Subspace Defense: Discarding Adversarial Perturbations by Learning a Subspace for Clean Signals

Rui Zheng[1]*, Yuhao Zhou[1]*, Zhiheng Xi[1], Tao Gui[2†], Qi Zhang[1†], Xuanjing Huang[31†]
School of Computer Science, Fudan University[1]
Institute of Modern Languages and Linguistics, Fudan University[2]
International Human Phenome Institutes (Shanghai)[3]
{rzheng20, tgui, qz, xjhuang}@fudan.edu.cn, zhouyh21@m.fudan.edu.cn

Abstract

Deep neural networks (DNNs) are notoriously vulnerable to adversarial attacks that place carefully crafted perturbations on normal examples to fool DNNs. To better understand such attacks, a characterization of the features carried by adversarial examples is needed. In this paper, we tackle this challenge by inspecting the subspaces of sample features through spectral analysis. We first empirically show that the features of either clean signals or adversarial perturbations are redundant and span in low-dimensional linear subspaces respectively with minimal overlap, and the classical low-dimensional subspace projection can suppress perturbation features out of the subspace of clean signals. This makes it possible for DNNs to learn a subspace where only features of clean signals exist while those of perturbations are discarded, which can facilitate the distinction of adversarial examples. To prevent the residual perturbations that is inevitable in subspace learning, we propose an independence criterion to disentangle clean signals from perturbations. Experimental results show that the proposed strategy enables the model to inherently suppress adversaries, which not only boosts model robustness but also motivates new directions of effective adversarial defense.

Keywords: Robustness, Adversarial Training, Subspace

1. Introduction

Pre-trained language models (PLMs) are excellent feature extractors that map discrete inputs into fixed-length representations, which are then fed into a classifier for downstream tasks (Devlin et al., 2019; Liu et al., 2019). Despite their great success, PLMs have been proven to be vulnerable to adversarial examples generated by placing perturbations on clean inputs (Garg and Ramakrishnan, 2020; Zhang et al., 2020). Adversarial perturbations are often imperceptible to human, but can induce models to make erroneous predictions (Goodfellow et al., 2015; Ren et al., 2019). Extensive researches have shown that adversarial vulnerabilities are prevalent in various NLP tasks, raising security issues in practical applications (Wallace et al., 2019; Zhang et al., 2021; Lin et al., 2021; Chen et al., 2023; Zheng et al., 2023a).

Developing an understanding of the properties of adversarial perturbations and examples is a key requirement for adversarial defense. Tsipras et al. (2019) and Ilyas et al. (2019) argue that the non-robust parts of the features (those that generalize well but are brittle) can be manipulated by attackers to generate adversarial examples. Moreover, the adversarial examples lie in low-probability data regions (not naturally occurring) and close to (but not on) clean data submanifold (Szegedy et al., 2014; Tanay and Griffin, 2016). Akhtar et al. (2018) further emphasize that adversarial perturbations on different data are highly correlated and redundant. There exists a low-dimensional region, and perturbations belonging to this region can fool the model when added to any data point (Bao et al., 2023). To summarize, the known properties of adversarial perturbations are: 1) they originate from non-robust features; 2) they push data away from (but are close to) the clean data submanifold; and 3) they are highly correlated and redundant.

A number of methods are proposed to eliminate the affects of adversarial perturbations (Zheng et al., 2023b). Adversarial training is one of the most reliable techniques that generate adversarial examples for the training process and optimize model parameters to improve robustness (Madry et al., 2018; Zhu et al., 2020). Allen-Zhu and Li (2021) point out that adversarial training works by purifying some dense and complex features, rather than removing non-robust features. The information bottleneck-based methods (Alemi et al., 2017; Fischer and Alemi, 2020; Kim et al., 2021) aim to filter out excessive and noisy information that may invite adversarial attacks by compressing the mutual information between inputs and representations. However, in high-dimensional feature spaces, the mutual information is calculated by approximate techniques with high cost.

* Equal contribution.
† Corresponding authors.

LREC-COLING 2024, pages 15410–15421

In this paper, we show that, if studied from a feature perspective, a significant difference between clean signals and adversarial perturbations can be observed. By applying principal component analysis (PCA) (Jolliffe and Cadima, 2016) to the features encoded by PLMs, we show that the features of either clean signals or adversarial perturbations are redundant and lie in low-dimensional linear subspaces respectively with minimal overlap. This suggests that adversarial perturbations can be suppressed by discarding features outside of the clean signal subspace. To verify this, we project the adversarial examples onto clean subspaces, which significantly improves the robustness of the model while maintaining satisfactory performance on the main task. Furthermore, as shown in Figure 2, we find that the clean subspace projector acts like a noise filter to eliminate the high feature magnitudes introduced by adversarial perturbations.

Based on the above analysis, we propose a new defense strategy, named subspace defense, which adaptively learns (with an auxiliary linear layer) a subspace for clean signals, where only features of clean signals exist while those of perturbations are discarded. The subspace defense layer aims to learn a low-dimensional linear structure for the features of clean signals to retain as many task-relevant features as possible, and therefore inevitably preserve irrelevant features. Therefore, we introduce the Hilbert-Schmidt independence criterion (HSIC) (Gretton et al., 2005) to ensure independence between preserved and discarded features, reducing the residual adversarial perturbations. Our key contributions are summarized as follows:

- We identify that, from a feature perspective, the features of either clean signals or adversarial perturbations are redundant and lie in low-dimensional linear subspaces respectively with minimal overlap, and the classical projection can suppress perturbation features outside the subspace of clean signals.

- We propose a new defense strategy, named subspace defense, which adaptively learns (with an auxiliary linear layer) a subspace for clean signals, where only features of clean signals exists while those of perturbations are discarded.

- We empirically show that our subspace defense strategy can consistently improve the robustness of PLMs. Subspace defense strategy can accelerate the robustness convergence of adversarial training, thus avoid lengthy training processes.

2. Related Work

2.1. Textual Adversarial Attack

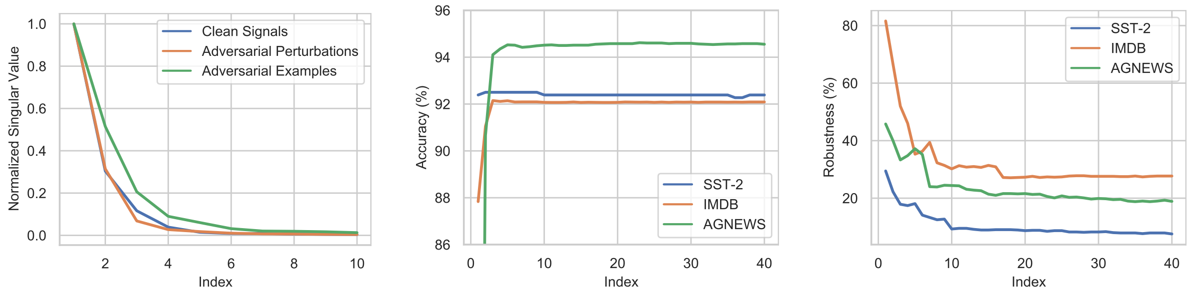
Text perturbation, unlike image attacks that operate in a continuous input space, needs to be performed discretly (Zhang et al., 2020; Wang et al., 2021b). Text attacks typically craft adversarial examples by deliberately manipulating characters (Ebrahimi et al., 2018; Gao et al., 2018), words (Ren et al., 2019; Jin et al., 2020; Li et al., 2020a; Alzantot et al., 2018; Zang et al., 2020; Maheshwary et al., 2021), phrases (Iyyer et al., 2018), or even the entire sentence (Wang et al., 2020). The word-level attacks, which are the most common ones, use the greedy algorithm (Ren et al., 2019) or combinatorial optimization (Alzantot et al., 2018) to search for the minimum number of substitutions. Moreover, these attacks all guarantee the fluency of adversarial examples from the perspective of semantics (Li et al., 2020a) or embedding space (Jin et al., 2020) to generate more stealthy adversarial examples.

2.2. Textual Adversarial Defense

In order to counter the adversarial attackers, a variety of defense methods are proposed to improve the robustness of the model (Zheng et al., 2023c). Adversarial training-based approaches generate adversarial examples during the training process and optimize the model to defend against them (Madry et al., 2018; Zhu et al., 2020; Liu et al., 2022). The information bottleneck approaches aim to filter out the redundant information contained in word embeddings and feature representations (Alemi et al., 2017; Fischer and Alemi, 2020; Kim et al., 2021). Zheng et al. (2022) prove that it is possible to extract robust subnetworks from the pre-trained model, and these subnetworks can be used for robust training as a robust alternative to the original model (**?**). In this paper, we eliminate non-robust and redundant features by projecting the perturbed features onto a low-dimensional subspace of clean signals. Compared with the information bottleneck-based approaches, our method is more efficient and has a stronger suppression effect against adversarial noise.

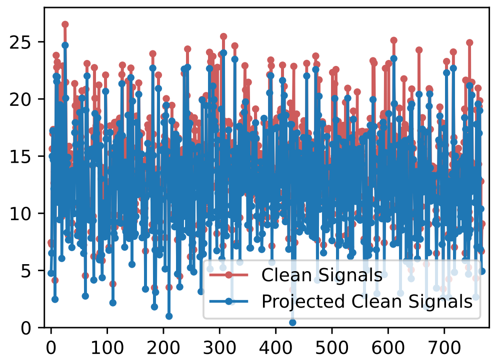
3. Spectral Analysis in Feature Space

In this section, we first show empirically that the features of either clean signals or adversarial perturbations are redundant and span in linear subspaces with minimal overlap between each other. We then prove that the classical projection can suppress perturbation features outside the subspace of clean signals.

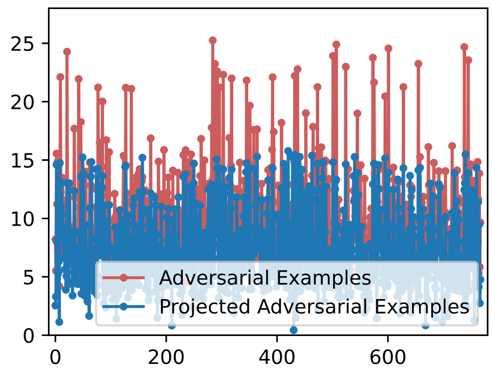


(a) Spectral Analysis (b) Accuracy on Projected Features (c) Robustness on Projected Features

Figure 1: (a) Spectral analysis of features of clean signals, adversarial perturbations, adversarial examples on SST-2. (b) and (c) respectively show the accuracy $(\%)$ and robustness evaluation (accuracy under TextFooler attack) after projecting the perturbed features on p-clean signal subspace.



(a) Clean Signals



(b) Adversarial Examples

Figure 2: Averaged feature magnitudes of clean signals, adversarial examples and their corresponding projected counterparts. Low-dimensional $(p = 2)$ clean subspace projector acts like a noise filter to eliminate the high feature magnitudes introduced by adversarial perturbations.

3.1. Threat Model

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with C classes, $\mathbf{x}_i$ denotes the **input embedding vector** and y_i is the label. Pre-trained language model $f(\cdot)$, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), is composed of a feature extractor $h(\cdot) : \mathbb{R}^{\mathcal{D}} \to \mathbb{R}^d$ and a linear classifier $g(\cdot) : \mathbb{R}^d \to \mathbb{R}^C$, where d is the feature size. In BERT-like models, the special `[CLS]` token at the top layer is used to aggregate contextual information and serve as feature representations. Then features are fed to a classifier for sentence-level tasks, such as sentiment analysis or natural language inference.

Adversarial attack. Consider a natural input $\mathbf{x}$ and a target threat model f, the adversarial attacker adds a small perturbation $\Delta\mathbf{x}$ to $\mathbf{x}$ such that: $\arg\max f(\mathbf{x} + \Delta\mathbf{x}) \neq y_{\text{true}}$, where $f(\mathbf{x} + \Delta\mathbf{x})$ is the wrong predication of $\mathbf{x} + \Delta\mathbf{x}$, and y_{true} denotes the true class label of $\mathbf{x}$. In practice, the perturbation $\Delta\mathbf{x}$ is often imperceptible to humans, and the perturbed input $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}$ that causes the model to misclassify $\mathbf{x}$ is called as an adversarial example. In the NLP domain, $\Delta\mathbf{x}$ consists of adding, removing or replacing a set of phrases, words or characters in the original text.

3.2. Spectral Analysis

The learned feature of $\mathbf{x}_i$ is $h(\mathbf{x}_i)$, after centralizing clean features (i.e. $\frac{1}{N}\sum_{i=1}^N h(\mathbf{x}_i) = 0$), the clean feature matrix can be denoted as $\mathbf{H} = [h(\mathbf{x}_1), h(\mathbf{x}_2), \ldots, h(\mathbf{x}_N)]^T \in \mathbb{R}^{N\times d}$ with $N \gg d$. We decompose the clean feature matrix $\mathbf{H}$ by singular value decomposition (SVD) (Horn and Johnson, 2012):

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{N\times N}$ and $\mathbf{V} \in \mathbb{R}^{d\times d}$ are orthogonal matrices, $\mathbf{\Sigma}$ is an $N \times d$ rectangular diagonal matrix with non-negative real numbers on the diagonal. The columns of $\mathbf{U}$ and $\mathbf{V}$ are called left and right singular vectors, respectively. The diagonal entries $\sigma_i = \mathbf{\Sigma}_{i,i}$ of $\mathbf{\Sigma}$ are uniquely determined by $\mathbf{H}$ and are known as the singular values with $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_n > 0$. The number of non-zero singular values is equal to the rank of $\mathbf{H}$. Similarly, the feature matrix of adversarial examples is $\mathbf{H}' = [h(\mathbf{x}'_1), h(\mathbf{x}'_2), \ldots, h(\mathbf{x}'_N)]^T \in \mathbb{R}^{N \times d}$, and the feature matrix of perturbations is denoted as $\Delta\mathbf{H} = \mathbf{H} - \mathbf{H}'$.

Singular value distributions. We plot the singular values of the clean, adversarial and perturbation feature matrices to visualize their differences. All features are extracted from the test data of SST-2 and their corresponding adversarial examples. As shown in Figure 1(a), the singular values of the natural and perturbation features decay faster than those of the adversarial features. This means that the natural examples and the adversarial perturbations lie in low-dimensional feature subspaces, which are smaller than the feature subspace of adversarial examples. More importantly, this indicates that there is little overlap between the clean feature subspace and the perturbation feature subspace. These analyses provide the possibility of projecting adversarial examples onto the feature subspace of clean signals to suppress the adversarial perturbations (Vaswani et al., 2018).

3.3. Subspace Projection

Keeping the first p principal components of the clean matrix $\mathbf{H}$, we use the first p right singular vectors to generate a projection onto the subspace of clean signals for adversarial example $\mathbf{x}'_i$:

$$\hat{h}_p(\mathbf{x}'_i) = \mathbf{V}_p \mathbf{V}_p^T h(\mathbf{x}'_i), \tag{2}$$

where $\mathbf{V}_p \in \mathbb{R}^{d \times p}$ consists of the first p right singular vectors, $\mathbf{V}_p \mathbf{V}_p^T \in \mathbb{R}^{d \times d}$ denotes the orthoprojector onto the p-dimension subspace $\mathcal{M}_p$. This formulation can be interpreted as preserving the maximum feature component that can be characterized in $\mathcal{M}_p$. We visualize the **accuracy** and **accuracy under attack** (for robust evaluation) of the projected adversarial features $\hat{h}_p(\mathbf{x}'_i)$.

Improving robustness. As shown in Figure 1(b), the accuracy is high even in the low-dimensional subspace, which is consistent with the sharp decrease in singular values for accuracy. In addition, the low-dimensional clean subspace projection can effectively improve the robustness of the model, while the adversarial subspace projection cannot. Therefore, by performing clean subspace projection with appropriate dimensionality, not only the performance on accuracy can be guaranteed, but also the effect of adversarial examples can be effectively reduced (although not completely avoided).

Figure 2 illustrates the averaged magnitudes of clean features, adversarial features, and their corresponding projected features. The feature magnitudes of adversarial examples are generally higher than that of clean examples, and there are outliers in some dimensions that deviate significantly from the clean features. Adversarial perturbation exhibits a significant distorting effect on the clean features, which leads to incorrect predictions of the classifier. As shown in Figure 2(b), the subspace projection can effectively narrow the magnitude gap between the features of clean and adversarial examples.

4. Proposed Method

Based on the above analysis, we introduce our subspace defense strategy, which dynamically learns a subspace in which only features of clean signals are preserved and features of perturbations are discarded.

4.1. Subspace Learning Module

Our previous results show that clean features span in a low-dimensional linear subspace. When projecting learned features into this clean subspace, it is possible to obtain both good generalization and robustness. Inspired by this, our goal is to learn an r-dimensional linear feature subspace for the clean examples and to remove redundant features that could be manipulated by the attacker.

In subspace learning module, we first apply a projection layer $\mathrm{Proj}(\cdot)$ to project the feature $h(\mathbf{x}_i) \in \mathbb{R}^d$ onto the r-dimensional subspace $h(\mathbf{x}_i) \to \hat{h}_r(\mathbf{x}_i) \in \mathbb{R}^r$, and then use the back-projection layer $\mathrm{Proj_b}(\cdot)$ to project $\hat{h}_r(\mathbf{x}_i)$ onto the original feature space $\hat{h}_r(\mathbf{x}_i) \to \hat{h}(\mathbf{x}_i) \in \mathbb{R}^d$. Formally, the final learned feature is:

$$\begin{aligned} \hat{h}_r(\mathbf{x}_i) &= \mathrm{Proj}\left(h(\mathbf{x}_i)\right), \\ \hat{h}(\mathbf{x}_i) &= \mathrm{Proj_b}(\hat{h}_r(\mathbf{x}_i)), \end{aligned} \tag{3}$$

where $\mathrm{Proj}(\cdot)$ and $\mathrm{Proj_b}(\cdot)$ are two linear layers defined as follows: $\mathrm{Proj}(h(\mathbf{x}_i)) = \mathbf{W}_1 h(\mathbf{x}_i) + \mathbf{b}_1$ and $\mathrm{Proj_b}(\hat{h}_r(\mathbf{x}_i)) = \mathbf{W}_2 \hat{h}_r(\mathbf{x}_i) + \mathbf{b}_2$ with $\mathbf{W}_1 \in \mathbb{R}^{r \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{d \times r}$, $\mathbf{b}_1 \in \mathbb{R}^r$ and $\mathbf{b}_2 \in \mathbb{R}^d$ are trainable parameters. Ideally, we would like to optimize the subspace learning module so that it maintains the underlying linear structure of the clean features. Thus, we consider the reconstruction loss:

$$\mathcal{L}_{recon} = \|h(\mathbf{x}_i) - \hat{h}(\mathbf{x}_i)\|_2^2. \tag{4}$$

By using subspace learning module, features outside the clean subspace (typically adversarial perturbations) will be discarded.

Comparison with Autoencoder. The autoencoder operates by receiving data, compressing and encoding the data, and then reconstructing the data from the encoded representation (Kramer, 1991; LeCun et al., 2015). The model is trained to minimize losses and the data is reconstructed as similar as possible. Through this process, the autoencoder learns important features of the data.

The goal of both the autoencoder and our proposed method is to determine which aspects of the input need to be preserved and which can be discarded. Our method differs from autoencoder in that: (1) We use linear layers as projectors, while autoencoder usually uses a nonlinear encoder and decoder; (2) An autoencoder obtains representations of samples from the input space, while our subspace learning module processes these representations to obtain a cleaner one.

4.2. Hilbert-Schmidt Independence Criterion

Subspace learning, such as PCA has the characteristic of being the optimal orthogonal transformation for keeping the subspace that has largest "variance". In Equation (2), the matrix $\mathbf{V}_p\mathbf{V}_p^T$ projects the features onto the p-dimensional subspace $\mathcal{M}_p$, while matrix $\mathbf{I}_p - \mathbf{V}_p\mathbf{V}_p^T$ denotes the projector onto the orthogonal complement space $\mathcal{M}_p^{\perp}$. Therefore, preserved features and discarded features are independent of each other. To achieve this property, we need to measure the degree of independence between the continues random variables $\hat{h}(\mathbf{x}_i)$ and $h(\mathbf{x}_i) - \hat{h}(\mathbf{x}_i)$ in high-dimensional spaces, and it is infeasible to rely on histogram-based measures. Thus, we chose to adopt the Hilbert-Schmidt Independence Criteria (HSIC) (Gretton et al., 2005).

For two random variables $\mathbf{u}$ and $\mathbf{v}$, $\mathrm{HSIC}(\mathbf{u},\mathbf{v})$ is the Hilbert-Schmidt norm of the cross-covariance operator between $\mathbf{u}$ and $\mathbf{v}$ in Reproducing Kernel Hilbert Space (RKHS). HSIC is able to capture non-linear dependencies between random variables, $\mathrm{HSIC}(\mathbf{u},\mathbf{v}) = 0$ if and only if $\mathbf{u} \perp \mathbf{v}$. Formally, $\mathrm{HSIC}(\mathbf{u},\mathbf{v})$ is defined as:

$$\begin{aligned}\mathrm{HSIC}(\mathbf{u},\mathbf{v}) = \;&\mathbb{E}_{\mathbf{uvu'v'}}[k(\mathbf{u},\mathbf{u}')k(\mathbf{v},\mathbf{v}')]\\ &+\mathbb{E}_{\mathbf{uu'}}[k(\mathbf{u},\mathbf{u}')]\mathbb{E}_{\mathbf{vv'}}[k(\mathbf{v},\mathbf{v}')]\\ &-2\mathbb{E}_{\mathbf{uv}}[\mathbb{E}_{\mathbf{u'}}[k(\mathbf{u},\mathbf{u}')]\mathbb{E}_{\mathbf{v'}}[k(\mathbf{v},\mathbf{v}')]],\end{aligned}$$

where $\mathbf{u}'$, $\mathbf{v}'$ are independent copies of $\mathbf{u}$, $\mathbf{v}$, and k is the radial basis function (RBF) kernel.

In practice, the finite-sample estimates of $\mathrm{HSIC}(\mathbf{u},\mathbf{v})$ are used for statistical testing (Gretton et al., 2007), feature similarity measures (Kornblith et al., 2019), and model regularization (Quadrianto et al., 2019). The unbiased estimator of $\mathrm{HSIC}(\mathbf{u},\mathbf{v})$ with n samples can be defined as (Song et al., 2012):

$$\begin{aligned}\mathrm{HSIC}(\mathbf{u},\mathbf{v}) = \;&\frac{1}{n(n-3)}\Big[tr(\tilde{\mathbf{u}}\tilde{\mathbf{v}}^T)\\ &+\frac{\mathbf{1}^T\tilde{\mathbf{u}}\mathbf{1}\mathbf{1}^T\tilde{\mathbf{v}}^T\mathbf{1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1}^T\tilde{\mathbf{u}}\tilde{\mathbf{v}}^T\mathbf{1}\Big],\end{aligned}$$

where $\tilde{\mathbf{u}}_{ij} = (1-\delta_{ij})k(\mathbf{u}_i,\mathbf{u}_j)$ and $\tilde{\mathbf{v}}_{ij} = (1-\delta_{ij})k(\mathbf{v}_i,\mathbf{v}_j)$, i.e., the diagonal entries of $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{v}}$ are set to zero.

To ensure the preserved features $\hat{h}(\mathbf{x}_i)$ and discard features $h(\mathbf{x}_i)-\hat{h}(\mathbf{x}_i)$ are independent of each other, we have:

$$\mathcal{L}_{hsic} = \sum_{i=1}^{n} \mathrm{HSIC}\left(\hat{h}(\mathbf{x}_i), h(\mathbf{x}_i) - \hat{h}(\mathbf{x}_i)\right). \quad (5)$$

Note that the reconstructed feature $\hat{h}(\mathbf{x}_i)$ is then fed to the classifier for downstream tasks.

4.3. Model Training

The subspace defense module can be used as an auxiliary component of the network and can be trained using standard training or different types of adversarial training. Here, we take the original adversarial training as an example and define loss functions to train simultaneously the network and our subspace defense module:

$$\mathcal{L}_{all-at} = \mathcal{L}_{adv} + \mathcal{L}_{recon} + \lambda \cdot \mathcal{L}_{hsic}, \quad (6)$$

where $\lambda > 0$ is the balancing hyperparameter, $\mathcal{L}_{adv} = \mathcal{L}_{ce}(\mathbf{x}', \boldsymbol{\theta})$ is the adversarial loss, and the adversarial example $\mathbf{x}'$ is generated by:

$$\mathbf{x}' = \arg \max_{\|\mathbf{x}'-\mathbf{x}\|_F \le \epsilon} \mathcal{L}(\mathbf{x}', y, \boldsymbol{\theta}), \quad (7)$$

where $\|\mathbf{x}' - \mathbf{x}\|_F \le \epsilon$ denotes the Frobenius normalization ball centered at $\mathbf{x}$ with radius ϵ.

A wide range of attack methods have been proposed for the crafting of adversarial examples. For example, PGD iteratively perturbs normal example $\mathbf{x}$ for a number of steps K with fixed step size η. If the perturbation goes beyond the ϵ-ball, it is projected back to the ϵ-ball (Madry et al., 2018):

$$\mathbf{x}_i^k = \prod \left(\mathbf{x}_i^{k-1} + \eta \cdot \mathrm{sign}(\nabla_{\mathbf{x}} \ell(\mathbf{x}_i^{k-1}, y_i, \theta))\right),$$

where $\mathbf{x}_i^k$ is the adversarial example at the k-th step, $\mathrm{sign}(\cdot)$ denotes the sign function and $\prod(\cdot)$ is the projection function.

5. Experiments

In this section, we conduct several experiments to demonstrate the effectiveness of our method over multiple NLP tasks such as text classification. We first compare our method against five competitive baselines in terms of accuracy on clean datasets and robust evaluation. Then, we perform an ablation study to confirm the importance of adversarial loss objective. We use widely adopted $\mathrm{BERT}_{\mathrm{BASE}}$ as the backbone model which is implemented by Huggingface Transformers[1] (Wolf et al., 2020) library.

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

Dataset	Method	Clean%	BERT-Attack		TextFooler		TextBugger	
			Aua%	#Query	Aua%	#Query	Aua%	#Query
IMDB	Fine-tune	92.1	4.0	907.6	5.3	701.7	15.0	563.3
	PGD	92.1	9.0	1886.5	17.8	1331.4	36.2	799.3
	FreeLB	92.0	27.9	2528.0	34.7	1724.7	51.6	1079.4
	InfoBERT	92.1	28.9	2581.8	36.0	1744.9	53.4	1096.5
	RobustT	91.8	69.9	3262.2	71.0	2112.8	71.9	1139.9
	Ours	**92.3**	**78.6**	**4275.6**	**78.6**	**2673.7**	**83.5**	**1749.5**
AGNews	Fine-tune	94.7	4.1	412.9	14.7	306.4	40.0	166.2
	PGD	**95.0**	20.9	593.2	36.0	399.2	56.4	193.9
	FreeLB	**95.0**	19.9	581.8	33.2	396.0	52.9	201.1
	InfoBERT	94.4	11.1	517.0	25.1	374.7	47.9	193.1
	RobustT	94.9	21.8	617.5	35.2	415.6	49.0	206.9
	Ours	93.8	**38.6**	**744.1**	**49.3**	**448.1**	**60.1**	**219.7**
SST-2	Fine-tune	92.1	3.8	106.4	6.1	90.5	28.7	46.0
	PGD	**92.2**	13.4	151.3	18.1	118.5	44.2	53.6
	FreeLB	91.7	23.9	174.7	29.4	132.6	49.7	53.8
	InfoBERT	92.1	14.4	162.3	18.3	121.4	40.3	51.2
	RobustT	90.9	20.8	169.2	28.6	149.8	43.1	53.9
	Ours	91.3	**36.5**	**201.2**	**46.3**	**167.3**	**54.5**	**62.3**

Table 1: Main results on adversarial robustness evaluation. The proposed method achieves a significant improvement of robustness compared with other baselines. The best performance is marked in bold.

5.1. Datasets

We consider three commonly used text classification datasets in our experiments: Stanford Sentiment Treebank of binary classification (SST-2) (Socher et al., 2013), AG News corpus (AGNews) (Zhang et al., 2015) and Internet Movie Database (IMDB) (Maas et al., 2011). The first two are binary sentiment analysis tasks that classify reviews into positive or negative sentiment, and the last is a classification task in which articles are categorized as world, sports, business or sci/tech.

5.2. Baselines

We compare our method against the basic fine-tuning method and five competitive adversarial defense methods in terms of accuracy on clean datasets and robust evaluation. (1) Fine-tune (Devlin et al., 2019): The official implementation for BERT on downstream tasks. (2) FreeLB (Zhu et al., 2020): An enhanced gradient-based adversarial training method which is not targeted at specific attack methods. (3) PGD (Madry et al., 2018): Projected gradient descent that formulates adversarial training algorithms into solving a min-max problem that minimizes the empirical loss on adversarial examples that can lead to maximized adversarial risk. (4) InfoBERT (Wang et al., 2021a): A learning framework for robust fine-tuning of PLMs from an information-theoretic perspective. (5) RobustT (Zheng et al., 2022): Robust Lottery Ticket Hypothesis finds the full PLM contains subnetworks, i.e., robust tickets, that can achieve a better robustness performance.

5.3. Attack Methods and Evaluation Metrics

Three well-received attack methods are adopted to evaluate our method against baselines. (1) BERT-Attack (Li et al., 2020b) generates adversarial samples using pre-trained masked language models exemplified by BERT, which can generate fluent and semantically preserved samples. (2) TextFooler (Jin et al., 2020) identifies the words in a sentence which is important to the victim model, and then replaces them with synonyms that are semantically similar and syntactically correct until the model's prediction for that sentence changes. (3) TextBugger (Li et al., 2019) generates misspelled words by using character-level and word-level perturbations. We use TextAttack[2] toolkit to implement these attack methods in adversatial attack experiments.

The evaluation metrics used in our experiments are shown bellow: Clean accuracy (**Clean%**) denotes the accuracy on the test dataset. Accuracy under attack (**Aua%**) represents the accuracy under adversarial attacks. Number of queries (**#Query**) refers to the average number of queries made by the attacker to the victim model. For the same attack method, models with higher robust-

[2]https://github.com/QData/TextAttack

Dataset	Method	Clean%	Aua%	
			TextFooler	TextBugger
QNLI	Fine-tune	**91.6**	4.7	10.5
	PGD	91.2	12.2	18.7
	FreeLB	90.5	12.8	12.0
	InfoBERT	91.5	16.4	20.9
	RobustT	91.5	17.0	25.9
	Ours	91.4	**33.5**	**43.1**
MNLI	Fine-tune	**84.4**	7.7	4.3
	PGD	83.9	14.5	15.7
	FreeLB	82.9	11.0	8.4
	InfoBERT	84.1	10.8	8.4
	RobustT	84.0	18.4	22.6
	Ours	84.2	**21.6**	**33.5**
QQP	Fine-tune	91.3	24.8	27.8
	PGD	91.2	32.0	33.5
	FreeLB	91.2	27.4	28.1
	InfoBERT	**91.9**	34.4	35.9
	RobustT	91.5	**47.2**	**46.0**
	Ours	91.6	34.1	35.3

Table 2: Adversarial robustness evaluation of the proposed method on QNLI, MNLI and QQP datasets.

ness are expected to have higher clean accuracy, accuracy under attack and number of queries in the robustness evaluation.

6. Results and Discussions

In this section, we illustrate the effectiveness of our approach and show the impact of the individual components in our approach on the model's robustness.

6.1. Main Results on Robustness Evaluation

From the results shown in Table 1, we can observe that: 1) The proposed approach achieves significant robustness improvements compared to other defense baselines. This is because the proposed method can remove perturbations by projecting feature representations onto a subspace of clean signals without compromising task accuracy. 2) Although InfoBERT can also filter out redundant information by compressing the mutual information between inputs and representations, its performance is far from satisfactory. This suggests that mutual information is difficult to optimize in a high-dimensional feature space and is not as effective as directly removing low-contributing features.

We also evaluate the performance of our proposed approach on other tasks, such as natural language inference and paraphrase identification. As seen in Table 2, our proposed method consistently improves the robustness of the model on the QNLI, MNLI, and QQP datasets.

Dataset	Method	Clean%	Aua%
IMDB	**Ours**	92.3	**78.6**
	w/o Projector	92.0	34.7
	w/o HSIC	**92.5**	41.3
	w/o Adv	92.2	45.5
AGNews	**Ours**	93.8	**49.3**
	w/o Projector	**95.0**	33.2
	w/o HSIC	93.9	37.3
	w/o Adv	93.5	41.0
SST-2	**Ours**	91.3	**43.6**
	w/o Projector	91.7	29.4
	w/o HSIC	**92.3**	26.5
	w/o Adv	92.1	36.4

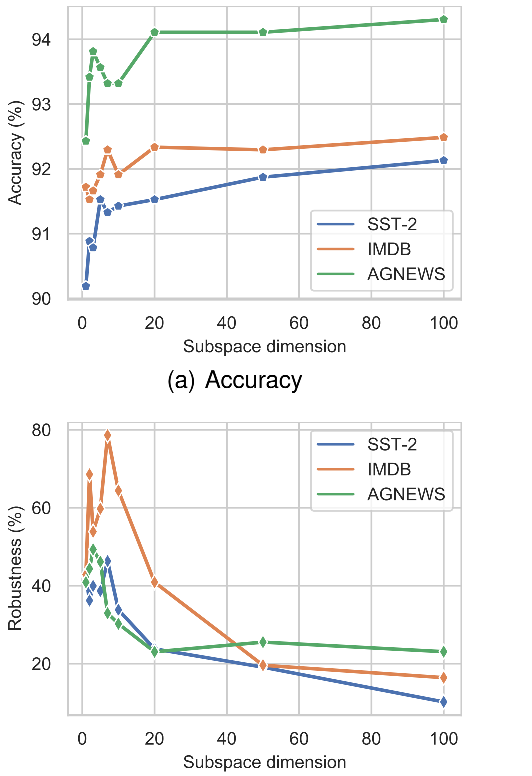
Table 3: Ablation study on text classification datasets. **Aua**% is obtained after using TextFooler attack.

6.2. Ablation Study

To illustrate the contribution of each component of our method, we perform the ablation study with the following components removed: the projector to clean signal subspace (projector), Hilbert-Schmidt independence criteria (HSIC), and adversarial examples (Adv). We can observe that: 1) The subspace projector is important for performance and the improvements in our approach come mainly from this component. 2) Both independence criteria and adversarial training can further help our approach to improve robustness, which also indicates that our approach can be well integrated with existing adversarial training.

6.3. Effect of Subspace Dimension

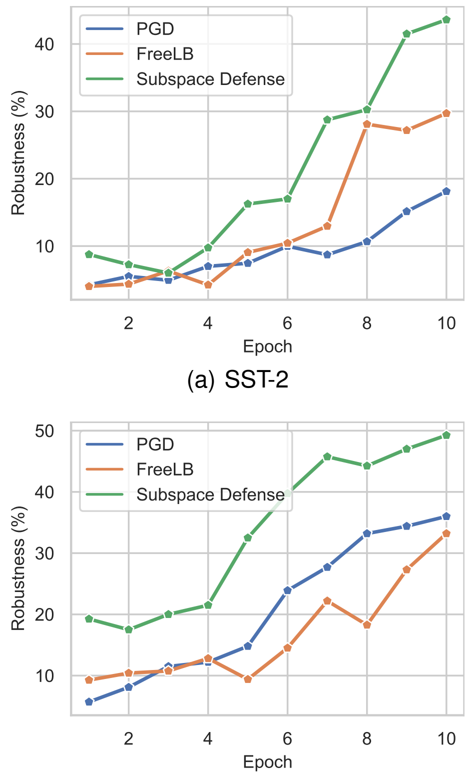
In Figure 3, we show the performance of our proposed method as the subspace dimension increases. When the dimension increases to a certain level, the task accuracy reaches its peak, which indicates that we need a suitable dimension to achieve good task performance and larger dimensions are useless. The robustness of the model can be maintained at a high level in low-dimensional subspaces until a certain threshold; beyond this threshold, the robustness deteriorates. The dimension controls the extent to which features are discarded, and no perturbations are discarded when the dimension is too large. When the subspace dimension is between 5 and 10, our proposed method can achieve a win-win result in



(a) Accuracy

(b) Robustness

Figure 3: Accuracy and robustness evaluation (accuracy under TextFooler attack) of model under different subspace dimension, both of which reach the peak when the subspace dimension is between 5 and 10.



(a) SST-2

(b) AGNews

Figure 4: Robustness of each epoch throughout training on SST-2 and AGNews with different training strategy. Compared to adversarial training methods like PGD and FreeLB, the proposed subspace defense speeds up robust training and converges much faster in terms of accuracy under attack.

terms of accuracy and robustness.

6.4. Speedup Robust Training

An important property of the proposed method is to accelerate the convergence of the training process. The training curves in Figure 4 show that the proposed method converges much faster in terms of robustness on the SST-2 and AGNEWS datasets compared to adversarial training methods such as PGD and FreeLB. That is because learning the low-dimensional structure of clean signals is easier than learning to resist adversarial examples. The advantage in convergence speed will make our method easy to apply in practice.

6.5. Impact of Projector Structure

To better understand the impact of the structure of the autoencoder on performance, we compare the following structures: 1-layer linear MLP, 1-layer MLP $+$ ReLU, 2-layer linear MLP (applied in our method), and 2-layer MLP $+$ ReLU. Table 4 shows that all structures help to improve the robustness of the model. More layers are slightly better for autoencoder, but non-linearity is harmful. The non-linearity destroys the linear structure of the subspace, so that even subspaces in low dimensions may contain perturbations. The above results show that the robustness of the model can be significantly improved by adding a simple autoencoder to the existing model.

7. Conclusion

In this paper, we characterize the feature properties of both clean signals and adversarial perturbations via low-dimensional subspace projection. Further, we provide an initial intuition as to how subspace learning is an effective method for defending adversaries, which suggests that subspace projector eliminates feature magnitudes of adversarial perturbations. Further investigation in this direction may lead to new techniques for both adversarial attack and defense.

8. Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work

Structures	3 dimension		5 dimension		7 dimension		10 dimension	
	Clean%	**Aua%**	**Clean%**	**Aua%**	**Clean%**	**Aua%**	**Clean%**	**Aua%**
1-layer MLP	91.3	23.0	91.6	24.1	91.5	23.5	92.1	19.9
1-layer MLP + ReLU	91.9	12.6	91.2	23.5	91.2	23.9	91.5	17.1
2-layer MLP	**92.0**	**26.4**	**92.1**	**31.1**	**92.2**	**28.9**	**92.2**	**28.8**
2-layer MLP+ReLU	**92.0**	17.1	**92.1**	27.7	91.9	24.5	92.1	18.4

Table 4: Impact of projector structure. Four structures are compared across different subspace dimensions on SST-2 in terms of accuracy and accuracy under attack. Structure with more layers performs slightly better and non-linearity impairs the robustness.

was partially funded by National Natural Science Foundation of China (No. 62206057, 61976056, 62076069, 61906176), Shanghai Rising-Star Program (23QA1400200), Natural Science Foundation of Shanghai (23ZR1403500), Program of Shanghai Academic Research Leader under grant 22XD1401100, CCF-Baidu Open Fund, and CCF-Baichuan Fund. The computations in this research were performed using the CFFF platform of Fudan University.

9. Bibliographical References

Naveed Akhtar, Jian Liu, and Ajmal Mian. 2018. Defense against universal adversarial perturbations. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3389–3398. Computer Vision Foundation / IEEE Computer Society.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zeyuan Allen-Zhu and Yuanzhi Li. 2021. Feature purification: How adversarial training performs robust deep learning. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 977–988. IEEE.

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, Brussels, Belgium. Association for Computational Linguistics.

Rong Bao, Rui Zheng, Liang Ding, Qi Zhang, and Dacheng Tao. 2023. CASN:class-aware score network for textual adversarial detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 671–687, Toronto, Canada. Association for Computational Linguistics.

Xuanting Chen, Junjie Ye, Can Zu, Nuo Xu, Rui Zheng, Minlong Peng, Jie Zhou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. How robust is gpt-3.5 to predecessors? a comprehensive study on language understanding tasks. *arXiv preprint arXiv:2303.00293*.

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

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 (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Ian Fischer and Alexander A. Alemi. 2020. CEB improves model robustness. *Entropy*, 22(10):1081.

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, San Francisco, CA, USA, May 24, 2018*, pages 50–56. IEEE Computer Society.

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, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Arthur Gretton, Olivier Bousquet, Alexander J. Smola, and Bernhard Schölkopf. 2005. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory, 16th International Conference, ALT 2005, Singapore, October 8-11, 2005, Proceedings*, volume 3734 of *Lecture Notes in Computer Science*, pages 63–77. Springer.

Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf, and Alexander J. Smola. 2007. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 585–592. Curran Associates, Inc.

Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 125–136.

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, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

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, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8018–8025. AAAI Press.

Ian T Jolliffe and Jorge Cadima. 2016. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

Junho Kim, Byung-Kwan Lee, and Yong Man Ro. 2021. Distilling robust and non-robust features in adversarial examples by information bottleneck. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17148–17159.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Mark A Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

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, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020a. 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, Online. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. BERT-ATTACK: adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6193–6202. Association for Computational Linguistics.

Jieyu Lin, Jiajie Zou, and Nai Ding. 2021. Using adversarial attacks to reveal the statistical bias in machine reading comprehension models. In *Proceedings of the 59th Annual Meeting* *of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 333–342, Online. Association for Computational Linguistics.

Qin Liu, Rui Zheng, Bao Rong, Jingyi Liu, Zhi-Hua Liu, Zhanzhan Cheng, Liang Qiao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Flooding-X: Improving BERT's resistance to adversarial attacks via loss-restricted fine-tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5634–5644, Dublin, Ireland. Association for Computational Linguistics.

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

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13525–13533. AAAI Press.

Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. 2019. Discovering fair representations in the data domain. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8227–8236. Computer Vision Foundation / IEEE.

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 Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

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

Le Song, Alexander J. Smola, Arthur Gretton, Justin Bedo, and Karsten M. Borgwardt. 2012. Feature selection via dependence maximization. *J. Mach. Learn. Res.*, 13:1393–1434.

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, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Thomas Tanay and Lewis D. Griffin. 2016. A boundary tilting persepective on the phenomenon of adversarial examples. *CoRR*, abs/1608.07690.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2019. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Namrata Vaswani, Thierry Bouwmans, Sajid Javed, and Praneeth Narayanamurthy. 2018. Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery. *IEEE Signal Process. Mag.*, 35(4):32–55.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021a. Infobert: Improving robustness of language models from an information theoretic perspective. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. CAT-gen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146, Online. Association for Computational Linguistics.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu, Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiaoqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021b. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

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

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. 2020. Word-level textual adversarial attacking as combinatorial optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6066–6080, Online. Association for Computational Linguistics.

Wei Emma Zhang, Quan Z. Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Trans. Intell. Syst. Technol.*, 11(3):24:1–24:41.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Xinze Zhang, Junzhe Zhang, Zhenhua Chen, and Kun He. 2021. Crafting adversarial examples for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1967–1977, Online. Association for Computational Linguistics.

Rui Zheng, Shihan Dou, Yuhao Zhou, Qin Liu, Tao Gui, Qi Zhang, Zhongyu Wei, Xuanjing Huang, and Menghan Zhang. 2023a. Detecting adversarial samples through sharpness of loss landscape. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11282–11298, Toronto, Canada. Association for Computational Linguistics.

Rui Zheng, Bao Rong, Yuhao Zhou, Di Liang, Sirui Wang, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Robust lottery tickets for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2211–2224, Dublin, Ireland. Association for Computational Linguistics.

Rui Zheng, Zhiheng Xi, Qin Liu, Wenbin Lai, Tao Gui, Qi Zhang, Xuanjing Huang, Jin Ma, Ying Shan, and Weifeng Ge. 2023b. Characterizing the impacts of instances on robustness. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2314–2332, Toronto, Canada. Association for Computational Linguistics.

Rui Zheng, Zhiheng Xi, Qin Liu, Wenbin Lai, Tao Gui, Qi Zhang, Xuanjing Huang, Jin Ma, Ying Shan, and Weifeng Ge. 2023c. Characterizing the impacts of instances on robustness. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2314–2332. Association for Computational Linguistics.

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, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.