon | 0.042 | 0.027 | **0.015** | 0.057 | 0.026 | **0.031** |
| | AT | 0.084 | 0.036 | 0.048 | 0.048 | 0.005 | 0.043 |
| | +SIWCon | 0.062 | 0.021 | **0.041** | 0.044 | 0.013 | **0.031** |

Table 2: The comparisons on the overall influence of the information in the words of different importance on prediction. The **bold** values of $AOPC_{Diff.}$ indicate the most similar influence and the best achievement of the goal of SIWCon. *A* is short for *AOPC*.

**SIWCon has a slight impact on clean accuracy.** The results of clean accuracy are illustrated in Table 1. SIWCon only slightly impacts the clean accuracy when combined with other training methods. *Normal+SIWCon* sometimes outperforms *Normal* method, and the average accuracy difference between the two methods is only 0.97%. *AT+SIWCon* causes a slight drop in model accuracy compared to *Normal* method, with the negative impact mainly resulting from the integration of adversarial examples rather than the usage of SIWCon. The average accuracy difference between *AT* and *AT+SIWCon* is only 0.36%, and 1.53% between *Normal* and *AT*.

**SIWCon improves model robustness.** The results of robustness are illustrated in Table 1. SIWCon is a self-supervised regularization method that relies solely on the training data (not including labels) and their augmentations generated by removing words, without learning from any adversarial perturbations. Nevertheless, SIWCon is effective in improving model robustness. Under the *unforeseen* scenario, the average *AUA*. of *Normal+SIWCon* is 10.60% higher than *Normal* method (17.45% vs. 6.85%). Under the *foreseen* scenario, SIWCon can further improve the robustness of models, with the average *AUA*. of *AT+SIWCon* being 7.35% higher than that of *AT* (40.98% vs. 33.63%). These results demonstrate the effectiveness of SIWCon and its potential to be combined with more training methods, using as a plug-and-play regularization.

**SIWCon makes words of different importance have a similar influence.** The results of word influence are illustrated in Table 2. SIWCon similarizes the influence of information in words of different importance, as evidenced by the average $AOPC_{Diff.}$ of *Normal+SIWCon* being 0.017 lower than that of *Normal*, and of *AT+SIWCon* being 0.016 lower than that of *AT*. Recall the question we raised in section §1, the increased *AUA*. and decreased $AOPC_{Diff.}$ when using SIWCon in training empirically give an affirmative answer.

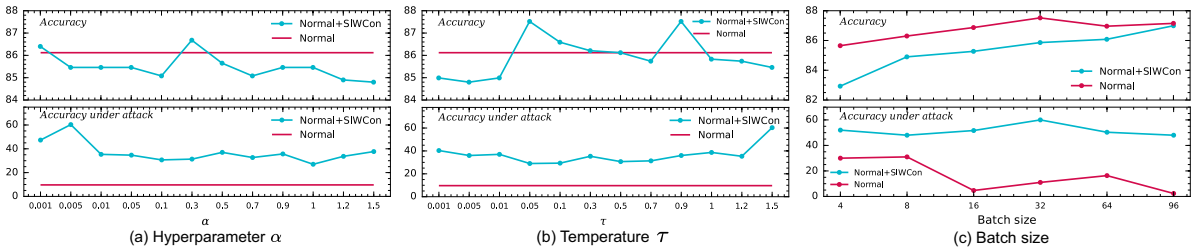
### 4.4 Further Analysis on SIWCon

In this section, we conduct further analysis and ablation study on BERT and MR dataset.

**Hyperparameter $\alpha$.** The influence of $\alpha$ is illustrated in Figure 3(a). We find that when $\alpha$ is set to different values, the robustness of BERT can always be effectively improved, as the *AUA*. of *Normal+SIWCon* is always higher than that of *Normal*. When $\alpha$ is small, BERT tends to be more robust. Different values of $\alpha$ also have a slight impact on the clean accuracy, as the *ACC*. of *Normal+SIWCon* is always close to that of *Normal*.

**Temperature $\tau$.** The influence of $\tau$ is shown in Figure 3(b). Similar to $\alpha$, when $\tau$ is set to various values, the robustness of the model is consistently improved, while the *ACC*. fluctuates around that of the normally trained BERT. However, $\tau$ has a greater impact on the clean accuracy than $\alpha$.

**Batch Size.** The influence of batch size is shown in Figure 3(c). SIWCon is benefit from larger batch size. As the batch size increases, the gap in clean accuracy (*ACC.*) between the models trained with




(a) Hyperparameter $\alpha$ (b) Temperature $\tau$ (c) Batch size

Figure 3: Influence of the hyperparameter $\alpha$, temperature $\tau$, and batch size.

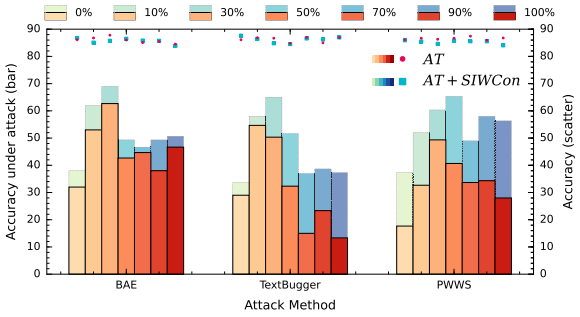



Figure 4: The comparisons of *ACC.* and *AUA.* under different adversarial training setting. *0%* denotes no adversarial examples are generated, and *AT* downgrades to *Normal*, which is the unforeseen scenario. Other ratios indicate the number of adversarial examples incorporated in training, which is the foreseen scenario.

and without SIWCon decreases, while the gap in robustness (*AUA.*) tends to increase. We conjecture that this is due to the contrastive nature of SIWCon regularization, as larger batch sizes provide more negative examples, thereby facilitating the regularizing (Chen et al., 2020a).

**Attack Methods and Examples Ratio.** We test the performance of SIWCon with more attack methods under different adversarial training settings, and the results are shown in Figure 4. We observe that SIWCon consistently outperforms the basic training method in terms of model robustness when using different attack methods. Additionally, we find that when a higher proportion of adversarial examples are incorporated into adversarial training, robustness may sometimes be reduced. However, SIWCon effectively mitigates this negative impact.

**Ablation Study.** We replace the data augmentation operations $t^{imp}(\cdot)$ and $t^{ump}(\cdot)$ in SIWCon with new augmentation operations that randomly drop out words in sentences to perform ablation study. The results in Table 3 show that the influence-based data augmentation operations used in SIWCon help the model (i) improve robustness, as *AUA.* of SIWCon are higher than the random methods, and (ii)

| | *ACC.* ↑ | *AUA.* ↑ | $AOPC_{Comp.}$ | $AOPC_{Suff.}$ | $AOPC_{Diff.}$ ↓ |
|---|---|---|---|---|---|
| | | | *DeepWordBug* | | |
| SIWCon | 85.46 | **60.33** | 0.030 | 0.015 | **0.015** |
| w/ random | **86.49** | 38.67 | 0.035 | 0.014 | 0.021 |
| | | | *TextFooler* | | |
| SIWCon | 84.31 | **30.67** | 0.038 | 0.022 | **0.016** |
| w/ random | **85.45** | 22.34 | 0.044 | 0.020 | 0.024 |

Table 3: Ablation study on BERT and MR. *w/ random* means the augmentations of each sentence are sampled randomly rather than based on attributions.

similarize the influence of the words of different importance on prediction, as $AOPC_{Diff.}$ of SIWCon are lower than the random methods.

### 4.5 Further Analysis on Model Behavior

**Loss Landscape.** Following the filter normalization scheme proposed by Li et al. (2018), we fine-tune BERT on the MR training set, and plot the loss landscape of BERT on the MR testing set, as shown in Figure 5. It is shown that the loss landscape of *Normal+SIWCon* (b) is visibly smoother and changes more slowly than the normally trained BERT (a). Furthermore, adversarial training (c) makes the loss landscape smoother than the *Normal* method (a), while when it is combined with SIWCon (d), the loss landscape is further smoothened. According to the finding of Mok et al. (2021) that a robust model should have a smooth loss landscape, the visualization results demonstrate that SIWCon is effective for improving model robustness.

**Sentence Representation.** We fine-tune BERT on MR and then, for a normal sentence, we generate two groups of sentences by *cumulatively* removing the 40% most and least important words in the sentence (e.g., abcd → ~~a~~bcd → ~~ab~~cd), following the gradient attribution (3). We also utilize PWWS (Ren et al., 2019) to generate adversarial examples from the normal sentence. The sentence representations visualized by t-SNE (van der Maaten and Hinton, 2008) and the reduction paths (Feng et al., 2018) are shown in Figure 6.

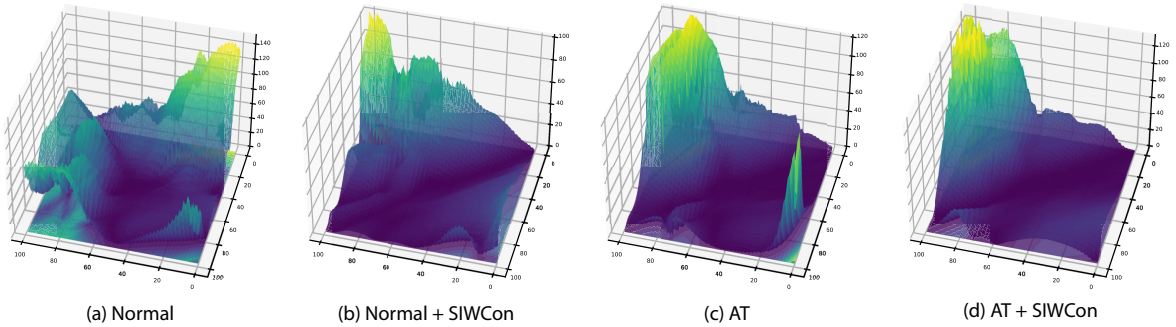



Figure 5: The loss landscape of BERT tuned with different methods.

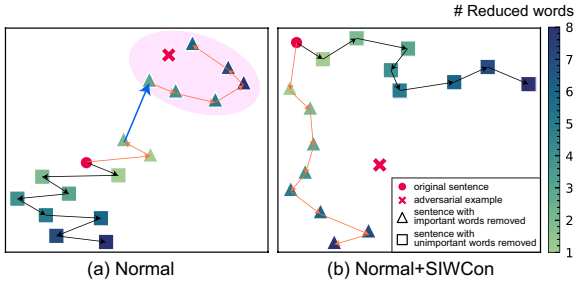



Figure 6: The visualization of sentence representations and reduction paths. The results are obtained from the MR instance *"A sports movie with action that's exciting on the field and a story you care about off it."* Darker examples indicate more words are removed. *Black* and *orange* arrows respectively illustrate the reduction path of unimportant and important words. *Blue* arrow highlights the reduction that drastically biases the prediction. Representations in *pink* area belong to the neighborhood of the adversarial example.

More results can be found in Appendix B.1.

The representation of the normal sentence (●) can be seen as a point with complete information for supporting the prediction contained, the bias of incomplete sentences (△ and □) from the normal sentence (●) can be seen as the information loss caused by word removal, and the location of the adversarial example indicates when how much information is lost, the example can no longer maintain the original prediction. When unimportant words are removed (□), the representations for both models are steadily biased from the normal sentence, and removal will not drastically bias the representations towards the adversarial example, indicating that the information in unimportant words is not influential on prediction. However, the two models behave differently when important words are removed (△). For Normal method, the representations are biased towards the adversarial example, and the prediction will be drastically biased when a few important words are removed (indicated by the *blue* arrow). For SIWCon, the representations are steadily biased in a similar manner as when unimportant words are removed, and the representations do not fall into the neighborhood of the adversarial example, indicating that important words are less influential on prediction and it is more difficult for attack methods to find adversarial examples.

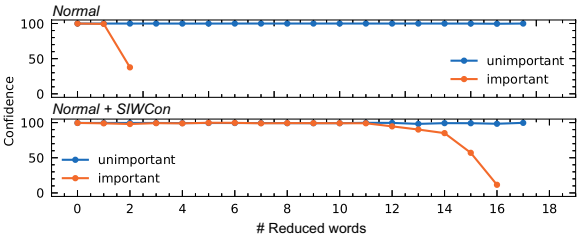



Figure 7: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance *"A high-spirited buddy movie about the reunion of Berlin anarchists who face arrest 15 years after their crime."*

**Confidence Changing.** We illustrate the change in model confidence with the removal of words on case instance in Figure 7. More results can be found in Appendix B.2. We cumulatively remove the most or least important words in a sentence, and the change in confidence can be seen as the influence of the information in the removed words. SIWCon reduces the influence of the information in important words, as more important words need to be removed to shift the model's prediction.

## 5 Conclusion

This paper presents SIWCon, a self-supervised regularization method based on contrastive learning. SIWCon improves the robustness of language models by encouraging the words of different importance to have more similar influence on prediction. Experiments show that SIWCon effectively improves model robustness without depending on adversarial perturbation. We hope the insights provided in this paper will inspire further research.

## Limitations

The loss objective of the proposed SIWCon regularization is computed on augmented data, which increases the time required for the model to complete training. We evaluate SIWCon on classification tasks, but it may be applied to various other tasks, such as reading comprehension and textual entailment. More evaluations are expected to be done in future works. The proposed SIWCon regularization is effective in defending against word-level adversarial attacks, as the basic elements of the augmentation methods are words. However, similar regularization techniques can also be applied to characters and sentences, and we leave evaluating the effectiveness of such variants in future works.

## Ethics Statement

In this paper, we propose a self-supervised regularization method for improving the model robustness, which does not need to learn from any adversarial examples. Since adversarial examples are always difficult to generate for language models, our method can thus reduce the financial and environmental cost of robustness improvement. Furthermore, our method forces models consider different words to have a similar degree of influence on prediction, potentially reducing the model's bias. All the datasets we use are publicly available, and we do not violate their licenses.

## Acknowledgements

The authors would like to thank the anonymous reviewers for their comprehensive and constructive comments. This research was supported by National Research and Development Program of China (No.2019YFB1005200).

## References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2016. Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP, Rep4NLP@ACL 2016*, pages 1–7.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. 2020b. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

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*, pages 4171–4186.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 31–36.

Hongchao Fang and Pengtao Xie. 2020. CERT: contrastive self-supervised learning for language understanding. *ArXiv preprint*, abs/2005.12766.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan L. Boyd-Graber. 2018. Pathologies of neural models make interpretation difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops, SP Workshops 2018*, pages 50–56.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 879–895.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pages 9726–9735.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. 2019. Achieving verified robustness to symbol substitutions via interval bound propagation. 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 2019,*, pages 4081–4091.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1875–1885.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. 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 4129–4142.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8018–8025.

Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. 2020. Interpretation of NLP models through input marginalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3154–3167.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 6391–6401.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure. *ArXiv preprint*, abs/1612.08220.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2018. Deep text classification can be fooled. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4208–4215.

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.

Jisoo Mok, Byunggook Na, Hyeokjun Choe, and Sungroh Yoon. 2021. Advrush: Searching for adversarially robust neural architectures. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pages 12302–12312.

John Morris, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020b. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis D. Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 171–186.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1069–1078.

Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. 2021. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *IEEE Conference on Computer Vision and Pattern Recognition, 2021*, pages 11205–11214.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 1085–1097.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, pages 2662–2670.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the visualization of what a deep neural network has learned. *IEEE Trans. Neural Networks Learn. Syst.*, 28(11):2660–2673.

Xuelin Situ, Ingrid Zukerman, Cécile Paris, Sameen Maruf, and Gholamreza Haffari. 2021. Learning to explain: Generating stable explanations fast. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021*, pages 5340–5355.

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.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, Proceedings of Machine Learning Research, pages 3319–3328.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings*.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv preprint*, abs/1807.03748.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Xiaosen Wang, Jin Hao, Yichen Yang, and Kun He. 2021. Natural language adversarial defense through synonym encoding. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 823–833.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: contrastive learning for sentence representation. *ArXiv preprint*, abs/2012.15466.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5065–5075.

Jin Yong Yoo, John Morris, Eli Lifland, and Yanjun Qi. 2020. Searching for a search method: Benchmarking search algorithms for generating NLP adversarial examples. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 323–332.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pages 818–833.

Pengwei Zhan, Yang Wu, Shaolei Zhou, Yunjian Zhang, and Liming Wang. 2022a. Mitigating the inconsistency between word saliency and model confidence with pathological contrastive training. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2226–2244.

Pengwei Zhan, Chao Zheng, Jing Yang, Yuxiang Wang, Liming Wang, Yang Wu, and Yunjian Zhang. 2022b. PARSE: an efficient search method for black-box adversarial text attacks. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022*, pages 4776–4787.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. 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 2019*, pages 4903–4912.

# A Additional Experimental Details

## A.1 Details on Dataset

MR contains movie reviews from Rotten Tomatoes, and the examples are labeled as positive or negative, with 8,530 for training and 1,066 for testing. SST2 contains sentences labeled as positive or negative, with 67,349 for training and 1,821 for testing. IMDB contains binary polar movie reviews from Internet Movie Database, which are also labeled as positive or negative, with 25,000 for training and 25,000 for testing.

## A.2 Details on Model

The experiments are conducted on three models with different architectures. The LSTM (Hochreiter and Schmidhuber, 1997) consists of a 300-dimensional GloVe embedding layer (Pennington et al., 2014), a Bi-LSTM layer with 150 hidden units, and a dense layer. The TextCNN is similar to the architecture in (Kim, 2014), while the embedding is also replaced with the 300-dimensional GloVe embedding. The BERT (Devlin et al., 2019) used in our experiment is the base uncased version.

## A.3 Details on Baseline

When SIWCon is combined with adversarial training, the overall objective is formulated as:

$$\min_\theta \ \mathcal{L}_{ce}(\boldsymbol{X}, Y) + \mathcal{L}_{ce}(\boldsymbol{X}^{adv}, Y) + \alpha \, \mathcal{L}_{SIWCon}(\boldsymbol{X}). \quad (11)$$

This joint training objective helps the model to learn both the normal and adversarial examples distribution and simultaneously regularizes the model on the word influence.

# B Additional Experimental Results

## B.1 Analysis on Sentence Representation

We give more visualizations of sentence representations and reduction paths in Figure 8-13. The instance sentences are randomly picked from MR dataset, the sentence representations are obtained on BERT, and the adversarial examples are generated by PWWS. Similar as the results in main text, darker examples indicate more words are removed. *Black* and *orange* arrows respectively illustrate the reduction path of unimportant and important words. *Blue* arrow highlights the reduction that drastically biases the prediction. Representations in *pink* area belong to the neighborhood of the adversarial example.

## B.2 Analysis on Confidence Changing

We provide more results on the change in model confidence with the removal of words in 14-17. The instance sentences are randomly picked from MR dataset, and the results are obtained on BERT.

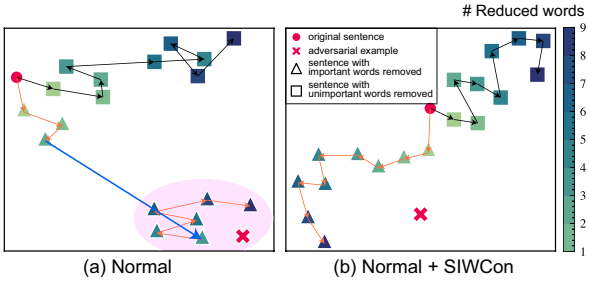



Figure 8: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance *"I enjoyed time of favor while I was watching it, but I was surprised at how quickly it faded from my memory."*

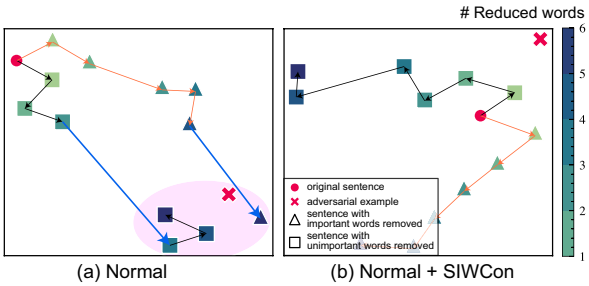



Figure 9: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance *"If nothing else, this movie introduces a promising, unusual kind of psychological horror."*

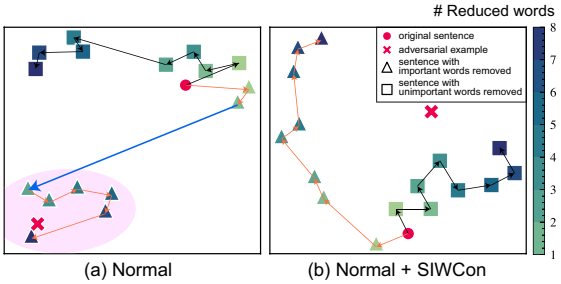



Figure 10: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance *"Everytime you think undercover brother has run out of steam, it finds a new way to surprise and amuse."*

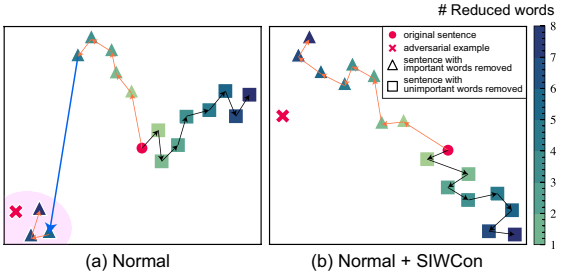



Figure 11: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance *"A real movie, about real people, that gives us a rare glimpse into a culture most of us don't know."*

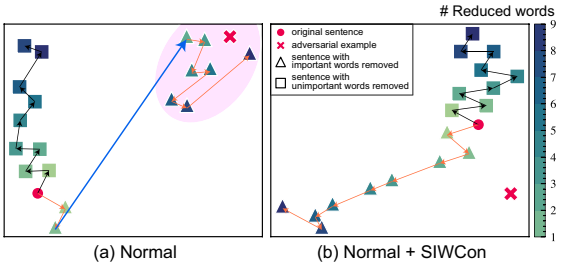



Figure 12: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance *"There's a lot of tooth in roger dodger. but what's nice is that there's a casual intelligence that permeates the script."*

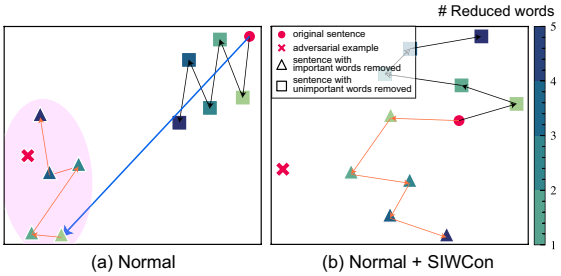



Figure 13: The visualization of sentence representations and reduction paths. The results are obtained on the BERT sentences representation of the MR instance *"This is the best American movie about troubled teens since 1998's whatever."*

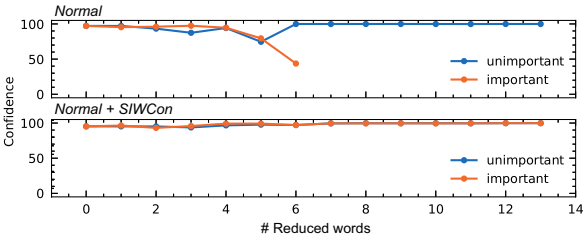



Figure 14: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance *"The entire movie has a truncated feeling, but what's available is lovely and lovable."*

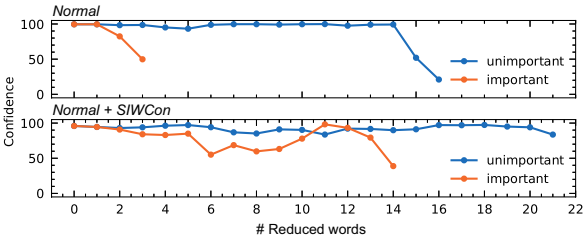



Figure 15: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance *"I enjoyed time of favor while i was watching it, but I was surprised at how quickly it faded from my memory."*

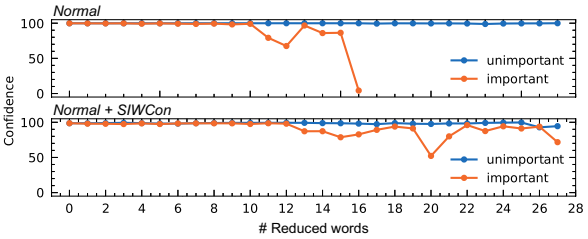



Figure 16: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance *"Some actors have so much charisma that you'd be happy to listen to them reading the phone book. Hugh grant and Sandra bullock are two such likeable actors."*

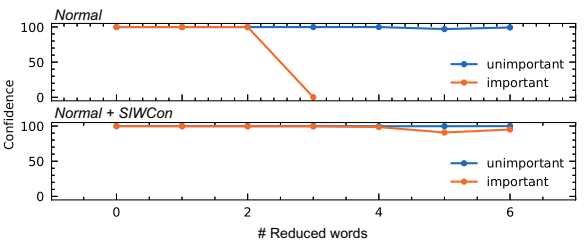



Figure 17: The change in model confidence with the removal of words until label shift. The results are obtained on the MR instance *"An engaging overview of Johnson's eccentric career."*

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*In Section Limitations.*

☑ A2. Did you discuss any potential risks of your work?
*In Section Ethics Statement.*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*In Abstract and Section 1.*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☑ Did you use or create scientific artifacts?

*In Section 3 and Section 4.*

☑ B1. Did you cite the creators of artifacts you used?
*In Section 1, Section 4.2, and Appendix A.*

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*In Section Ethics Statement.*

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*In Section Ethics Statement.*

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*The datasets we used in the paper are widely used benchmark datasets.*

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*In Section 4.2, Appendix A.1, and Appendix A.2.*

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*In Appendix A.1*

### C ☑ Did you run computational experiments?

*In Section 4.*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*In Section Limitations and Appendix A.2.*

---

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*In Section 4.2, Appendix A.*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*In Section 4.2.*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*In Section 4.2.*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*No response.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*No response.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*No response.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*No response.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*No response.*