Detecting Adversarial Samples through Sharpness of Loss Landscape

Rui Zheng[1*], **Shihan Dou**[1*], **Yuhao Zhou**[1], **Qin Liu**[2], **Tao Gui**[3†],
Qi Zhang[1†], **Zhongyu Wei**[1], **Xuanjing Huang**[1], **Menghan Zhang**[3]
[1] School of Computer Science, Fudan University
[2] Viterbi School of Engineering, University of Southern California
[3] Institute of Modern Languages and Linguistics, Fudan University
{rzheng20,tgui,qz,xjhuang}@fudan.edu.cn, qliu4147@usc.edu
{shdou21,zhouyh21}@m.fudan.edu.cn

Abstract

Deep neural networks (DNNs) have been proven to be sensitive towards perturbations on input samples, and previous works highlight that adversarial samples are even more vulnerable than normal ones. In this work, this phenomenon is illustrated from the perspective of sharpness via visualizing the input loss landscape of models. We first show that adversarial samples locate in steep and narrow local minima of the loss landscape (*high sharpness*) while normal samples, which differs distinctly from adversarial ones, reside in the loss surface that is more flatter (*low sharpness*). Based on this, we propose a simple and effective sharpness-based detector to distinct adversarial samples by maximizing the loss increment within the region where the inference sample is located. Considering that the notion of sharpness of a loss landscape is relative, we further propose an adaptive optimization strategy in an attempt to fairly compare the relative sharpness among different samples. Experimental results show that our approach can outperform previous detection methods by large margins (average +6.6 F1 score) for four advanced attack strategies considered in this paper across three text classification tasks. Our codes are publicly available at https://github.com/ruizheng20/sharpness_detection.

1 Introduction

Despite the popularity and success of pre-trained language models (PLMs), they are vulnerable to textual adversarial attacks (Garg and Ramakrishnan, 2020; Zhang et al., 2020). These attacks are designed to generate semantically consistent and syntactically correct adversarial samples that can fool the model into making incorrect predictions (Ren et al., 2019; Maheshwary et al., 2021). Adversarial vulnerability raises concerns about the safe practice of NLP systems in a variety of tasks (Wallace et al., 2019; Zhang et al., 2021; Lin et al., 2021).

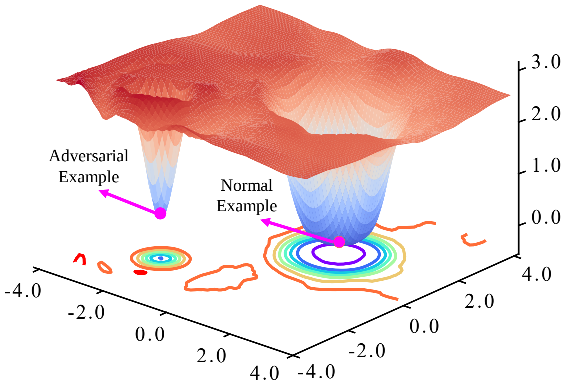


Figure 1: Input loss landscape of model with respect to normal and adversarial samples. Adversarial sample locates in a sharp local minimum on the input loss landscape, while the normal one resides in a wide area.

In machine learning, there are two main streams to counter adversarial attacks: adversarial detection and defense (Cohen et al., 2020). The purpose of detection is to distinguish the adversarial samples from the normal ones and discard them during the inference phase (Mozes et al., 2021; Yoo et al., 2022), while defense aims to predict the correct results of adversarial texts (Li et al., 2021b; Zheng et al., 2022; Omar et al., 2022; Liu et al., 2022b). The detect-discard strategy is an important step towards a robust model and can be integrated with existing defense methods. A significant challenge in adversarial detection is to explore an effective characteristic for recognition.

The existing state-of-the-art adversarial detection methods can be broadly classified into two categories: 1) perturbation-based methods (Mozes et al., 2021; Mosca et al., 2022; Wang et al., 2022) and 2) distribution-based methods (Yoo et al., 2022; Liu et al., 2022a). The perturbation-based methods assume that adversarial samples are more sensitive to perturbations in the input space than normal

[*] Equal contribution.
[†] Corresponding author.

samples. These methods are based on the model's reaction when the input words are perturbed by substitution (Mozes et al., 2021; Wang et al., 2021) or deletion (Mosca et al., 2022). However, these methods rely on empirically designed perturbations and it is difficult to find an optimal perturbation in the discrete text space. More importantly, no attempt has been made to explore why the sensitivity assumption is valid or to provide more details for this assumption.

We delve into the input loss landscape to characterize the model's sensitivities with respect to normal and adversarial samples. By visualizing the input loss landscape of the model, we observe a significant difference between the adversarial and normal samples: the loss surfaces on local minima with respect to adversarial samples are steep and narrow (*high sharpness*), while those of normal samples are much flatter (*low sharpness*). The above-mentioned significant distinction makes it eligible for distinguishing adversarial samples from normal ones. However, it remains a challenge on how to effectively measure the sharpness of an input loss landscape.

In this work, we formulate the sharpness calculation as a constrained optimization problem whose objective is to find a neighbor within the region where the inference sample is located to maximize the loss increment. The convergence quality of this constrained optimization problem can be assessed by "Frank-Wolfe gap" (i.e., the gap between the global optimum and the current estimate) (Frank and Wolfe, 1956; Lacoste-Julien, 2016). With this criterion, we find that samples tend to converge to different levels, which hinders a fair comparison of relative sharpness between samples (Dinh et al., 2017). Therefore, we design an adaptive optimization strategy that guides the solutions to converge gradually to the same level, thereby significantly improving the detection performance. Our contributions are as follows:

- We analyze the geometric properties of the input loss landscape. We reveal that the adversarial samples have a deep and sharp local minima on the input loss landscape.
- We propose a detection metric based on the sharpness of input loss landscape, which can be formulated as a constrained optimization problem.
- We design an adaptive optimization strategy to guide the calculation of sharpness to converge to the same level, which can further improve the detection performance.

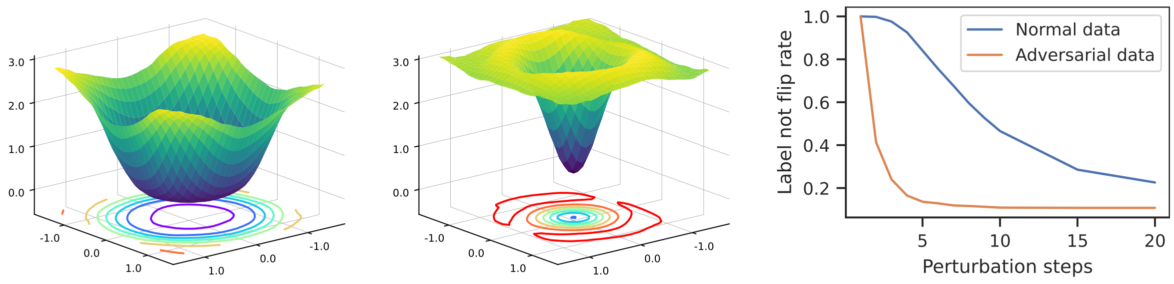
2 Related Work

2.1 Textual Adversarial Attack

Unlike image attacks that operate in a high-dimensional continuous input space, text perturbation needs to be performed in a discrete input space (Zhang et al., 2020). Text attacks typically generate adversarial samples by manipulating characters (Ebrahimi et al., 2018; Gao et al., 2018), words (Ren et al., 2019; Jin et al., 2020; Li et al., 2020; 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 most widely used word-level attacks use the greedy algorithm (Ren et al., 2019) and combinatorial optimization (Alzantot et al., 2018) to search for the minimum number of substitute words. Moreover, these attacks guarantee the fluency of adversarial samples in semantics (Li et al., 2020) or embedding space (Jin et al., 2020) to generate more stealthy adversarial samples. Recent studies have shown that most of the adversarial samples generated are of low quality, unnatural, and rarely appear in reality (Hauser et al., 2021; Wang et al., 2022).

2.2 Textual Adversarial Detection

Existing adversarial detection methods are mainly divided into two categories: 1) perturbation-based methods and 2) distribution-based methods. Zhou et al. (2019) propose a discriminator that learns to recognize word-level adversarial substitutions and then correct them. Yoo et al. (2022) assume that the representation distribution of original samples follows a multivariate Gaussian and use robust density estimation (Feinman et al., 2017) to determine the likelihood of a sentence being perturbed. Liu et al. (2022a) introduce the local intrinsic dimensionality (Ma et al., 2018) from image processing to text domain. Wang et al. (2022) apply the anomaly detector to identify unnatural adversarial samples and then use textual transformations to mitigate the adversarial effect. Mozes et al. (2021) find that word-level adversarial attacks tend to replace input words with less frequent ones, and exploit the frequency property of adversarial word substitutions to detect adversarial samples. Mosca et al. (2022) introduce a logits-based metric to capture the model's reaction when the input words are omitted. However, these methods rely on empirically



(a) Normal Sample

(b) Adversarial Sample

(c) Sensitivity to Perturbations

Figure 2: Difference between normal and adversarial samples. (a) Input loss landscape of normal samples; (b) Input loss landscape of adversarial samples. The adversarial samples' loss landscape has a **sharp** bottom, while that of normal samples is much flatter. (c) Label not flip rate, defined as the proportion of samples that keep the prediction results unchanged under adversarial perturbations. Normal samples are more robust in the face of adversarial perturbations than adversarial samples. Experimental results are obtained from BERT trained on AGNews, and adversarial samples are generated via BERT-Attack (Li et al., 2020).

designed word-level perturbations, making it difficult to find an optimal perturbation.

3 Delving into Input Loss Landscape

Our aim is to better understand adversarial samples, and thereby derive a potentially effective detector. In this section, we investigate the geometric properties of the input loss landscape and show a clear correlation between the sharpness of loss landscape and adversarial samples.

3.1 Visualizing Loss Landscape

Assume we have a PLM h with a loss function $\ell(\mathbf{x}^0, y)$, where $\mathbf{x}^0$ is the normal input text, y is the label and $h(\mathbf{x}^0)$ denotes the output logit. As the labels are unknown to the user in adversarial sample detection, we use the "predicated" label $y^* = \arg\max_y p(y|\mathbf{x}^0)$ in place of the golden label y. Following the visualization method proposed by (Goodfellow and Vinyals, 2015) and (Li et al., 2018), we project the high-dimensional loss surface into a 2D hyperplane, where two projection vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are chosen and normalized as the basis vectors for the x and y axes. Then the loss values around the input $\mathbf{x}$ can be calculated as:

$$V(i,j) = \ell(\mathbf{x} + i \cdot \boldsymbol{\alpha} + j \cdot \boldsymbol{\beta}, y^*). \qquad (1)$$

The coordinate (i,j) denotes the distance the origin moves along $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ directions, and $V(i,j)$ is the corresponding loss value that measures the confidence in the model prediction y^* when perturbing the original input $\mathbf{x}$. In Appendix A.1, we show more details about the input loss landscape.

3.2 Results

Figs. 2(a) and (b) show two visualizations of the loss surface in the input embedding space, which gives an intuition of the huge difference between the normal and adversarial samples: (1) The adversarial samples' loss surface has a deep and sharp bottom, while the normal samples' has a much flatter local minimum. (2) By visualizing the contour map, we find that adversarial samples are located in a very narrow valley on the loss landscape, while the normal ones reside in a wide area. The above observations suggest that, the adversarial samples are more sensitive to perturbations than normal samples. As shown in Figure 2(c), once small perturbations are injected into the inputs of adversarial samples, their loss will increase significantly and the predictions are easily flipped. The significant difference in the sharpness of the input loss landscape makes it eligible for distinguishing adversarial samples from normal ones.

This difference stems from two inherent properties of model training and adversarial sample generation. First, the model training progressively minimizes the loss of each normal training sample, while the adversarial samples are not available during the training process. Thus, normal samples are in general relatively far away from the decision boundary (Yu et al., 2019). Second, attackers aim to generate human-imperceptible adversarial perturbations, so the attack process stops once the perturbation successfully fools the model, which often results in just-cross-boundary adversarial samples (Alzantot et al., 2018; Li et al., 2020).

4 Proposed Method

In this section, we first show how a detector can be potentially designed by using loss sharpness to distinguish between adversarial and normal samples.

4.1 Sharpness of Input Loss Landscape

The sharpness of ℓ (for the model) at $\mathbf{x}$ measures the maximum increase of the prediction loss when moving $\mathbf{x}$ to a nearby input. Thus, we have the objective:

$$\max_{\|\mathbf{x}-\mathbf{x}^0\|_F \le \epsilon} \ell(\mathbf{x}, y^*) - \ell(\mathbf{x}^0, y^*), \tag{2}$$

where $\mathbf{x}$ is an input within a Frobenius ball around normal sample $\mathbf{x}^0$ with radius ϵ. This maximization problem is typically nonconcave with respect to the input $\mathbf{x}$.

Classical first-order optimization algorithms, such as projected gradient descent (PGD) (Madry et al., 2018), can be used to estimate sharpness. Starting from a given input $\mathbf{x}^0$, PGD generate a sequence $\{\mathbf{x}^k\}$ of iterates that converge to the optimal solution. If the current estimates $\mathbf{x}^k$ goes beyond the ϵ-ball, it is projected back to the ϵ-ball:

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

where η is the step size, $\mathrm{sign}(\cdot)$ denotes the sign function and $\prod_{\|\delta\| \le \epsilon}(\cdot)$ is the projection function

4.2 Convergence Analysis

The non-convexity of loss function in deep neural network makes the constrained optimization problem in Equation (2) also non-convex. How well this non-convex optimization is solved directly affects the ability to distinguish adversarial samples from normal ones. Since the gradient norm of ℓ is not an appropriate criterion for non-convex objectives, we introduce the "Frank-Wolfe (FW) gap" (Frank and Wolfe, 1956) to measure the gap between global optimum and current estimate. Consider the FW gap of Equation (2) at $\mathbf{x}^k$(Wang et al., 2019):

$$g(\mathbf{x}^k) = \max_{\mathbf{x} \in \mathcal{X}} \left\langle \mathbf{x} - \mathbf{x}^k, \nabla_{\mathbf{x}} f(\mathbf{x}^k) \right\rangle, \tag{3}$$

where $\mathcal{X} = \{\mathbf{x} | \|\mathbf{x} - \mathbf{x}^0\|_F \le \epsilon\}$ is the input domain of the ϵ-ball around normal sample $\mathbf{x}^0$, $f(\mathbf{x}^k) = \ell(\mathbf{x}^k, y^*)$ and $\langle \cdot \rangle$ is the inner product. An appealing property of FW gap is that it is invariant to an affine transformation of the domain $\{\mathbf{x} | \|\mathbf{x} - \mathbf{x}^0\|_F \le \epsilon\}$ in Equation (2) and is not tied to any specific choice of norm, unlike the criterion $\|\nabla_{\mathbf{x}} f(\mathbf{x}^k)\|$. Moreover, we always have $g(\mathbf{x}^k) \ge 0$, and a smaller value of $g(\mathbf{x}^k)$ indicates a better solution of the constrained optimization problem.

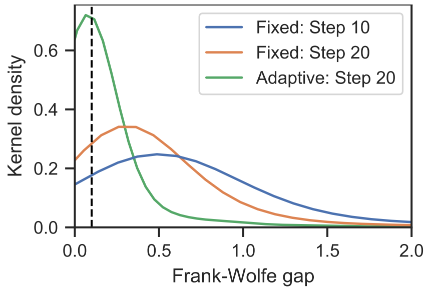


Figure 3: Distributions of "Frank-Wolfe gap" for PGD with different fixed steps and adaptive optimization, while the step size η_0 is 0.03. Adaptive strategies facilitate the convergence quality of different samples to quickly approach the same level.

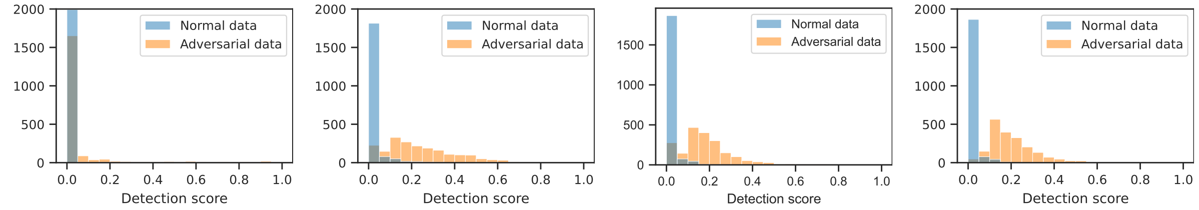
The FW gap has the following closed form solutions and can be computed for free in our proposed algorithm:

$$\begin{aligned} g(\mathbf{x}^k) &= \max_{\mathbf{x} \in \mathcal{X}} \left\langle \mathbf{x} - \mathbf{x}^k, \nabla_{\mathbf{x}} f(\mathbf{x}^k) \right\rangle \\ &= \max_{\mathbf{x} \in \mathcal{X}} \left\langle \mathbf{x} - \mathbf{x}^0 + \mathbf{x}^0 - \mathbf{x}^k, \nabla_{\mathbf{x}} f(\mathbf{x}^k) \right\rangle \\ &= \max_{\mathbf{x} \in \mathcal{X}} \left\langle \mathbf{x} - \mathbf{x}^0, \nabla_{\mathbf{x}} f(\mathbf{x}^k) \right\rangle \\ &\quad + \left\langle \mathbf{x}^0 - \mathbf{x}^k, -\nabla_{\mathbf{x}} f(\mathbf{x}^k, y^*) \right\rangle \\ &= \sqrt{\epsilon} \|\nabla_{\mathbf{x}} f(\mathbf{x}^k)\|_F - \left\langle \mathbf{x}^k - \mathbf{x}^0, \nabla_{\mathbf{x}} f(\mathbf{x}^k) \right\rangle. \end{aligned}$$

The sample-wise criterion $g(\mathbf{x}^k)$ reflects the convergence quality of $\mathbf{x}^k$ with respect to both input constraint and the loss function. Optimal convergence where $g(\mathbf{x}^k) = 0$ is achieved when 1) $\nabla_{\mathbf{x}} f(\mathbf{x}^k) = 0$, i.e., $\mathbf{x}^k$ is a stationary point in the interior of $\mathcal{X}$; or 2) $x^k - x^0 = \sqrt{\epsilon} \cdot \mathrm{sign}(\nabla_{\mathbf{x}} f(\mathbf{x}^k))$, that is, local maximum point of $f(\mathbf{x}^k)$ is reached on the boundary of $\mathcal{X}$. The FW gap allows monitoring and controlling the convergence quality of the sharpness optimization among different samples.

4.3 Adaptive Optimization

As shown in Figure 3, optimizing the maximization problem in Equation (2) at a fixed step size leads to different FW gaps among the samples. However, the concept of sharpness of a minimum is relative, and it is difficult to fairly compare the sharpness of different minima when the convergence quality of



(a) FGWS (Mozes et al., 2021) (b) RDE (Yoo et al., 2022) (c) Sharpness (ours) (d) Sharpness with. Adaptive

Figure 4: Detection score distribution of the proposed methods and two baselines. The proposed detection scores are more discriminative than other baselines. Detection scores are normalized into the range $[0, 1]$. Experimental results are obtained from BERT trained on AGNews, and adversarial samples are generated via BERT-Attack (Li et al., 2020).

Equation (2) is not the same. Thus, the inconsistent convergence quality reduces the disparity between normal and adversarial samples. It motivates us to monitor and control the quality of convergence to the identical level for all samples. Therefore, we propose to optimize the sharpness by adaptively decreasing the step size (increasing convergence quality) and stop the optimization process when a predefined convergence criterion is reached. Our proposed adaptive step size at the k-th step is:

$$\eta^k = \min\left\{0, \frac{g_{\min} - g(\mathbf{x}^k)}{g_{\min} - g(\mathbf{x}^0)} \cdot \eta^0\right\}, \qquad (4)$$

where η^0 is the initial step size and $g_{\min}$ is the predefined convergence criterion. According to the estimation of the FW gap, the step size decreases linearly towards zero as the optimization proceeds, and is zero after the convergence criterion is achieved. We use the early stopping strategy to save computational overhead during inference by halting the optimization process when the FW gap is less than $g_{\min}$. For non-convex objective, the first-order optimization method requires at most $\mathcal{O}(1/g_{\min}^2)$ iterations to find an approximate stationary point with gap smaller than $g_{\min}$.

5 Experimental Setup

We validate the effectiveness of the proposed method on three classification benchmarks: IMDB (Maas et al., 2011), SST-2 (Socher et al., 2013) and AGNews (Zhang et al., 2015). The first two are binary sentiment analysis tasks that classify reviews into positive or negative sentiment, and the last one is a classification task in which articles are categorized as world, sports, business or sci/tech. We use the widely used BERT$_{\mathrm{BASE}}$ as the target model and use three attacks to generate adversarial samples for detection.

5.1 Baselines

We compare our proposed detectors based on sharpness of input loss landscape (**Sharpness**) with several strong baselines in adversarial sample detection. **MD** (Lee et al., 2018): A simple yet effective method for detecting out-of-distribution and adversarial samples in the image processing domain. The main idea is to induce a generative classifier under Gaussian discriminant analysis, which results in a detection score based on Mahalanobis distance. **DISP** (Zhou et al., 2019): A novel framework learns to identify perturbations and can correct malicious perturbations. To detect adversarial attacks, the perturbation discriminator verifies the likelihood that a token in the text has been perturbed. **FGWS** (Mozes et al., 2021) leverages the frequency properties of adversarial word substitution to detect adversarial samples. Word-level attacks have a tendency to replace the input word with a less frequent one. **RDE** (Yoo et al., 2022): To model the probability density of the entire sentence, which uses parametric density estimation for features and generates the likelihood of a sentence being perturbed. **MDRE** (Liu et al., 2022a) is a multi-distance representation ensemble method based on the distribution characteristics of adversarial sample representations.

5.2 Adversarial Attacks

We selected three widely used attack methods according to the experimental setting used in previous work. PWWS (Ren et al., 2019) is based on a greedy algorithm that uses word saliency and prediction probability to determine word substitution order and maintains a very low word substitution rate. TextFooler (Jin et al., 2020) first identifies important words in the sentence and then replaces them with semantically similar and gram-

Dataset	Method	PWWS			TextFooler			BERT-Attack			TextFooler-adj		
		ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
SST-2	DISP (Zhou et al., 2019)	74.4	70.9	–	71.2	66.0	–	70.8	65.4	–	**79.2**	58.9	–
	MD (Lee et al., 2018)	77.5	77.2	82.0	79.6	77.0	83.4	82.7	83.2	86.1	63.8	70.3	68.6
	FGWS (Mozes et al., 2021)	82.5	81.3	85.0	72.0	63.5	69.1	70.3	63.7	69.1	64.3	68.2	69.9
	RDE (Yoo et al., 2022)	79.5	77.6	80.1	78.0	73.4	80.1	83.4	81.3	85.9	69.3	72.3	77.1
	MDRE (Liu et al., 2022a)	78.8	79.8	–	82.7	87.2	–	83.8	84.2	–	66.6	66.2	–
	Sharpness (Ours)	**85.4**	**83.8**	**91.7**	**87.0**	**86.3**	**92.8**	**90.2**	**89.7**	**95.4**	72.2	**75.0**	**75.1**
IMDB	DISP (Zhou et al., 2019)	66.8	68.2	–	68.8	70.6	–	67.3	68.8	–	68.0	67.3	–
	MD (Lee et al., 2018)	82.5	79.4	88.9	84.7	81.8	91.8	84.7	82.3	91.8	77.0	79.4	81.1
	FGWS (Mozes et al., 2021)	77.5	74.0	80.4	74.7	69.7	76.8	74.4	69.3	78.1	76.9	78.9	85.1
	RDE (Yoo et al., 2022)	82.0	74.4	90.1	83.2	75.6	92.8	83.5	76.6	92.7	78.7	80.2	86.3
	MDRE (Liu et al., 2022a)	82.7	83.6	–	84.3	86.1	–	81.3	85.5	–	78.8	80.2	–
	Sharpness (Ours)	**88.7**	**85.7**	**94.5**	**90.9**	**87.9**	**96.0**	**90.5**	**87.6**	**95.7**	**84.7**	**83.7**	**90.7**
AGNews	DISP (Zhou et al., 2019)	86.9	86.6	–	86.7	86.4	–	83.5	82.6	–	**85.8**	61.5	–
	MD (Lee et al., 2018)	77.3	76.9	83.8	79.9	79.6	85.1	82.7	78.6	85.2	52.8	67.2	62.3
	FGWS (Mozes et al., 2021)	75.0	70.6	76.6	68.3	59.6	69.2	68.2	59.4	69.1	69.8	74.6	73.2
	RDE (Yoo et al., 2022)	85.8	81.4	93.3	85.0	86.7	94.5	88.2	88.2	94.6	55.1	67.7	67.0
	MDRE (Liu et al., 2022a)	84.2	85.5	–	85.0	85.4	–	84.7	84.0	–	59.6	55.1	–
	Sharpness (Ours)	**94.9**	**93.8**	**98.4**	**96.3**	**95.8**	**98.8**	**96.3**	**95.9**	**98.8**	70.4	**70.2**	**72.8**

Table 1: Adversarial detection performance of our proposed method and baselines on BERT-base. The proposed detector outperforms the baselines consistently. The best performance is marked in bold.

matically correct synonyms until the prediction changes. BERT-Attack (Li et al., 2020) uses BERT to generate adversarial text, so that the generated adversarial samples are fluent and semantically preserved. TextFooler-adj (Morris et al., 2020) adjusts constraints to better preserve semantics and syntax, which makes adversarial samples less detectable.

5.3 Evaluation Metrics

Following previous works, we use the following three metrics to measure the effectiveness of a method in detecting adversarial samples. (1) **Detection accuracy (ACC)** corresponds to the maximum classification probability over all possible thresholds. (2) **F1-score (F1)** is defined as the harmonic mean of precision and recall. (3) **Area Under ROC (AUC)** is a threshold-independent metric that can be interpreted as the probability that a positive sample is assigned a higher detection score than a negative sample. The ROC curve describes the relationship between the true positive rate (TPR) and the false positive rate (FPR). For all three metrics, a higher value indicates better performance.

5.4 Implementation Details

We fine-tune the BERT-based victim model using the official default settings. For SST-2, we use the officially provided validation set, while for IMDB and AGNews, we use 10% of the data in the training set as the validation set. The validation set and the adversarial samples generated based the validation set are used for the selection of hyperparameters and thresholds. All three attacks are implemented using TextAttack framework with the default parameter settings.[1] Following Mozes et al. (2021), we build a balanced set consisting of $2,000$ test instances and $2,000$ adversarial samples to evaluate the detectors. For SST-2, we use all $1,872$ test data to construct the balanced set. Hyperparameters and decision thresholds of the proposed methods are presented in Appendix A.3.

6 Experimental Results and Analysis

In this section, we show the performance of the proposed method in a comprehensive way and investigate the effect of hyperparameters on performance.

6.1 Main Results

Unless specifically stated otherwise, we follow a common practice (Mozes et al., 2021; Yoo et al., 2022; Mosca et al., 2022) to ensure that our detection mechanism is tested on successful adversarial samples that can actually fool the model. Table 1 reports the detection performance of our method under various configurations. We can observe that: 1) Compared with previous detection methods, the proposed detector based on sharpness achieves significant improvements in three evaluation metrics. This demonstrates the effectiveness of sharpness of the input loss landscape in detecting adversar-

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

Dataset	Method	PWWS			TextFooler			BERT-Attack			TextFooler-adj		
		ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
SST-2	DISP (Zhou et al., 2019)	74.4	68.8	–	71.1	64.9	–	70.7	64.9	–	**78.0**	53.3	–
	MD (Lee et al., 2018)	77.2	77.9	81.1	76.7	75.2	83.2	82.4	83.0	86.0	59.2	68.9	64.0
	FGWS (Mozes et al., 2021)	69.3	61.7	65.4	65.7	55.6	62.8	64.6	53.5	61.4	64.6	68.0	68.8
	RDE (Yoo et al., 2022)	78.6	77.7	79.9	77.4	73.2	79.3	82.9	81.0	85.7	71.3	72.1	76.4
	MDRE (Liu et al., 2022a)	78.8	79.5	–	81.7	82.5	–	85.3	85.7	–	68.8	69.6	–
	Sharpness (Ours)	**83.7**	**83.1**	**90.4**	**86.4**	**86.2**	**92.4**	**89.8**	**89.3**	**95.1**	71.3	**76.1**	**77.5**
IMDB	DISP (Zhou et al., 2019)	62.4	51.9	–	64.1	53.7	–	63.2	52.6	–	59.6	61.0	–
	MD (Lee et al., 2018)	74.5	77.2	85.2	74.0	82.4	77.8	74.4	78.9	90.7	70.1	74.9	75.5
	FGWS (Mozes et al., 2021)	63.5	49.8	60.1	62.1	47.4	61.1	58.6	39.6	60.3	56.1	57.5	63.2
	RDE (Yoo et al., 2022)	76.1	74.3	78.9	76.8	72.3	78.1	77.5	8.6	78.6	68.8	70.5	76.9
	MDRE (Liu et al., 2022a)	75.4	77.5	–	76.5	80.2	–	76.5	78.8	–	69.8	70.2	–
	Sharpness (Ours)	**79.2**	**81.9**	**88.4**	**80.1**	**84.8**	**90.8**	**81.3**	**84.7**	**91.4**	**77.5**	**80.1**	**78.4**
AGNews	DISP (Zhou et al., 2019)	**85.4**	81.0	–	86.1	84.5	–	83.1	81.5	–	86.2	61.0	–
	MD (Lee et al., 2018)	73.2	71.5	79.7	77.9	77.0	83.8	79.2	78.9	85.2	58.0	68.8	68.2
	FGWS (Mozes et al., 2021)	67.7	58.4	68.7	64.7	52.8	65.4	64.1	51.6	64.3	58.8	59.1	60.5
	RDE (Yoo et al., 2022)	77.0	78.5	85.9	85.1	84.4	90.2	86.6	85.7	91.4	62.0	69.2	70.7
	MDRE (Liu et al., 2022a)	75.8	77.2	–	81.8	82.4	–	84.1	84.4	–	66.3	62.4	–
	Sharpness (Ours)	84.7	**83.9**	**90.4**	**90.7**	**90.5**	**95.2**	**94.1**	**94.1**	**97.4**	**75.3**	**76.5**	**76.4**

Table 2: Adversarial detection performance of our proposed method and baselines on RoBERTa-base. The best performance is marked in bold.

Dataset	Method	PWWS	TextFooler	BERT-Attack
SST-2	MD	56.4	56.4	61.0
	FGWS	0.0	0.0	0.0
	RDE	54.4	51.3	65.5
	Sharpness	**68.3**	**70.4**	**83.6**
IMDB	MD	61.5	68.1	67.8
	FGWS	0.0	0.0	0.0
	RDE	69.5	73.3	74.1
	Sharpness	**80.1**	**86.2**	**85.0**
AGNews	MD	37.4	40.0	40.8
	FGWS	0.0	0.0	0.0
	RDE	72.6	77.3	74.4
	Sharpness	**94.6**	**96.1**	**96.0**

Table 3: Adversarial detection performance on the metric **TNR@95TPR**.

ial samples. 2) The performance of FGWS decreases under TextFooler and BERT-Attack, which are more subtle attacks with less significant frequency differences, as FGWS relies on the occurrence of rare words. FGWS also performs poorly on AGNews, most likely because it covers four different news domains, resulting in its low word frequency. These results are consistent with the results reported by Yoo et al. (2022). 3) DISP is a threshold-independent method and therefore AUC metric is not applicable. DISP does not perform well except on AGNews dataset.

6.2 More Rigorous Metric

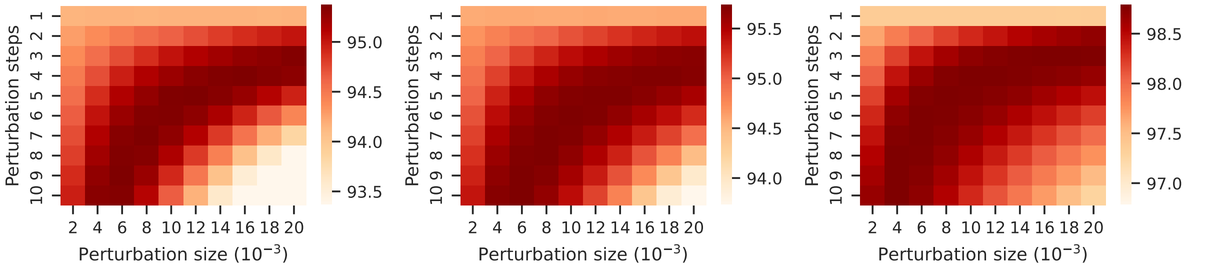
TNR@95TPR is short for true negative rate (TNR) at 95% true positive rate (TPR), which is widely used in out-of-distribution detection (Li et al., 2021a; Liang et al., 2018). But to our knowledge, no textual adversarial sample detector has been evaluated using this metric. TNR@95TPR can be interpreted as the probability of a normal sample being correctly classified (Acc-) when the probability of an adversarial sample being correctly classified (Acc+) is as high as 95%. As can be seen in Table 3, with this strict evaluation metric, there is a significant advantage for our prediction-loss-based detector, while FGWS fails to detect the normal samples at all.

6.3 More Model

In previous experiments, all results are based on the BERT-base model, and we also evaluate the performance of the proposed method on RoBERTa-base (Liu et al., 2019). Table 2 shows the detection results using RoBERTa as the victim model. The overall trend among detection methods is similar to Table 1. From the results in Tables 1 and 2, it can be concluded that our proposed methods perform as stable as the traditional statistical-based methods (MD and RDE) under different experimental settings, while empirically designed DISP and FGWS do not perform consistently.

6.4 Ablation Study

To better illustrate the contribution of adaptive optimization strategy to the proposed detector, we perform an ablation study by removing adaptive



(a) SST-2 (b) IMDB (c) AGNews

Figure 5: Heatmaps of AUC for the proposed method, with different perturbation sizes and numbers of steps on BERT-base. Adversarial samples are generated via BERT-Attack. The darker the color, the better the performance.

Dataset	Method	ACC	F1	AUC
SST-2	Sharpness	90.2	89.7	95.4
	w/o Adaptive	78.9	83.0	81.1
IMDB	Sharpness	90.5	87.6	95.7
	w/o Adaptive	80.6	80.4	89.6
AGNews	Sharpness	96.3	96.2	98.7
	w/o Adaptive	87.6	87.2	92.4

Table 4: Ablation study on the proposed detector. We remove the adaptive optimization strategy (**w/o** Adaptive) to illustrate the the importance of adaptive optimization strategy. Results are obtained from BERT-base, and adversarial samples are generated via BERT-Attack.

optimization (**w/o** Adaptive). The experimental results are shown in Table 4. We can observe that the adaptive optimization strategy is important for the sharpness calculation. The inconsistent convergence quality reduces the disparity between normal and adversarial samples.

6.5 Hyper-parameter Investigation

6.5.1 Detection Threshold

To investigate the influence of detection thresholds, we analyze the performance with different thresholds on the three datasets, as shown in Figure 6. The performance of the proposed detector gradually improves as the threshold increases, but when the threshold is too large, the results of the detectors are concentrated in one certain category, leading to a decrease in performance. The peak performance of both detectors occurs near the midpoint of the potential thresholds, indicating that our method performs well on both normal and adversarial samples.

6.5.2 Parameters of Optimization

Figure 5 shows the detection performance with different step sizes and numbers of steps. In order to show more intuitively the effect of optimization steps and size on the AUC metric, we preserve the results within 2 percents below the highest value, and the rest of the data are shown as light-colored blocks in Figure 5. We can observe that the proposed detector achieve sufficiently consistent performance under various optimization parameters (i.e., the number of steps K and step size η), and the detection performance is decided by $\delta_K \approx K \times \eta$.

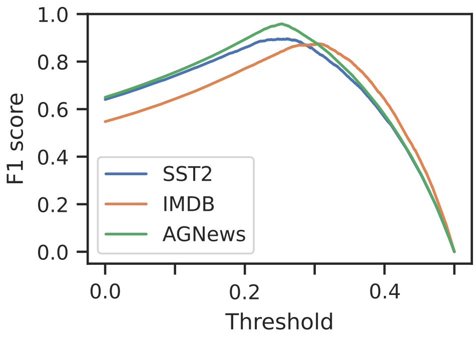


Figure 6: F1 score with respect to detection threshold of the proposed detector using BERT-base as the victim model and adversarial samples are generated via BERT-Attack.

7 Conclusion

Our work starts from a finding: adversarial samples locate in steep and narrow local minima of the loss landscape while normal samples, which differs distinctly from adversarial ones, reside in the loss surface that is more flatter. Based on this, we propose a simple and effective sharpness-based detector that uses an adaptive optimization strategy to compute sharpness. Experimental results have demonstrated the superiority of our proposed method compared to baselines, and analytical experiments have further verified the good performance of our method under different parameters.

Limitations

In this work, we propose a detector that aims to detect adversarial samples via sharpness of input loss landscape for model. However, the computational cost of the sharpness is high because it requires at most K-step gradient descents. Moreover, in this work, we mainly considered word-level adversarial sample detection as often studied in previous work, while character-level and sentence-level adversarial samples are not studied. These two problems will be explored in our future work.

Acknowledgements

Detecting Adversarial Samples through Sharpness of Loss Landscape: The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.62206057), Shanghai Rising-Star Program (23QA1400200), and Natural Science Foundation of Shanghai (23ZR1403500), except the fourth auther Qin Liu, who is funded by Graduate Fellowship from University of Southern California.

References

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

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. 2020. Detecting adversarial samples using influence functions and nearest neighbors. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14441–14450. Computer Vision Foundation / IEEE.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. 2017. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1019–1028. PMLR.

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.

Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. 2017. Detecting adversarial samples from artifacts. *CoRR*, abs/1703.00410.

Marguerite Frank and Philip Wolfe. 1956. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

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 and Oriol Vinyals. 2015. Qualitatively characterizing neural network optimization problems. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jens Hauser, Zhao Meng, Damián Pascual, and Roger Wattenhofer. 2021. BERT is robust! A case against synonym-based adversarial examples in text classification. *CoRR*, abs/2109.07403.

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.

Simon Lacoste-Julien. 2016. Convergence rate of frank-wolfe for non-convex objectives. *CoRR*, abs/1607.00345.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*

31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pages 7167–7177.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6391–6401.

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

Xiaoya Li, Jiwei Li, Xiaofei Sun, Chun Fan, Tianwei Zhang, Fei Wu, Yuxian Meng, and Jun Zhang. 2021a. kFolden: k-fold ensemble for out-of-distribution detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3102–3115, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021b. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shiyu Liang, Yixuan Li, and R. Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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.

Na Liu, Mark Dras, and Wei Emma Zhang. 2022a. Detecting textual adversarial examples based on distributional characteristics of data representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 78–90, Dublin, Ireland. Association for Computational Linguistics.

Qin Liu, Rui Zheng, Bao Rong, Jingyi Liu, ZhiHua Liu, Zhanzhan Cheng, Liang Qiao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022b. 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 pretraining approach. *CoRR*, abs/1907.11692.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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.

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

Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh. 2022. "that is a suspicious reaction!": Interpreting logits variation to detect NLP adversarial attacks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7806–7816, Dublin, Ireland. Association for Computational Linguistics.

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

Marwan Omar, Soohyeon Choi, DaeHun Nyang, and David Mohaisen. 2022. Robust natural language processing: Recent advances, challenges, and future directions. *CoRR*, abs/2201.00768.

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.

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.

Jiayi Wang, Rongzhou Bao, Zhuosheng Zhang, and Hai Zhao. 2022. Distinguishing non-natural from natural adversarial samples for more robust pre-trained language model. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 905–915, Dublin, Ireland. Association for Computational Linguistics.

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.

Xiaosen Wang, Yifeng Xiong, and Kun He. 2021. Randomized substitution and vote for textual adversarial example detection. *CoRR*, abs/2109.05698.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. 2019. On the convergence and robustness of adversarial training. 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 6586–6595. PMLR.

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: ACL 2022*, pages 3656–3672, Dublin, Ireland. Association for Computational Linguistics.

Fuxun Yu, Zhuwei Qin, Chenchen Liu, Liang Zhao, Yanzhi Wang, and Xiang Chen. 2019. Interpreting and evaluating neural network robustness. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4199–4205. ijcai.org.

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, 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.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4904–4913, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Input Loss Landscape

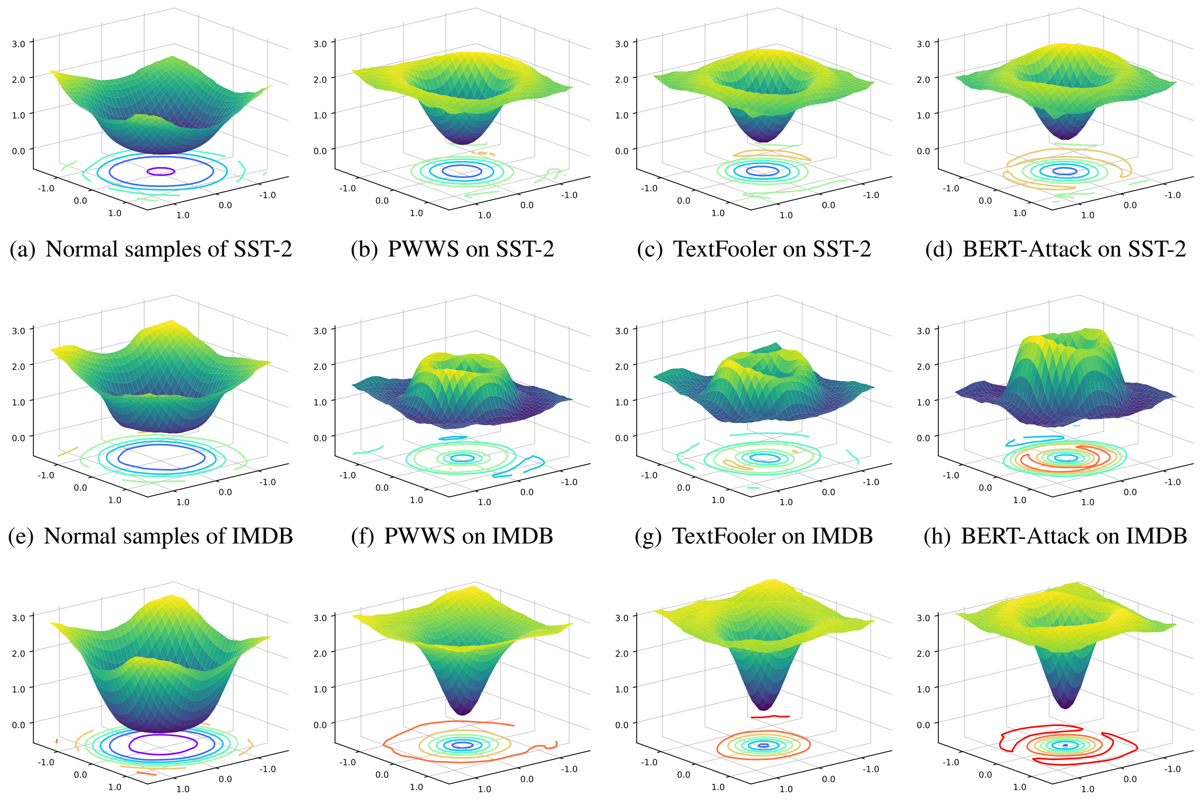
In Figure 7, we comprehensively show the differences between the input loss landscapes of normal and adversarial samples on three datasets. Adversarial samples are generated by the three textual adversarial attacks that we used in the experimental section. Front elevation view of the input loss landscape on IMDB is shown in Figure 8. The sharp input loss landscape of adversarial samples is not a coincidence; it is a general phenomenon.

A.2 Detection Score

As a supplement, we show the detection score distributions of the proposed detectors and other baseline methods on the SST-2 and IMDB datasets in Figure 9. Our detection scores are still more discriminative than the other baselines.

A.3 Hyperparameters

The optimal hyperparameter values are task-specific, but the following range of possible values works well in all tasks: 1) the number of steps K: $1, 2, \ldots, 10$; 2) step size η is tuned via a grid search within the range of $[2e^{-3}, 2e^{-2}]$ with interval $2e^{-2}$; 3) decision threshold is chosen via a grid search within the range of $[0, 1]$ with interval $1e^{-2}$.



(a) Normal samples of SST-2

(b) PWWS on SST-2

(c) TextFooler on SST-2

(d) BERT-Attack on SST-2

(e) Normal samples of IMDB

(f) PWWS on IMDB

(g) TextFooler on IMDB

(h) BERT-Attack on IMDB

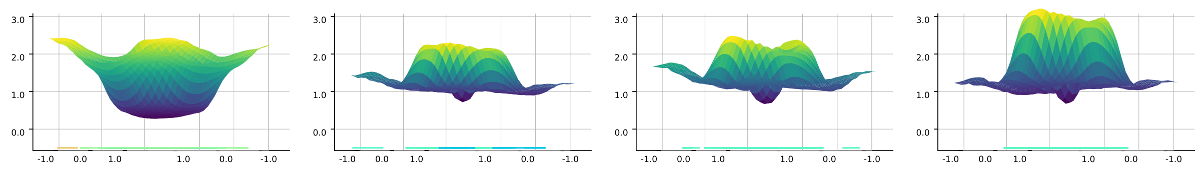
(i) Normal samples of AG-News

(j) PWWS on AGNews

(k) TextFooler on AGNews

(l) BERT-Attack on AGNews

Figure 7: Input loss landscapes of normal and adversarial samples on SST-2, IMDB and AGNews datasets. The adversarial samples are generated by textual adversarial attacks including PWWS, TextFooler and BERT-Attack.



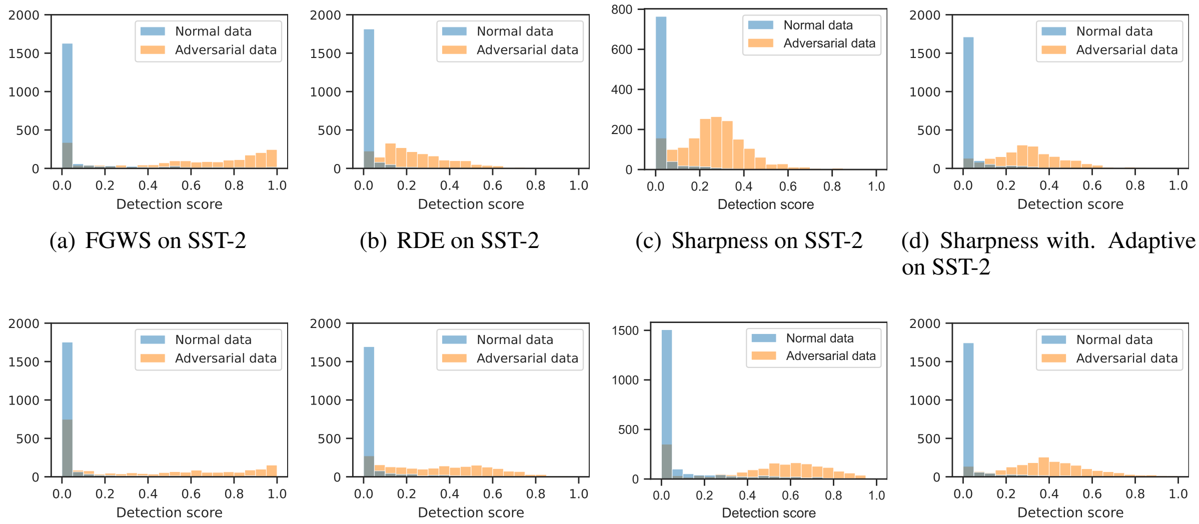
(a) Normal samples of IMDB

(b) PWWS on IMDB

(c) TextFooler on IMDB

(d) BERT-Attack on IMDB

Figure 8: Front evaluation view of the input loss landscape on IMDB.



(a) FGWS on SST-2

(b) RDE on SST-2

(c) Sharpness on SST-2

(d) Sharpness with. Adaptive on SST-2

(e) FGWS on IMDB

(f) RDE on IMDB

(g) Sharpness on IMDB

(h) Sharpness with. Adaptive on IMDB

Figure 9: Detection score distribution of the proposed methods and two baselines on SST-2 and IMDB datasets. Detection scores are normalized into the range $[0, 1]$. The proposed detection scores are more discriminative than other baselines. Adversarial samples are generated via BERT-Attack.