Disentangled Information Bottleneck for Adversarial Text Defense

Anonymous ACL submission

Abstract

Adversarial text defense is a significant strategy to protect modern NLP models from being attacked. Typical text defense methods usually enhance the model's robustness by model retraining or equipping it with a data preprocessing step, aiming to eliminate the non-robust features and preserve the robust ones. Although some efforts have been made to recognize the robust features, e.g., by the information bottleneck (IB) technique, how to fully disentangle the robust and non-robust representation remains a big challenge. To alleviate this problem, we propose a novel text defense method, named Disentangled Information Bottleneck (DisIB), with two major merits. Firstly, we separate the robust features and non-robust features with a disentangled two-line framework rather than the one-line compression network in IB. This prevents the loss of robust features caused by information compression and produces **complete** robust features. Secondly, we design a discriminator network to approximate the minimum mutual information of the two lines, which sufficiently **disentangles** robust and non-robust features. To validate the effectiveness of our DisIB, we conduct a total of 96 defense experiments on four datasets by defending four popular attack methods. Experimental results elaborate that our method significantly outperforms six baselines, with accuracy improvements ranging from 3.8% to 20.7%.

1 Introduction

The Transformer-based deep learning frameworks have achieved milestone success in the Natural Language Processing (NLP) community, such as BERT (Devlin et al., 2019), T5 (Raffel et al., 2020), and ChatGPT (Wu et al., 2023). However, existing studies have proven that these deep models are super vulnerable to adversarial examples, which are slightly modified inputs (Raman et al., 2023). This phenomenon brings great risk to the security implementation of modern NLP tasks, including

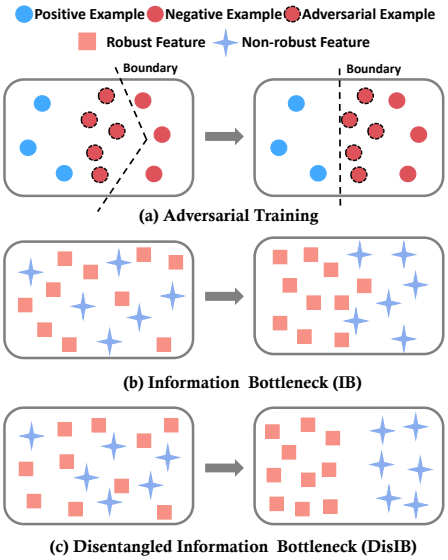


Figure 1: Principles of different defense methods. (a) Adversarial training pushes the sensitive decision boundary to be more tolerant. (b) IB incompletely disentangles non-robust and robust features. (c) DisIB completely disentangles non-robust and robust features.

text classification (Minaee et al., 2022), machine translation (Popel et al., 2020), language inference (Li et al., 2022), text generation (Yu et al., 2022), etc. How to design adversarial defense strategies to improve the robustness of deep models has become a significant research topic (Li et al., 2023).

According to whether the defender modifies the NLP model, existing text defense methods can be roughly categorized into (1) passive defense, which usually eliminates the adversarial perturbations with a data pre-processing step but does not change the victim model, and (2) active defense, which directly optimizes the model itself. In the first group, several data pre-processing operations have been proposed, such as spell-checking/correction (Li et al., 2019; Hládek et al., 2020), feature density detection (Yoo et al., 2022), AI-generated text decoder (Huang et al., 2024), to filter out adversarial

perturbations. The defense performance of these methods is highly dependent on the perturbation localization and recovery accuracy, and their defending scopes are usually limited to the defender-specified attackers.

In the second group, the defender aims to improve the victim model via network parameters or structure optimization. Specifically, the parameter optimization, e.g., adversarial training (Formento et al., 2024) and certified robust boundary training (Raman et al., 2023), extract robust features by moving the sensitive decision boundary to be more tolerant so that the robust feature space can be enlarged (see Figure 1 (a)). The model structure optimization targets to disentangle the robust and non-robust features, such as DiffusionBERT (He et al., 2023), multi-head confusion (MHC) (Le et al., 2022), and information bottleneck (IB) layer insertion (Zhang et al., 2022). Unlike DiffusionBERT and MHC who seek robust features with empirical strategies, the IB defines task-relevant word embeddings as robust features, showing better theoretical explainability and practical defense performance. Particularly, the IB constructs a one-line network structure to extract only robust features with an information compression layer. However, information compression usually pushes the robust features to the non-robust side (a.k.a., information loss), leading to incomplete feature separation and limited defense performance (see Figure 1 (b)).

In this work, we propose a novel method, i.e., Disentangled Information Bottleneck (DisIB), to extract complete yet fully disentangled robust features and improve the defense accuracy. Specifically, we design the supervised disentanglement strategy with two major merits. Firstly, we present a two-line defense framework, consisting of an encoder-decoder-based robust feature extraction line and an encoder-reconstructor-based non-robust feature extraction line to topologically solve the information loss problem. Secondly, we build a discriminator network to estimate the joint distribution probability of the two-line features and define a feature-disentangle loss function to minimize the mutual information between the two lines. This reduces the overlap between the robust features and non-robust features and improves the degree of feature disentanglement (see Figure 1 (c)). Owing to the relatively **complete** and fully **disentangled** robust features, the NLP model can make more accurate decisions even if the input text is perturbed.

In summary, our contributions are as follows.

- We construct a two-line adversarial text defense framework, dubbed DisIB, to disentangle robust and non-robust features. The two-line topology structure can naturally prevent information loss caused by the compression operation as in IB, which ensures the extracted robust feature is relatively **complete**.

- We design a discriminator to estimate the joint distribution probability of the two-line features and define a feature-disentangle objective function to minimize overlapping information between them, which fully **disentangles** the robust features from non-robust ones.

- We evaluate the effectiveness of our DisIB by comparing it with six typical baselines with totally 96 defense experiments. Qualitative and quantitative experiments demonstrate the superiority of our algorithm in both feature disentanglement and defense performance (with 3.8% to 20.7% accuracy improvements).

2 Related Works

This section briefly reviews the typical text defense methods, including passive and activedefense.

Passive defense methods do not change the victim model but often equip it with a data preprocessing step to eliminate adversarial perturbations. For example, (Li et al., 2019) employed a context-aware spelling check service to defend character-level attacks. (Yoo et al., 2022) developed a perturbation detection method against word-level attacks based on feature density estimation. To defend both character-level and word-level attacks, (Gupta et al., 2023) trained a model capable of intercepting and rewriting adversarial inputs. (Huang et al., 2024) proposed SCRN, which employs a reconstruction network to add and remove noise from the text. These methods usually design perturbation location strategy and recovery method according to a specific attacker, so their generalization ability are relatively limited for unseen attacks.

Active defense approaches directly optimize the victim model by retraining network parameters or reconstructing network structure. Typical parameter optimization strategies include adversarial training and certified robustness. Adversarial training was primarily proposed in image domain by (Goodfellow et al., 2015), which joins adversarial examples to the training set and retrains the model. Subsequently, various improved adversarial train-

ing methods have been proposed and successfully applied in text defense field. For example, (Zhu et al., 2020) proposed FreeLB, which minimizes the adversarial loss across different regions around input samples by adding adversarial perturbations to word embeddings. (Li and Qiu, 2021) employs a token-level accumulated perturbation vocabulary with a normalization sphere constraint for better perturbation initialization. (Formento et al., 2024) proposed Semantic Robust Defense (SemRoDe), which minimizes the distance between the base and adversarial domains, thereby aligning the two domains and producing a smooth decision boundary. In general, adversarial training also relies on existing attackers to generate adversarial samples, and as the number of adversarial samples increases, the model performance will be gradually reduced on the clean data. The certified robustness techniques provide theoretical guarantees of robustness. (Shi et al., 2020) derived the robustness boundary of models under Transformer architecture by boundary propagation techniques. (Moon et al., 2023) combined randomized smoothing (RS) with masked inference (MI) to smooth decision boundary and denoise adversarial perturbations. The proof process of robust boundary is constrained by various factors, e.g., model structure and optimization method, so their application scope is often limited.

Model structure optimization methods learn robust features by empirically or theoretically changing certain layers of the victim model. (Le et al., 2022) modified and retrained the last layer with multi-expert heads to confuse the attackers. (He et al., 2023) combined the diffusion model with BERT to enhance the denoising ability. Based on information bottleneck theory (Tishby and Zaslavsky, 2015), (Wang et al., 2021) presented an Information Bottleneck regularizer and an Anchored Feature regularizer to extract robust features. (Zhang et al., 2022) inserted an Information Bottleneck (IB) layer into BERT to compress non-robust features and capture robust features relevant to the task. Recently, (Zhao et al., 2024) proposed disentangled text representation learning (DTRL), which extracts robust features through a task classifier and non-robust features via an adversarial example dependent classifier. Generally, theoretical methods show better theoretical explainability and practical performance than empirical defense, but they still meet the information loss and incomplete feature disentanglement problems.

3 Algorithm

In this section, we first review the most related baseline IB in §3.1 and then discuss the details of the proposed DisIB in §3.2. Figure 2 shows the model framework of our DisIB.

3.1 Information Bottleneck

Let $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ denotes the training set, where $x_i \in X$ represents input and $y_i \in Y$ represents output. The IB method adds an additional layer to the original network, aiming to compress X into a robust variable R while retaining enough information required to predict Y. This can be achieved by minimizing the objective function below:

$$\mathcal{L}_{IB} = -I(Y;R) + \beta I(X;R), \tag{1}$$

where $\beta \in [0,1]$ balances the compression and prediction, and $I(A;B)$ denotes the mutual information between variables A and B:

$$I(A;B) = \sum_{a\in A}\sum_{b\in B} p(a,b)\log\left(\frac{p(a,b)}{p(a)p(b)}\right), \tag{2}$$

The larger mutual information indicates a stronger association. After optimization, the robust feature R can be more relevant to task prediction Y and less relevant to input X. However, related studies have indicated that compression inevitably leads to information loss (Pan et al., 2021), and the incomplete robust feature R is hard to guarantee a satisfactory defense accuracy.

3.2 Disentangled Information Bottleneck

To avoid information loss, we propose a two-line framework to disentangle robust and non-robust features. Specifically, we introduce an additional variable $N = \{n_i\}_{i=1}^{N}$ as non-robust features to complement robust features $R = \{r_i\}_{i=1}^{N}$. Then we adjust the objective function $\mathcal{L}_{DisIB}$ as:

$$\mathcal{L}_{DisIB} = -I(Y;R) - I(X;N,Y) + I(N;R). \tag{3}$$

Similar to Eq. (1), maximizing $I(Y;R)$ ensures that the robust feature R contains enough information to predict the task-relevant output Y. Different from Eq. (1), we design two novel items $I(X;N,Y)$ and $I(N;R)$, where the former reconstructs X guided by (N,Y) so that non-robust feature N covers the Y-irrelevant information of X, and the latter minimizes the overlap between R and N. This disentangled two-line structure is significant in avoiding information loss, and the $I(N;R)$

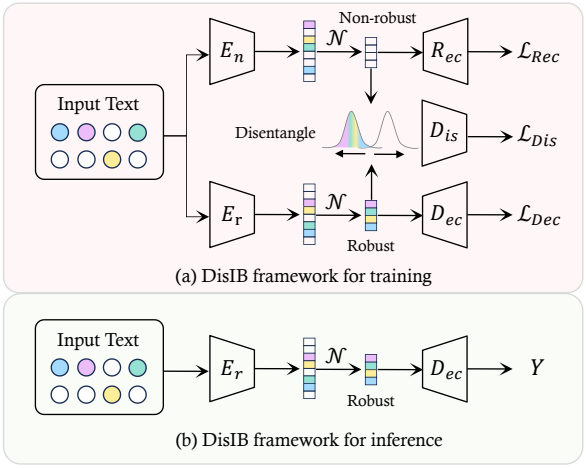


Figure 2: The framework of the proposed DisIB. In the training stage (a), we construct a two-line network to separately extract the robust and non-robust features to avoid information loss. Then the information overlap between the two lines is reduced with a discriminator D_{is}. In the inference stage (b), the optimized robust feature is employed to make correct predictions.

helps to sufficiently separate robust and non-robust features. Next, we will introduce how to optimize the three items in Eq. (3).

In Eq. (3), the calculation of mutual information is challenging due to the complexity of the joint and marginal distributions. Therefore, we derive variational approximations to $I(Y;R)$ and $I(X;N,Y)$ terms by applying the classical variational Bayesian strategy (Barber and Agakov, 2004), which allows for effective estimation of the variational lower bound without requiring a large number of samples:

$$\begin{aligned} I(Y;R) &= \mathbb{E}_{p(y,r)} \log p(y|r) - \mathbb{E}_{p(y)} \log p(y) \\ &\geq \mathbb{E}_{p(y,r)} \log q(y|r) + H(Y), \end{aligned} \tag{4}$$

$$\begin{aligned} I(X;N,Y) &= \mathbb{E}_{p(x,n,y)} \log p(x|n,y) \\ &\quad - \mathbb{E}_{p(x)} \log p(x) \\ &\geq \mathbb{E}_{p(x,n,y)} \log q(x|n,y) + H(X), \end{aligned} \tag{5}$$

where $q(y|r)$ and $q(x|n,y)$ represent variational probabilistic mappings. $H(Y)$ and $H(X)$ denote the information entropy of Y and X, respectively. Then, we decompose the joint distribution into multiple conditional probability distributions by Markov chain $Y \leftrightarrow X \leftrightarrow R$ and $Y \leftrightarrow X \leftrightarrow N$, thereby simplifying the computation process:

$$p(y,r) = \mathbb{E}_{q_{data}(x)} q_{data}(y|x) p(r|x), \tag{6}$$

$$p(x,n,y) = \mathbb{E}_{q_{data}(x)} q_{data}(y|x) p(n|x), \tag{7}$$

where $q_{data}(x)$ denotes the statistics probability distribution of x in the training data. $q_{data}(y|x)$ is variational posterior mappings of y, $p(r|x)$ and $p(n|x)$ can be viewed as robust and non-robust extractors. By substituting Eq. (6) and Eq. (7) into Eq. (4) and Eq. (5) and dropping constants $H(X)$ and $H(Y)$, we can calculate $I(Y;R)$ and $I(X;N,Y)$:

$$\begin{aligned} I(Y;R) &\geq \mathbb{E}_{p(y,r)} \log q(y|r) \\ &= \mathbb{E}_{q_{data}(x)} \mathbb{E}_{q_{data}(y|x)} \mathbb{E}_{p(r|x)} \log q(y|r), \end{aligned} \tag{8}$$

$$\begin{aligned} I(X;N,Y) &\geq \mathbb{E}_{p(x,n,y)} \log q(x|n,y) \\ &= \mathbb{E}_{q_{\text{data}}(x)} \mathbb{E}_{q_{\text{data}}(y|x)} \mathbb{E}_{p(n|x)} \log q(x|n,y). \end{aligned} \tag{9}$$

Since $H(N)$ and $H(R)$ depend on the unfixed probabilistic distributions of N and R, they are no longer constants. Therefore, the above method cannot compute the $I(N;R)$ term. To address this problem, we derive $I(N;R)$ by the Kullback-Leibler distance between joint distribution $p(n,r)$ and the product of marginal distribution $p(n)p(r)$:

$$\begin{aligned} I(N;R) &= D_{KL}[p(n,r) \parallel (p(n)p(r)] \\ &= \mathbb{E}_{p(n,r)} \log \left[\frac{p(n,r)}{p(n)p(r)} \right]. \end{aligned} \tag{10}$$

However, $p(n,r)$ and $p(n)p(r)$ are hard to estimate due to the dependence between n and r. Therefore, we utilize density-ratio-trick (Nguyen et al., 2007; Sugiyama et al., 2012; Kim and Mnih, 2018) to directly calculate the ratio $S(n,r) = \frac{p(n,r)}{p(n)p(r)}$ with three steps. Firstly, we sample x from dataset and sample $p(n,r)$ from $p(n,r|x) = p(n|x)p(r|x)$ by Markov chain $N \leftrightarrow X \leftrightarrow R$. Secondly, we shuffle the sample of $p(n,r)$ along the batch axis to reduce the correlation between n and r to sample $p(n)p(r)$ (Belghazi et al., 2018). Finally, we utilize a discriminator to estimate the probability, i.e., $D_{is}(n,r)$. So $1 - D_{is}(n,r)$ approximate the probability of input from $p(n)p(r)$, the ratio can be calculated:

$$S(n,r) = \frac{D_{is}(n,r)}{1 - D_{is}(n,r)} \tag{11}$$

3.2.1 Training stage

In the training phase, we train the two-line DisIB framework as shown in Figure 2 (a), including a robust feature learning line and a non-robust feature extraction line.

Robust Feature Extraction (Line 1) is implemented through a Transformer-based encoder-decoder network, as this structure is well-fit the

mainstream NLP models. Specifically, given the input x_i, the output of encoder E_r, i.e., $E_r(x_i)$, is first processed with a normal sampling step $\mathcal{N}$. This step can be considered as a simply optimized variational approximation of robust features space to learn the robust bottleneck representation r_i:

$$r_i = \mathcal{N}(E_r(x_i), \sigma_r^2) \tag{12}$$

where $\mu_r = E_r(x_i)$ and σ_r are the mean and standard variance of sampling, respectively.

Subsequently, the variable r_i is fed into a decoder D_{ec}, which produces probabilities over possible outcomes y_i. This parameterizes the variational probabilistic mapping $q(y_i|r_i)$, so maximizing Eq. (8) is equivalent to minimizing the cross-entropy loss of the decoder $\mathcal{L}_{D_{ec}}$:

$$\mathcal{L}_{D_{ec}}(D_{ec}(r_i), y_i) = -\log D_{ec}(r_i)_{y_i}. \tag{13}$$

After training the encoder-decoder network branch, the feature r_i contains a high amount of information relevant to the task prediction y_i.

Non-robust Feature learning (Line 2) is designed as an encoder-reconstructor network, where the encoder E_n and the normal sampling $\mathcal{N}$ (with standard variance σ_n) generate a bottleneck non-robust representation n_i:

$$n_i = \mathcal{N}(E_n(x_i), \sigma_n^2). \tag{14}$$

Different from Line 1, we designed a reconstructor R_{ec} in Line 2, which takes the concatenated (n_i, y_i) as input and generates corresponding reconstruction $x_i^{'}$ to parameterize variational probabilistic mappings $q(x|n, y)$. So the reconstruction loss can be utilized to implement Eq. (9):

$$\mathcal{L}_{R_{ec}}(R_{ec}(n_i, y_i), x_i) = ||R_{ec}(n_i, y_i) - x_i||_2^2. \tag{15}$$

Maximizing the mutual information between x_i and $x_i^{'}$, the feature n_i at least covers all task prediction irrelevant information from x_i, which is the non-robust feature.

Disentangle the two lines. We involve a discriminator D_{is} to eliminate overlapping information between n_i and r_i, which takes the concatenated input (n_i, r_i) and outputs the probability that the input originates from joint distribution $p(n, r)$ rather than the product of marginal distribution $p(n)p(r)$. We train the discriminator with the feature-disentangle loss $\mathcal{L}_{D_{is}}$:

$$\begin{aligned}\mathcal{L}_{Dis(n_i,r_i)} = \min_{p} \max_{D_{is}} [&\mathbb{E}_{p(n)p(r)} \log D_{is}(n_i, r_i) \\ &+ \mathbb{E}_{p(n,r)} \log(1 - D_{is}(n_i, r_i))],\end{aligned} \tag{16}$$

Algorithm 1 Disentangled Information Bottleneck

Input : Training set $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$;
Output : Encoders E_r, E_n, Decoder D_{ec}, Reconstructor R_{ec} and Discriminator D_{is};

1: **while** not converge **do**
2: Select batch $\{x_i, y_i\}$ randomly
3: Extract r_i and n_i by Eq. (12) and Eq. (14)
4: Calculate discriminator train loss $\mathcal{L}_{Dis(n_i,r_i)}$ by Eq. (16)
5: Update discriminator D_{is}
6: Calculate total loss $\mathcal{L}_{DisIB}$ by Eq. (17)
7: Update E_r, E_n, D_{ec} and R_{ec}
8: **end while**
9: **return** E_r, E_n, D_{ec}, R_{ec}, and D_{is}.

which maximizes the output $D_{is}(n_i, r_i)$ and minimizes the corresponding probability distribution to train the discriminator D_{is}. Ultimately, the overall loss function is:

$$\mathcal{L}_{DisIB} = \mathcal{L}_{D_{ec}} + \mathcal{L}_{R_{ec}} - \mathcal{L}_{Dis(n_i,r_i)}, \tag{17}$$

We train the two-line framework by minimizing the total loss Eq. (17) and train the discriminator by minimizing Eq. (16) to disentangle robust and non-robust features. The complete training procedure is given in Algorithm 1.

3.2.2 Inference stage

As a defender, the inference stage only needs the complete and sufficiently disentangled robust feature to make correct decisions. Therefore, only encoder E_r and decoder D_{ec} remain active during inference as shown in Figure 2 (b).

4 Experiments

We provide the **SOURCE CODE** and **experiment settings** in the supplementary materials to ensure all the results in this section are reproducible.

4.1 Datasets

We conduct our experiments on four public datasets. AG's News Corpus (AG News) (Zhang et al., 2015) is a four-class news genre classification task. The Stanford Sentiment Treebank (SST-2) (Socher et al., 2013), Movie Reviews (MR) (Pang and Lee, 2005), and Internet Movie Database (IMDB) (Maas et al., 2011) are sentiment analysis tasks with binary classification. The average sentence lengths are 39, 17, 19, and 238 words, respectively.

	Methods	*Acc%*↑	TextFooler			TextBugger			Deepwordbug			PWWS		
			AUA%↑	*ASR%*	*Query*↑	*AUA%*↑	*ASR%*↓	*Query*↑	*AUA%*↑	*ASR%*↓	*Query*↑	*AUA%*↑	*ASR%*↓	*Query*↑
AG News	BERT	94.8	18.0	81.0	335.9	43.1	54.5	182.3	34.6	63.5	110.0	38.9	58.9	358.2
	TAVAT	95.3	44.2	53.6	443.5	52.7	44.7	235.2	51.1	46.4	117.1	53.3	44.1	374.3
	InfoBERT	94.6	28.2	70.2	348.7	41.4	56.2	166.0	36.9	61.0	95.3	38.1	59.7	353.4
	DTRL	94.6	69.5	26.3	511.8	64.1	32.1	269.2	64.4	31.9	124.3	77.0	19.0	396.0
	RSMI	94.3	61.7	34.6	498.9	64.3	31.8	275.9	61.8	34.7	126.8	78.4	16.9	397.4
	IB	95.1	70.8	25.7	515.3	62.7	33.6	293.7	59.4	37.6	117.9	73.5	23.0	389.3
	IB+FreeLB	95.0	76.0	19.9	540.3	71.6	24.7	312.2	64.0	32.7	122.9	81.6	14.0	402.3
	Ours	94.5	78.0	17.5	543.6	76.2	19.4	330.9	69.3	26.7	127.3	80.6	14.9	399.4
	Ours+FreeLB	**95.5**	**87.7**	**8.1**	**582.3**	**85.5**	**10.5**	**352.1**	**77.5**	**18.9**	**132.5**	**88.4**	**7.4**	**408.5**
SST-2	BERT	91.6	5.6	93.9	93.2	27.5	70.0	47.7	17.1	81.3	33.5	12.9	85.9	135.4
	TAVAT	90.9	14.4	84.2	113.3	37.5	58.8	61.7	27.8	69.4	37.1	20.1	77.9	138.1
	InfoBERT	92.1	15.0	83.7	94.4	37.3	59.5	44.2	27.0	70.7	29.7	21.0	77.2	131.6
	DTRL	88.7	17.7	80.1	120.6	34.0	61.2	66.0	26.5	70.0	37.7	26.8	69.1	139.7
	RSMI	86.1	14.4	82.8	123.3	31.5	63.7	57.9	25.9	70.2	38.3	24.2	71.9	145.5
	IB	91.5	24.2	73.6	131.5	40.0	56.3	68.0	31.0	66.1	39.3	32.4	64.5	145.9
	IB+FreeLB	92.3	23.9	74.3	132.7	40.1	56.9	65.6	33.0	64.5	39.5	31.7	65.8	144.8
	Ours	91.2	29.1	67.2	150.1	43.9	50.6	79.6	40.4	54.5	45.9	36.0	59.5	148.1
	Ours+FreeLB	**92.5**	**45.7**	**50.6**	**166.0**	**51.9**	**43.9**	**105.6**	**57.7**	**37.6**	**45.6**	**51.3**	**44.4**	**152.8**
MR	BERT	83.9	8.7	89.6	116.9	31.3	62.7	55.7	18.8	77.6	40.5	16.0	80.9	149.6
	TAVAT	**85.7**	12.7	85.2	116.7	30.8	64.1	56.0	23.4	72.7	39.3	19.2	77.6	149.4
	InfoBERT	68.4	5.5	92.0	108.5	26.6	61.1	47.1	7.4	89.2	37.6	13.4	80.4	150.6
	DTRL	82.7	13.1	84.2	118.8	25.4	68.9	69.3	20.4	75.4	40.5	21.5	73.9	152.3
	RSMI	82.3	14.4	82.5	135.1	32.3	60.9	62.7	21.9	73.3	42.2	26.9	67.3	162.1
	IB	84.2	20.6	75.6	137.5	34.2	59.1	76.1	28.9	65.6	43.1	25.2	70.1	155.9
	IB+FreeLB	85.2	30.4	64.4	160.5	43.0	49.5	87.9	41.3	51.5	47.3	37.9	55.6	163.4
	Ours	84.2	31.4	62.8	168.7	40.7	51.6	110.2	49.6	41.0	48.8	43.0	48.9	163.6
	Ours+FreeLB	85.6	**42.4**	**50.5**	**190.9**	**47.0**	**45.1**	**122.1**	**55.4**	**35.3**	**52.2**	**51.6**	**39.7**	**172.5**
IMDB	BERT	92.2	1.2	98.7	730.7	9.0	90.2	592.8	32.8	64.4	340.3	1.8	98.1	1671.6
	TAVAT	94.8	55.6	41.4	2302.7	51.8	45.4	1388.6	61.8	34.8	640.0	30.6	67.7	1995.6
	InfoBERT	78.6	23.0	70.7	1749.5	5.0	93.6	687.5	35.4	55.0	506.4	16.0	79.6	2077.4
	DTRL	91.1	42.7	53.0	1824.4	39.2	56.7	1128.7	48.6	46.7	549.3	42.8	53.3	2123.9
	RSMI	91.6	48.1	47.7	1580.8	47.6	47.9	973.5	57.8	36.9	448.3	54.6	40.2	1738.8
	IB	93.8	57.4	38.4	2339.2	56.4	40.1	1431.9	65.2	30.6	635.3	53.2	43.3	2283.5
	IB+FreeLB	94.5	54.5	42.2	2248.7	47.6	49.8	1304.9	59.8	36.9	624.8	50.0	46.9	2248.5
	Ours	93.5	71.6	23.3	2644.0	70.2	25.0	1628.4	73.0	21.8	676.4	46.8	50.0	2116.8
	Ours+FreeLB	**95.0**	**84.6**	**11.0**	**2868.0**	**83.4**	**12.2**	**1786.7**	**84.6**	**11.1**	**716.8**	**62.6**	**34.1**	**2295.4**

Table 1: The Acc, AUA, ASR, and Query of several defense methods on four datasets under four attacks by protecting the BERT model. The best results are highlighted in bolded, and the second best results are denoted in underlined. The ↑ (↓) means higher (lower) is better.

4.2 Attack Methods

We assess the defense capability by defending four popular text attack methods, including word-level attacks TextFooler (Jin et al., 2020), PWWS (Ren et al., 2019), character-level attack Deepwordbug (Gao et al., 2018), and multi-level attack TextBugger (Li et al., 2019). All the attack experiments are conducted on the TextAttack (Morris et al., 2020).

4.3 Baselines and Victim Models

We evaluate the effectiveness of our DisIB by comparing it with six baselines, such as Vanilla BERT (Devlin et al., 2019), TAVAT (Li and Qiu, 2021), InfoBERT (Wang et al., 2021), DTRL (Zhao et al., 2024), RSMI (Moon et al., 2023), and IB (Zhang et al., 2022). Vanilla BERT denotes there is no defense, which is utilized as the baseline for all other defense methods. We employ these defense methods to protect three victim models, i.e., the fine-tuned BERT, RoBERTa (Liu et al., 2019) and DistilBERT (Sanh et al., 2019), which are publicly available from Huggingface[1].

4.4 Evaluation Metrics

The performance of defense algorithms is evaluated based on four metrics. (*i*) **Acc%** is the accuracy of clean samples. Effective defense methods should maintain the original accuracy. (*ii*) **AUA%** denotes the accuracy under attack. A robust model exhibiting a higher AUA%. (*iii*) **ASR%** indicates the attack success rate - robust defense methods will show a low ASR%. (*iv*) **Query** is the average number of attempts by an attacker to query the target model. Higher Query value indicates that the model is harder to attack.

4.5 Quantitative Results and Analysis

Table 1 lists the performance of our method and the baselines by protecting the BERT model. Overall,

[1]https://github.com/huggingface/transformers

	Methods	Acc%↑	TextFooler			Deepwordbug		
			AUA%↑	ASR%↓	Query↑	AUA%↑	ASR%↓	Query↑
RoBERTa	InfoBERT	95.1	34.0	64.3	401.7	43.0	54.8	109.6
	DTRL	82.5	61.9	33.2	486.5	52.1	43.6	109.4
	IB	**95.4**	75.5	21.0	532.6	59.6	37.3	120.0
	Ours	94.6	**75.9**	**19.8**	**535.2**	**67.3**	**28.8**	**126.2**
DistilBERT	InfoBERT	95.4	21.2	77.8	346.6	27.7	71.0	101.6
	DTRL	93.9	59.5	36.4	462.2	57.6	38.9	117.0
	IB	43.5	3.8	91.3	174.4	6.8	84.2	73.2
	Ours	**95.4**	**78.2**	**18.0**	**538.9**	**71.1**	**24.8**	**128.2**

Table 2: Defense performance comparison on different pre-trained models using AG News dataset.

Methods		PAIR ASR%↓	GCG ASR%↓	TriviaQA BAR%↑
Vicuna (7b-V1.5)	None	88	100	98
	IB	84	74	94
	Ours	**78**	**72**	**98**
LLaMA-2 (7b-chat-hf)	None	18	32	96
	IB	18	28	97
	Ours	**14**	**24**	**97**

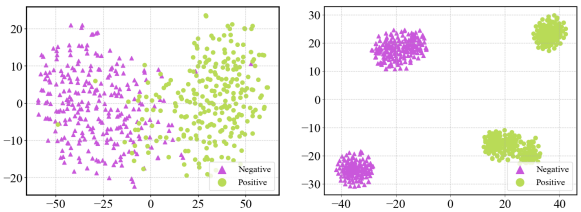
Table 3: Defense results on AdvBench.

our DisIB outperforms the baselines on most of the 96 defense experiments, with accuracy improvements ranging from 3.8% to 20.7%. Particularly, compared to the best baseline, our DisIB achieves an average of improvements on three robustness metrics, i.e., AUA (7.9%), ASR (7.8%), and Query (35.2). This indicates that our DisIB properly separates and eliminates the non-robust features added by both word-level and character-level attackers. Besides, the clean accuracy of our model on the AG News and SST-2 datasets is nearly equivalent to BERT, while on the MR and IMDB datasets, our model demonstrates superior accuracy. This means our method retains sufficient robust information for the model to make accurate predictions. Another good property is that our DisIB is compatible with traditional adversarial training methods, e.g., FreeLB (Zhu et al., 2020). As shown in Table 1, the Ours+FreeLB approach consistently outperforms all baselines in terms of robustness. We hope this result could shine new light on the direction of combined text defense.

Table 2 shows the defense results on RoBERTa and DistilBERT models. Our DisIB also attains the top-1 performance in most cases, which illustrates the good defense capability of our method across various classification models.

Defending LLM. This part evaluates the effectiveness of our DisIB in defending Large Language Model (LLM). Particularly, we replace the encoder in the robust feature extraction line with a LLM and utilize the LLM head as the decoder. We test our method on LLaMA-2 (Touvron et al., 2023) and Vicuna (Jain et al., 2023) against common jailbreak attacks, including GCG (Zou et al., 2023) and PAIR (Chao et al., 2023). We adopt AdvBench (Zou et al., 2023) as a harmful benchmark and generate 100 adversarial prompts with each attack method for training. To examine whether the defense methods refuse to answer benign prompts, we employ Benign Answering Rate (BAR) in the normal TriviaQA (Joshi et al., 2017) tasks. For evaluation, we employ the test set of SafeDecoding (Xu et al., 2024). As shown in Table 3, our method outperforms the baseline IB on both ASR and BAR. This indicates that our method can also defend against jailbreak attacks on LLM.

In addition, the **efficiency analysis** experiments, i.e., the inference runtime comparison, are listed in the supplementary materials Part B. The **parameter optimization** experiments are provided in supplementary material Part C, regarding the sampling standard deviation parameters, i.e., σ_r and σ_n.



(a) IB (b) DisIB

Figure 3: t-SNE visualization of robust features learned via (a) IB and (b) DisIB. Clearly, our DisIB can better separate the two labels.

4.6 Qualitative Results and Analysis

Feature Visualization. To validate the feature disentanglement capability of our DisIB, we randomly select 520 samples from the MR dataset and extract their robust features with both IB and DisIB. Figure 3 visualizes the labels of these robust features via t-SNE (Van der Maaten and Hinton, 2008). The results show that the robust features captured by our method completely separate the positive and negative labels by a large margin, while the IB cannot fully distinguish them. This indicates that *our method is superior to the IB in **disentangling** the robust and non-robust features*.

Sample Visualization. Except for the feature-

Methods			TextFooler			TextBugger			Deepwordbug			PWWS		
R_{ec}	D_{is}	E_n	*AUA%*↑	*ASR%*↓	*Query*↑	*AUA%*↑	*ASR%*↓	*Query*↑	*AUA%*↑	*ASR%*↓	*Query*↑	*AUA%*↑	*ASR%*↓	*Query*↑
✓	✓	✓	**29.3**	**67.0**	**144.9**	**43.8**	**50.7**	78.1	**40.0**	**15.9**	**41.8**	**36.4**	**58.9**	**145.9**
-	✓	✓	23.5	73.7	127.0	37.9	57.6	**91.9**	38.2	57.2	39.4	29.9	66.5	139.7
✓	-	✓	24.7	72.6	132.5	40.0	55.6	72.5	35.4	60.6	40.2	31.4	65.2	142.4
-	-	-	27.4	70.0	135.3	41.9	54.1	75.4	36.9	59.6	40.6	33.2	63.7	142.2

Table 4: Ablation studies. ✓ and - denotes with and without the corresponding module, respectively.

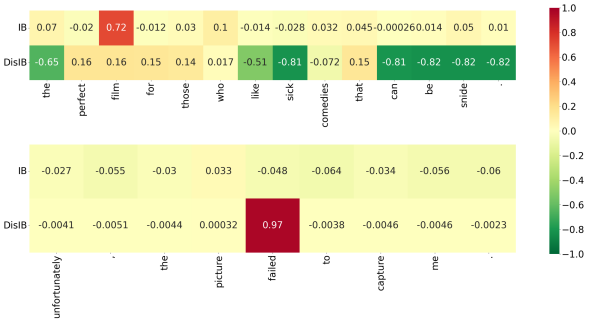


Figure 4: Visualisation of word significance. A higher value suggests that the word is more important in making predictions (robust features), whereas a smaller value indicates the word is less significant for models (non-robust features).

level visualization, we also make a more intuitive sample-level visualization to show the effectiveness of our DisIB in capturing task-relevant robust features. Specifically, we calculate the importance scores of individual words in the input sentence (Zhang et al., 2022). We conduct a series of normalized importance score calculations on the SST-2 dataset, and two examples are illustrated in Figure 4 (more examples can be found in the supplementary materials). In the first example, our DisIB extracts the important words, e.g., 'perfect', 'film', etc, while the IB only focuses on the word 'film' but ignores 'perfect', which carries a distinctly positive sentiment. In the second sentence, our method directly captures the word 'failed', which strongly indicates negative emotion. In contrast, the IB fails to identify this word and does not extract any useful information. The results demonstrate that our method can more effectively identify important words than IB. From these intuitive examples, we confirm that our proposed *DisIB is less likely to lose important information and extracts more* ***complete*** *robust features than IB*.

4.7 Ablation Study

We perform ablation studies to examine the effects of key components of our DisIB, including reconstructor (R_{ec}), discriminator (D_{is}), and the entire non-robust feature extraction line ($R_{ec}+D_{is}+E_n$). Table 4 reports the ablation study results. From Table 4 we know that the removal of the reconstructor resulted in an average decrease of AUA by 5.0%, which illustrates the necessity of the $I(X;N,Y)$ term. Besides, the removal of the discriminator caused an average of 3.6% AUA reduction, indicating the significance of the feature disentanglement step. The elimination of the entire non-robust feature extraction line resulted in a 2.0% AUA decrease, demonstrating the necessity and superiority of the two-line defense framework.

5 Conclusion

In this work, we proposed a novel text defense method, i.e., Disentangled Information Bottleneck (DisIB), which improves the adversarial robustness of modern NLP models by disentangling robust and non-robust features. Specifically, the DisIB is a two-line framework, which contains an encoder-decoder robust feature extraction line and an encoder-reconstructor non-robust feature extraction line. A novel objective function has been devised with a discriminator network to minimize the mutual information of the two lines. Experimental results elaborate the superiorities of our method in defending against both classification models and Large Language Models (LLMs).

6 Limitation

We summarize the limitations of this work from two aspects. Firstly, the two-line framework results in high computational cost and memory cost especially for the LLM training stage. Therefore, how to improve the training efficiency, e.g., optimizing only the information bottleneck architecture and distinguishing safety layers to avoid unnecessary training, should be a potential research direction. Secondly, multilingual defense scenario, e.g., Chinese text defense, is not sufficiently explored. Future work could also focus on investigating the generalizability of the text adversarial defense across different languages.

References

David Barber and Felix Agakov. 2004. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

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.

Brian Formento, Wenjie Feng, Chuan-Sheng Foo, Anh Tuan Luu, and See-Kiong Ng. 2024. SemRoDe: Macro adversarial training to learn representations that are robust to word-level attacks. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 8005–8028.

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*, pages 50–56.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *stat*.

Ashim Gupta, Carter Wood Blum, Temma Choji, Yingjie Fei, Shalin Shah, Alakananda Vempala, and Vivek Srikumar. 2023. Don't retrain, just rewrite: Countering adversarial perturbations by rewriting text. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 13981–13998.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. 2023. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 4521–4534.

Daniel Hládek, Ján Staš, and Matúš Pleva. 2020. Survey of automatic spelling correction. *Electronics*, 9(10):1670.

Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. 2024. Are AI-generated text detectors robust to adversarial perturbations? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 6005–6024.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

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*, pages 8018–8025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1601–1611.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *International conference on machine learning*, pages 2649–2658.

Thai Le, Noseong Park, and Dongwon Lee. 2022. Shield: Defending textual neural networks against multiple black-box adversarial attacks with stochastic multi-expert patcher. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 6661–6674.

J Li, S Ji, T Du, B Li, and T Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*.

Linyang Li and Xipeng Qiu. 2021. Token-aware virtual adversarial training in natural language understanding. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 8410–8418.

Linyang Li, Demin Song, and Xipeng Qiu. 2023. Text adversarial purification as defense against adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 338–350.

Shuang Li, Xuming Hu, Li Lin, and Lijie Wen. 2022. Pair-level supervised contrastive learning for natural language inference. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8237–8241.

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

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

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2022. Deep learning–based text classification: a comprehensive review. *ACM Comput. Surv.*, 54(3):62:1–62:40.

Han Cheol Moon, Shafiq Joty, Ruochen Zhao, Megh Thakkar, and Chi Xu. 2023. Randomized smoothing with masked inference for adversarially robust text classifications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 5145–5165.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. 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.

XuanLong Nguyen, Martin J Wainwright, and Michael Jordan. 2007. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. *Advances in neural information processing systems*, 20.

Ziqi Pan, Li Niu, Jianfu Zhang, and Liqing Zhang. 2021. Disentangled information bottleneck. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 9285–9293.

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

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature communications*, 11(1):1–15.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Mrigank Raman, Pratyush Maini, J Kolter, Zachary C Lipton, and Danish Pruthi. 2023. Model-tuning via prompts makes nlp models adversarially robust. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9266–9286.

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*, pages 1085–1097.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Zhouxing Shi, Huan Zhang, Kai Wei Chang, Minlie Huang, and Cho Jui Hsieh. 2020. Robustness verification for transformers. In *8th International Conference on Learning Representations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. 2012. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64:1009–1044.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop*, pages 1–5.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. Infobert: Improving robustness of language models from an information theoretic perspective. In *9th International Conference on Learning Representations*.

Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A brief overview of chatgpt: The history, status quo and potential future development. *IEEE CAA J. Autom. Sinica*, 10(5):1122–1136.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. 2024. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5587–5605.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of adversarial examples in text classification: Benchmark and baseline via robust density estimation. In *Findings of the Association for Computational Linguistics*, pages 3656–3672.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2022. A survey of knowledge-enhanced text generation. *ACM Comput. Surv.*, 54(11s):227:1–227:38.

Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. Improving the adversarial robustness of nlp models by information bottleneck. In *Findings of the Association for Computational Linguistics*, pages 3588–3598.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Jiahao Zhao, Wenji Mao, and Daniel Dajun Zeng. 2024. Disentangled text representation learning with information-theoretic perspective for adversarial robustness. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:1237–1247.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *8th International Conference on Learning Representations*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.