SAFER: A Structure-free Approach for Certified Robustness to Adversarial Word Substitutions

Mao Ye* UT Austin my21@cs.utexas.edu
Chengyue Gong* UT Austin cygong@cs.utexas.edu
Qiang Liu UT Austin lqiang@cs.utexas.edu

Abstract

State-of-the-art NLP models can often be fooled by human-unaware transformations such as synonymous word substitution. For security reasons, it is of critical importance to develop models with *certified robustness* that can provably guarantee that the prediction is can not be altered by any possible synonymous word substitution. In this work, we propose a certified robust method based on a new randomized smoothing technique, which constructs a stochastic ensemble by applying random word substitutions on the input sentences, and leverage the statistical properties of the ensemble to provably certify the robustness. Our method is simple and *structure-free* in that it only requires the black-box queries of the model outputs, and hence can be applied to any pre-trained models (such as BERT) and any types of models (world-level or subword-level). Our method significantly outperforms recent state-of-the-art methods for certified robustness on both IMDB and Amazon text classification tasks. To the best of our knowledge, we are the first work to achieve certified robustness on large systems such as BERT with practically meaningful certified accuracy.

1 Introduction

Deep neural networks have achieved state-of-the-art results in many NLP tasks, but also have been shown to be brittle to carefully crafted adversarial perturbations, such as replacing words with similar words (Alzantot et al., 2018), adding extra text (Wallace et al., 2019), and replacing sentences with semantically similar sentences (Ribeiro et al., 2018). These adversarial perturbations are imperceptible to humans, but can fool deep neural networks and break their performance. Efficient methods for defending these attacks are of critical importance for deploying modern deep NLP models to practical automatic AI systems.

In this paper, we focus on defending the synonymous word substitution attacking (Alzantot et al., 2018), in which an attacker attempts to alter the output of the model by replacing words in the input sentence with their synonyms according to a synonym table, while keeping the meaning of this sentence unchanged. A model is said to be *certified robust* if such an attack is guaranteed to fail, no matter how the attacker manipulates the input sentences. Achieving and verifying certified robustness is highly challenging even if the synonym table used by the attacker is known during training (see Jia et al., 2019), because it requires to check every possible synonymous word substitution, whose number is exponentially large.

Various defense methods against synonymous word substitution attacks have been developed (e.g., Wallace et al., 2019; Ebrahimi et al., 2018), most of which, however, are not certified robust in that they may eventually be broken by stronger attackers. Recently, Jia et al. (2019); Huang et al. (2019) proposed the first certified robust methods against word substitution attacking. Their methods are based on the interval bound propagation (IBP) method (Dvijotham et al., 2018) which computes the range of the model output by propagating the interval constraints of the inputs layer by layer.

However, the IBP-based methods of Jia et al. (2019); Huang et al. (2019) are limited in several ways. First, because IBP only works for certifying neural networks with continuous inputs, the inputs in Jia et al. (2019) and Huang et al. (2019) are taken to be the word embedding vectors of the input sentences, instead of the discrete sentences. This makes it inapplicable to character-level (Zhang et al., 2015) and subword-level (Bojanowski et al., 2017) model, which are more widely used in practice (Wu et al., 2016).

*Equal contribution

In this paper, we propose a *structure-free* certified defense method that applies to arbitrary models that can be queried in a black-box fashion, without any requirement on the model structures. Our method is based on the idea of randomized smoothing, which smooths the model with random word substitutions build on the synonymous network, and leverage the statistical properties of the randomized ensembles to construct provably certification bounds. Similar ideas of provably certification using randomized smoothing have been developed recently in deep learning (e.g., Cohen et al., 2019; Salman et al., 2019; Zhang et al., 2020; Lee et al., 2019), but mainly for computer vision tasks whose inputs (images) are in a continuous space (Cohen et al., 2019). Our method admits a substantial extension of the randomized smoothing technique to discrete and structured input spaces for NLP.

We test our method on various types of NLP models, including text CNN (Kim, 2014), Char-CNN (Zhang et al., 2015), and BERT (Devlin et al., 2019). Our method significantly outperforms the recent IBP-based methods (Jia et al., 2019; Huang et al., 2019) on both IMDB and Amazon text classification. In particular, we achieve an 87.35% certified accuracy on IMDB by applying our method on the state-of-the-art BERT, on which previous certified robust methods are not applicable.

2 Adversarial Word Substitution

In a text classification task, a model $f(\mathbf{X})$ maps an input sentence $\mathbf{X} \in \mathcal{X}$ to a label c in a set $\mathcal{Y}$ of discrete categories, where $\mathbf{X} = x_1, \ldots, x_L$ is a sentence consisting of L words. In this paper, we focus on adversarial word substitution in which an attacker arbitrarily replaces the words in the sentence by their synonyms according to a synonym table to alert the prediction of the model. Specifically, for any word x, we consider a pre-defined synonym set S_x that contains the synonyms of x (including x itself). We assume the synonymous relation is symmetric, that is, x is in the synonym set of all its synonyms. The synonym set S_x can be built based on GLOVE (Pennington et al., 2014).

With a given input sentence $\mathbf{X} = x_1, \ldots, x_L$, the attacker may construct an adversarial sentence $\mathbf{X}' = x'_1, \ldots, x'_L$ by perturbing at most $R \leq L$ words x_i in $\mathbf{X}$ to any of their synonyms $x'_i \in S_{x_i}$,

$$S_{\mathbf{X}} := \left\{ \mathbf{X}' \colon \ \left\| \mathbf{X}' - \mathbf{X} \right\|_0 \leq R, \ \ x'_i \in S_{x_i}, \forall i \right\},$$

where $S_{\mathbf{X}}$ denotes the candidate set of adversarial sentences available to the attacker. Here $\|\mathbf{X}' - \mathbf{X}\|_0 := \sum_{i=1}^{L} \mathbb{I}\{x'_i \neq x_i\}$ is the Hamming distance, with $\mathbb{I}\{\cdot\}$ the indicator function. It is expected that all $\mathbf{X}' \in S_{\mathbf{X}}$ have the same semantic meaning as $\mathbf{X}$ for human readers, but they may have different outputs from the model. The goal of the attacker is to find $\mathbf{X}' \in S_{\mathbf{X}}$ such that $f(\mathbf{X}) \neq f(\mathbf{X}')$.

Certified Robustness Formally, a model f is said to be *certified robust* against word substitution attacking on an input $\mathbf{X}$ if it is able to give consistently correct predictions for all the possible word substitution perturbations, i.e,

$$y = f(\mathbf{X}) = f(\mathbf{X}'), \quad \text{for all } \mathbf{X}' \in S_{\mathbf{X}}, \qquad (1)$$

where y denotes the true label of sentence $\mathbf{X}$. Deciding if f is certified robust can be highly challenging, because, unless additional structural information is available, it requires to exam all the candidate sentences in $S_{\mathbf{X}}$, whose size grows exponentially with R. In this work, we mainly consider the case when $R = L$, which is the most challenging case.

3 Certifying Smoothed Classifiers

Our idea is to replace f with a more smoothed model that is easier to verify by averaging the outputs of a set of randomly perturbed inputs based on random word substitutions. The smoothed classifier f^{RS} is constructed by introducing random perturbations on the input space,

$$f^{\mathrm{RS}}(\mathbf{X}) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}_{\mathbf{Z} \sim \Pi_{\mathbf{X}}} \left(f(\mathbf{Z}) = c \right),$$

where $\Pi_{\mathbf{X}}$ is a probability distribution on the input space that prescribes a random perturbation around $\mathbf{X}$. For notation, we define

$$g^{\mathrm{RS}}(\mathbf{X}, c) := \mathbb{P}_{\mathbf{Z} \sim \Pi_{\mathbf{X}}} \left(f(\mathbf{Z}) = c \right),$$

which is the "soft score" of class c under f^{RS}.

The perturbation distribution $\Pi_{\mathbf{X}}$ should be chosen properly so that f^{RS} forms a close approximation to the original model f (i.e., $f^{\mathrm{RS}}(\mathbf{X}) \approx f(\mathbf{X})$), and is also sufficiently random to ensure that f^{RS} is smooth enough to allow certified robustness (in the sense of Theorem 1 below).

In our work, we define $\Pi_{\mathbf{X}}$ to be the uniform distribution on a set of random word substitutions. Specifically, let P_x be a *perturbation set* for word x in the vocabulary, which is different from the *synonym set* S_x. In this work, we construct P_x based on the top K nearest neighbors under the cosine

similarity of GLOVE vectors, where K is a hyper-parameter that controls the size of the perturbation set; see Section 4 for more discussion on P_x.

For a sentence $\mathbf{X} = x_1, \ldots, x_L$, the sentence-level perturbation distribution $\Pi_{\mathbf{X}}$ is defined by randomly and independently perturbing each word x_i to a word in its perturbation set P_{x_i} with equal probability, that is,

$$\Pi_{\mathbf{X}}(\mathbf{Z}) = \prod_{i=1}^{L} \frac{\mathbb{I}\{z_i \in P_{x_i}\}}{|P_{x_i}|},$$

where $\mathbf{Z} = z_1, \ldots, z_L$ is the perturbed sentence and $|P_{x_i}|$ denotes the size of P_{x_i}. Note that the random perturbation $\mathbf{Z}$ and the adversarial candidate $\mathbf{X}' \in S_{\mathbf{X}}$ are different.

3.1 Certified Robustness

We now discuss how to certify the robustness of the smoothed model f^{RS}. Recall that f^{RS} is certified robust if $y = f^{\text{RS}}(\mathbf{X}')$ for any $\mathbf{X}' \in S_{\mathbf{X}}$, where y is the true label. A sufficient condition for this is

$$\min_{\mathbf{X}' \in S_{\mathbf{X}}} g^{\text{RS}}(\mathbf{X}', y) \geq \max_{\mathbf{X}' \in S_{\mathbf{X}}} g^{\text{RS}}(\mathbf{X}', c) \quad \forall c \neq y,$$

where the lower bound of $g^{\text{RS}}(\mathbf{X}', y)$ on $\mathbf{X}' \in S_{\mathbf{X}}$ is larger than the upper bound of $g^{\text{RS}}(\mathbf{X}', c)$ on $\mathbf{X}' \in S_{\mathbf{X}}$ for every $c \neq y$. The key step is hence to calculate the upper and low bounds of $g^{\text{RS}}(\mathbf{X}', c)$ for $\forall c \in \mathcal{Y}$ and $\mathbf{X}' \in S_{\mathbf{X}}$, which we address in Theorem 1 below. All proofs are in Appendix A.2.

Theorem 1. ***(Certified Lower/Upper Bounds)*** *Assume the perturbation set P_x is constructed such that $|P_x| = |P_{x'}|$ for every word x and its synonym $x' \in S_x$. Define*

$$q_x = \min_{x' \in S_x} |P_x \cap P_{x'}|/|P_x|,$$

where q_x indicates the overlap between the two different perturbation sets. For a given sentence $\boldsymbol{X} = x_1, \ldots, x_L$, we sort the words according to q_x, such that $q_{x_{i_1}} \leq q_{x_{i_2}} \leq \cdots \leq q_{x_{i_L}}$. Then

$$\min_{\boldsymbol{X}' \in S_{\boldsymbol{X}}} g^{RS}(\boldsymbol{X}', c) \geq \max(g^{RS}(\boldsymbol{X}, c) - q_{\boldsymbol{X}}, 0)$$
$$\max_{\boldsymbol{X}' \in S_{\boldsymbol{X}}} g^{RS}(\boldsymbol{X}', c) \leq \min(g^{RS}(\boldsymbol{X}, c) + q_{\boldsymbol{X}}, 1).$$

where $q_{\boldsymbol{X}} := 1 - \prod_{j=1}^{R} q_{x_{i_j}}$. Equivalently, this says

$$\left|g^{RS}(\boldsymbol{X}', c) - g^{RS}(\boldsymbol{X}, c)\right| \leq q_{\boldsymbol{X}}, \quad \textit{any label } c \in \mathcal{Y}.$$

The idea is that, with the randomized smoothing, the difference between $g^{\text{RS}}(\mathbf{X}', c)$ and $g^{\text{RS}}(\mathbf{X}, c)$ is at most $q_{\mathbf{X}}$ for any adversarial candidate $\mathbf{X}' \in S_{\mathbf{X}}$. Therefore, we can give adversarial upper and lower bounds of $g^{\text{RS}}(\mathbf{X}', c)$ by $g^{\text{RS}}(\mathbf{X}, c) \pm q_{\mathbf{X}}$, which, importantly, avoids the difficult adversarial optimization of $g^{\text{RS}}(\mathbf{X}', c)$ on $\mathbf{X}' \in S_{\mathbf{X}}$, and instead just needs to evaluate $g^{\text{RS}}(\mathbf{X}, c)$ at the original input $\mathbf{X}$.

We are ready to describe a practical criterion for checking the certified robustness.

Proposition 1. *For a sentence $\boldsymbol{X}$ and its label y, we define*

$$y_B = \arg \max_{c \in \mathcal{Y}, c \neq y} g^{RS}(\boldsymbol{X}, c).$$

Then under the condition of Theorem 1, we can certify that $f(\boldsymbol{X}') = f(\boldsymbol{X}) = y$ for any $\boldsymbol{X}' \in S_{\boldsymbol{X}}$ if

$$\Delta_{\boldsymbol{X}} \stackrel{def}{=} g^{RS}(\boldsymbol{X}, y) - g^{RS}(\boldsymbol{X}, y_B) - 2q_{\boldsymbol{X}} > 0. \quad (2)$$

Therefore, certifying whether the model gives consistently correct prediction reduces to checking if $\Delta_{\mathbf{X}}$ is positive, which can be easily achieved with Monte Carlo estimation as we show in the sequel.

Estimating $g^{\text{RS}}(\mathbf{X}, c)$ and $\Delta_{\mathbf{X}}$ Recall that $g^{\text{RS}}(\mathbf{X}, c) = \mathbb{P}_{\mathbf{Z} \sim \Pi_{\mathbf{X}}}(f(\mathbf{Z}) = c)$. We can estimate $g^{\text{RS}}(\mathbf{X}, c)$ with a Monte Carlo estimator $\sum_{i=1}^{n} \mathbb{I}\{f(\mathbf{Z}^{(i)}) = c\}/n$, where $\mathbf{Z}^{(i)}$ are i.i.d. samples from $\Pi_{\mathbf{X}}$. And $\Delta_{\mathbf{X}}$ can be approximated accordingly. Using concentration inequality, we can quantify the non-asymptotic approximation error. This allows us to construct rigorous statistical procedures to reject the null hypothesis that f^{RS} is not certified robust at $\mathbf{X}$ (i.e., $\Delta_{\mathbf{X}} \leq 0$) with a given significance level (e.g., 1%). See Appendix A.1 for the algorithmic details of the testing procedure.

We can see that our procedure is *structure-free* in that it only requires the black-box assessment of the output $f(\mathbf{Z}^{(i)})$ of the random inputs, and does not require any other structural information of f and f^{RS}, which makes our method widely applicable to various types of complex models.

Tightness A key question is if our bounds are sufficiently tight. The next theorem shows that the lower/upper bounds in Theorem 1 are tight and can not be further improved unless further information of the model f or f^{RS} is acquired.

Theorem 2. ***(Tightness)*** *Assume the conditions of Theorem 1 hold. For a model f that satisfies $f^{RS}(\boldsymbol{X}) = y$ and y_B as defined in Proposition 1, there exists a model f_* such that its related smoothed classifier g_*^{RS} satisfies $g_*^{RS}(\boldsymbol{X}, c) =$*

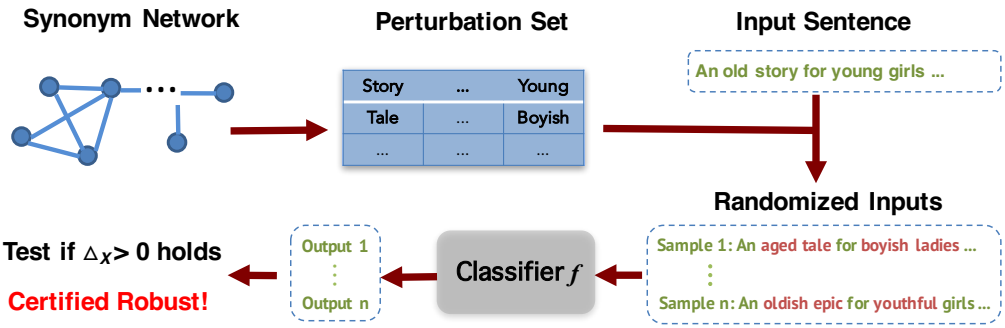

Figure 1: A pipeline of the proposed robustness certification approach.

$g^{RS}(\boldsymbol{X}, c)$ *for* $c = y$ *and* $c = y_B$, *and*

$$\min_{\boldsymbol{X}' \in S_{\boldsymbol{X}}} g_*^{RS}(\boldsymbol{X}', y) = \max(g_*^{RS}(\boldsymbol{X}, y) - q_{\boldsymbol{X}}, 0)$$

$$\max_{\boldsymbol{X}' \in S_{\boldsymbol{X}}} g_*^{RS}(\boldsymbol{X}', y_B) = \min(g_*^{RS}(\boldsymbol{X}, y_B) + q_{\boldsymbol{X}}, 1),$$

where $q_{\boldsymbol{X}}$ *is defined in Theorem 1.*

In other words, if we access g^{RS} only through the evaluation of $g^{\text{RS}}(\mathbf{X}, y)$ and $g^{\text{RS}}(\mathbf{X}, y_B)$, then the bounds in Theorem 1 are the tightest possible that we can achieve, because we can not distinguish between g^{RS} and the g_*^{RS} in Theorem 2 with the information available.

3.2 Practical Algorithm

Figure 1 visualizes the pipeline of the proposed approach. Given the synonym sets $S_{\mathbf{X}}$, we generate the perturbation sets $P_{\mathbf{X}}$ from it. When an input sentence $\mathbf{X}$ arrives, we draw perturbed sentences $\{\mathbf{Z}^{(i)}\}$ from $\Pi_{\mathbf{X}}$ and average their outputs to estimate $\Delta_{\mathbf{X}}$, which is used to decide if the model is certified robust for $\mathbf{X}$.

Training the Base Classifier f Our method needs to start with a base classifier f. Although it is possible to train f using standard learning techniques, the result can be improved by considering that the method uses the smoothed f^{RS}, instead of f. To improve the accuracy of f^{RS}, we introduce a data augmentation induced by the perturbation set. Specifically, at each training iteration, we first sample a mini-batch of data points (sentences) and randomly perturbing the sentences using the perturbation distribution $\Pi_{\mathbf{X}}$. We then apply gradient descent on the model based on the perturbed mini-batch. Similar training procedures were also used for Gaussian-based random smoothing on continuous inputs (see e.g., Cohen et al., 2019).

Our method can easily leverage powerful pre-trained models such as BERT. In this case, BERT is used to construct feature maps and only the top layer weights are finetuned using the data augmentation method.

4 Experiments

We test our method on both IMDB (Maas et al., 2011) and Amazon (McAuley, 2013) text classification tasks, with various types of models, including text CNN (Kim, 2014), Char-CNN (Zhang et al., 2015) and BERT (Devlin et al., 2019). We compare with the recent IBP-based methods (Jia et al., 2019; Huang et al., 2019) as baselines. Text CNN (Kim, 2014) was used in Jia et al. (2019) and achieves the best result therein. All the baseline models are trained and tuned using the schedules recommended in the corresponding papers. We consider the case when $R = L$ during attacking, which means all words in the sentence can be perturbed simultaneously by the attacker. Code for reproducing our results can be found in `https://github.com/lushleaf/Structure-free-certified-NLP`.

Synonym Sets Similar to Jia et al. (2019); Alzantot et al. (2018), we construct the synonym set S_x of word x to be the set of words with ≥ 0.8 cosine similarity in the GLOVE vector space. The word vector space is constructed by post-processing the pre-trained GLOVE vectors (Pennington et al., 2014) using the counter-fitted method (Mrkšić et al., 2016) and the "all-but-the-top" method (Mu and Viswanath, 2018) to ensure that synonyms are near to each other while antonyms are far apart.

Perturbation Sets We say that two words x and x' are connected synonymously if there exists a path of words $x = x_1, x_2, \ldots, x_\ell = x'$, such that all the successive pairs are synonymous. Let B_x to be the set of words connected to x synonymously. Then we define the perturbation set P_x to consist of the top K words in B_x with the largest GLOVE cosine similarity if $|B_x| \geq K$, and set $P_x = B_x$ if $|B_x| < K$. Here K is a hyper-parameter that controls the size of P_x and hence trades off the smoothness and accuracy of f^{RS}. We use $K = 100$ by default and investigate its effect in Section 4.2.

Method	IMDB	Amazon
Jia et al. (2019)	79.74	14.00
Huang et al. (2019)	78.74	12.36
Ours	**81.16**	**24.92**

Table 1: The certified accuracy of our method and the baselines on the IMDB and Amazon dataset.

Evaluation Metric We evaluate the certified robustness of a model f^{RS} on a dataset with the *certified accuracy* (Cohen et al., 2019), which equals the percentage of data points on which f^{RS} is certified robust, which, for our method, holds when $\Delta_{\mathbf{X}} > 0$ can be verified.

4.1 Main Results

We first demonstrate that adversarial word substitution is able to give strong attack in our experimental setting. Using IMDB dataset, we attack the vanilla BERT (Devlin et al., 2019) with the adversarial attacking method of Jin et al. (2020). The vanilla BERT achieves a 91% clean accuracy (the testing accuracy on clean data without attacking), but only a 20.1% adversarial accuracy (the testing accuracy under the particular attacking method by Jin et al. (2020)). We will show later that our method is able to achieve 87.35% certified accuracy and thus the corresponding adversarial accuracy must be higher or equal to 87.35%.

We compare our method with IBP (Jia et al., 2019; Huang et al., 2019). in Table 1. We can see that our method clearly outperforms the baselines. In particular, our approach significantly outperforms IBP on Amazon by improving the 14.00% baseline to 24.92%.

Thanks to its structure-free property, our algorithm can be easily applied to any pre-trained models and character-level models, which is not easily achievable with Jia et al. (2019) and Huang et al. (2019). Table 2 shows that our method can further improve the result using Char-CNN (a character-level model) and BERT (Devlin et al., 2019), achieving an 87.35% certified accuracy on IMDB. In comparison, the IBP baseline only achieves a 79.74% certified accuracy under the same setting.

4.2 Trade-Off between Clean Accuracy and Certified Accuracy

We investigate the trade-off between smoothness and accuracy while tuning K in Table 3. We can see that the clean accuracy decreases when K increases, while the gap between the clean accuracy and certified accuracy, which measures the smoothness, decreases when K increases. The best certified accuracy is achieved when $K = 100$.

Method	Model	Accuracy
Jia et al. (2019)	CNN	79.74
Huang et al. (2019)	CNN	78.74
Ours	CNN	81.16
	Char-CNN	82.03
	BERT	**87.35**

Table 2: The certified accuracy of different models and methods on the IMDB dataset.

K	20	50	100	250	1000
Clean (%)	88.47	88.48	88.09	84.83	67.54
Certified (%)	65.58	77.32	81.16	79.98	65.13

Table 3: Results of the smoothed model f^{RS} with different K on IMDB using text CNN. "Clean" represents the accuracy on the clean data without adversarial attacking and "Certified" the certified accuracy.

5 Conclusion

We proposed a robustness certification method, which provably guarantees that all the possible perturbations cannot break down the system. Compared with previous work such as Jia et al. (2019); Huang et al. (2019), our method is structure-free and thus can be easily applied to any pre-trained models (such as BERT) and character-level models (such as Char-CNN).

The construction of the perturbation set is of critical importance to our method. In this paper, we used a heuristic way based on the synonym network to construct the perturbation set, which may not be optimal. In further work, we will explore more efficient ways for constructing the perturbation set. We also plan to generalize our approach to achieve certified robustness against other types of adversarial attacks in NLP, such as the out-of-list attack. An naïve way is to add the "OOV" token into the synonyms set of every word, but potentially better procedures can be further explored.

Acknowledgement

This work is supported in part by NSF CRII 1830161 and NSF CAREER 1846421.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *ACL*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. In *ACL*.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *ICML*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. 2018. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *ACL*.

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

R. Jia, A. Raghunathan, K. Gksel, and P. Liang. 2019. Certified robustness to adversarial word substitutions. In *EMNLP*.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? natural language attack on text classification and entailment. In *AAAI*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *AAAI*.

Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi Jaakkola. 2019. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *NeurIPS*.

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 *ACL*.

Jure McAuley, Julian; Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *ACM RecSys*.

Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *NAACL*.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *ICLR*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *ACL*.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. 2019. Provably robust deep learning via adversarially trained smoothed classifiers. In *NeurIPS*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for nlp. In *EMNLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. 2020. Black-box certification with randomized smoothing: A functional optimization based framework. *arXiv preprint arXiv:2002.09169*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

A Appendix

A.1 Bounding the Error of Monte Carlo Estimation

As shown in Proposition 1, the smoothed model f^{RS} is certified robust at an input $\mathbf{X}$ in the sense of (1) if

$$\begin{aligned}\Delta_{\mathbf{X}} &= g^{\mathrm{RS}}(\mathbf{X}, y) - g^{\mathrm{RS}}(\mathbf{X}, y_B) - 2q_{\mathbf{X}} \\ &= g^{\mathrm{RS}}(\mathbf{X}, y) - \max_{c \neq y} g^{\mathrm{RS}}(\mathbf{X}, c) - 2q_{\mathbf{X}} > 0,\end{aligned}$$

where y is the true label of $\mathbf{X}$, and

$$g^{\mathrm{RS}}(\mathbf{X}, c) := \mathbb{P}_{\mathbf{Z} \sim \Pi_{\mathbf{X}}}\left(f(\mathbf{Z}) = c\right) = \mathbb{E}_{\mathbf{Z} \sim \Pi_{\mathbf{X}}}\left[\mathbb{I}\{f(\mathbf{Z}) = c\}\right].$$

Assume $\{\mathbf{Z}^{(i)}\}_{i=1}^n$ is an i.i.d. sample from $\Pi_{\mathbf{X}}$. By Monte Carlo approximation, we can estimate $g^{\mathrm{RS}}(\mathbf{X}, c)$ for all $c \in \mathcal{Y}$ jointly, via

$$\hat{g}^{\mathrm{RS}}(\mathbf{X}, c) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left\{f(\mathbf{Z}^{(i)}) = c\right\},$$

and estimate $\Delta_{\mathbf{X}}$ via

$$\hat{\Delta}_{\mathbf{X}} := \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left\{f(\mathbf{Z}^{(i)}) = y\right\} - \max_{c \neq y} \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left\{f(\mathbf{Z}^{(i)}) = c\right\} - 2q_{\mathbf{X}}.$$

To develop a rigorous procedure for testing $\Delta_{\mathbf{X}} > 0$, we need to bound the non-asymptotic error of the Monte Carlo estimation, which can be done with a simple application of Hoeffding's concentration inequality and union bound.

Proposition 2. *Assume $\{\mathbf{Z}^{(i)}\}$ is i.i.d. drawn from $\Pi_{\boldsymbol{X}}$. For any $\delta \in (0, 1)$, with probability at least $1 - \delta$, we have*

$$\Delta_{\boldsymbol{X}} \geq \hat{\Delta}_{\boldsymbol{X}} - 2\sqrt{\frac{\log \frac{1}{\delta} + \log |\mathcal{Y}|}{2n}}.$$

We can now frame the robustness certification problem into a hypothesis test problem. Consider the null hypothesis H_0 and alternatively hypothesis H_a:

$$\begin{aligned}\mathrm{H}_0 &: \Delta_{\mathbf{X}} \leq 0 \quad (f^{\mathrm{RS}} \text{ is not certified robust to } \mathbf{X}) \\ \mathrm{H}_a &: \Delta_{\mathbf{X}} > 0 \quad (f^{\mathrm{RS}} \text{ is certified robust to } \mathbf{X}).\end{aligned}$$

Then according to Proposition 2, we can reject the null hypothesis H_0 with a significance level δ if

$$\hat{\Delta}_{\mathbf{X}} - 2\sqrt{\frac{\log \frac{1}{\delta} + \log |\mathcal{Y}|}{2n}} > 0.$$

In all the experiments, we set $\delta = 0.01$ and $n = 5000$.

A.2 Proof of the Main Theorems

In this section, we give the proofs of the theorems in the main text.

A.2.1 Proof of Proposition 1

According to the definition of f^{RS}, it is certified robust at $\mathbf{X}$, that is, $y = f^{\mathrm{RS}}(\mathbf{X}')$ for $\forall \mathbf{X}' \in S_{\mathbf{X}}$, if

$$g^{\mathrm{RS}}(\mathbf{X}',\ y) \geq \max_{c \neq y} g^{\mathrm{RS}}(\mathbf{X}', c), \quad \mathbf{X}' \in S_{\mathbf{X}}. \tag{3}$$

Obviously

$$\begin{aligned}g^{\mathrm{RS}}(\mathbf{X}',\ y) - \max_{c \neq y} g^{\mathrm{RS}}(\mathbf{X}', c) &\geq \min_{\mathbf{X}' \in S_{\mathbf{X}}} g^{\mathrm{RS}}(\mathbf{X}',\ y) - \max_{c \neq y} \max_{\mathbf{X}' \in S_{\mathbf{X}}} g^{\mathrm{RS}}(\mathbf{X}', c) \\ &\geq \left(g^{\mathrm{RS}}(\mathbf{X},\ y) - q_{\mathbf{X}}\right) - \max_{c \neq y} \left(g^{\mathrm{RS}}(\mathbf{X}, c) + q_{\mathbf{X}}\right) \quad \text{//by Theorem 1.} \\ &= \Delta_{\mathbf{X}}.\end{aligned}$$

Therefore, $\Delta_{\mathbf{X}} > 0$ must imply (3) and hence certified robustness.

A.2.2 Proof of Theorem 1

Our goal is to calculate the upper and lower bounds $\max_{\mathbf{X}'\sim\Pi_{\mathbf{X}}} g^{\mathrm{RS}}(\mathbf{X}', c)$ and $\min_{\mathbf{X}'\sim\Pi_{\mathbf{X}}} g^{\mathrm{RS}}(\mathbf{X}', c)$. Our key idea is to frame the computation of the upper and lower bounds into a variational optimization.

Lemma 1. *Define $\mathcal{H}_{[0,1]}$ to be the set of all bounded functions mapping from $\mathcal{X}$ to $[0,1]$, For any $h \in \mathcal{H}_{[0,1]}$, define*

$$\Pi_{\boldsymbol{X}}[h] = \mathbb{E}_{\mathbf{Z}\sim\Pi_{\boldsymbol{X}}}[h(\boldsymbol{Z})].$$

Then we have for any $\boldsymbol{X}$ and $c \in \mathcal{Y}$,

$$\min_{\boldsymbol{X}'\sim\Pi_{\boldsymbol{X}}} g^{RS}(\boldsymbol{X}', c) \geq \min_{h\in\mathcal{H}_{[0,1]}} \min_{\boldsymbol{X}'\sim\Pi_{\boldsymbol{X}}} \left\{\Pi_{\boldsymbol{X}'}[h] \quad s.t. \quad \Pi_{\boldsymbol{X}}[h] = g^{RS}(\boldsymbol{X}, c)\right\} := g^{RS}_{low}(\boldsymbol{X}, c),$$
$$\max_{\boldsymbol{X}'\sim\Pi_{\boldsymbol{X}}} g^{RS}(\boldsymbol{X}', c) \leq \max_{h\in\mathcal{H}_{[0,1]}} \max_{\boldsymbol{X}'\sim\Pi_{\boldsymbol{X}}} \left\{\Pi_{\boldsymbol{X}'}[h] \quad s.t. \quad \Pi_{\boldsymbol{X}}[h] = g^{RS}(\boldsymbol{X}, c)\right\} := g^{RS}_{up}(\boldsymbol{X}, c).$$

Proof of Lemma 1. The proof is straightforward. Define $h_0(\mathbf{X}) = \mathbb{I}\{f(\mathbf{X}) = c\}$. Recall that

$$g^{\mathrm{RS}}(\mathbf{X}, c) = \mathbb{P}_{\mathbf{Z}\sim\Pi_{\mathbf{X}}}\left(f(\mathbf{Z}) = c\right) = \Pi_{\mathbf{X}}[h_0].$$

Therefore, h_0 satisfies the constraints in the optimization, which makes it obvious that

$$g^{\mathrm{RS}}(\mathbf{X}', c) = \Pi_{\mathbf{X}'}[h_0] \geq \min_{h\in\mathcal{H}_{[0,1]}} \left\{\Pi_{\mathbf{X}'}[h] \quad s.t. \quad \Pi_{\mathbf{X}}[h] = g^{\mathrm{RS}}(\mathbf{X}, c)\right\}.$$

Taking $\min_{\mathbf{X}'\in S_{\mathbf{X}}}$ on both sides yields the lower bound. The upper bound follows the same derivation. □

Therefore, the problem reduces to deriving bounds for the optimization problems.

Theorem 3. *Under the assumptions of Theorem 1, for the optimization problems in Lemma 1, we have*

$$g^{RS}_{low}(\boldsymbol{X}, c) \geq \max(g^{RS}(\boldsymbol{X}, c) - q_{\boldsymbol{X}},\ 0), \qquad g^{RS}_{up}(\boldsymbol{X}, c) \leq \min(g^{RS}(\boldsymbol{X}, c) + q_{\boldsymbol{X}},\ 1).$$

where $q_{\boldsymbol{X}}$ is the quantity defined in Theorem 1 in the main text.

Now we proceed to prove Theorem 3.

Proof of Theorem 3. We only consider the minimization problem because the maximization follows the same proof. For notation, we denote $p = g^{\mathrm{RS}}(\mathbf{X}, c)$. Applying the Lagrange multiplier to the constraint optimization problem and exchanging the min and max, we have

$$\begin{aligned}
g^{\mathrm{RS}}_{low}(\mathbf{X}, c) =& \min_{\mathbf{X}'\in S_{\mathbf{X}}} \min_{h\in\mathcal{H}_{[0,1]}} \max_{\lambda\in\mathbb{R}} \Pi_{\mathbf{X}'}[h] - \lambda\Pi_{\mathbf{X}}[h] + \lambda p \\
\geq& \max_{\lambda\in\mathbb{R}} \min_{\mathbf{X}'\in S_{\mathbf{X}}} \min_{h\in\mathcal{H}_{[0,1]}} \Pi_{\mathbf{X}'}[h] - \lambda\Pi_{\mathbf{X}}[h] + \lambda p \\
=& \max_{\lambda\in\mathbb{R}} \min_{\mathbf{X}'\in S_{\mathbf{X}}} \min_{h\in\mathcal{H}_{[0,1]}} \int h(\mathbf{Z})\left(d\Pi_{\mathbf{X}'}(\mathbf{Z}) - \lambda d\Pi_{\mathbf{X}}(\mathbf{Z})\right) + \lambda p \\
=& -\max_{\lambda\in\mathbb{R}} \max_{\mathbf{X}'\in S_{\mathbf{X}}} \int \left(\lambda d\Pi_{\mathbf{X}}(\mathbf{Z}) - d\Pi_{\mathbf{X}'}(\mathbf{Z})\right)_+ + \lambda p \\
=& -\max_{\lambda\geq 0} \max_{\mathbf{X}'\in S_{\mathbf{X}}} \int \left(\lambda d\Pi_{\mathbf{X}}(\mathbf{Z}) - d\Pi_{\mathbf{X}'}(\mathbf{Z})\right)_+ + \lambda p.
\end{aligned}$$

Here $d\Pi^0_{\mathbf{X}}(\mathbf{Z})$ and $d\Pi^0_{\mathbf{X}'}(\mathbf{Z})$ is the counting measure and $(s)_+ = \max(s, 0)$. Now we calculate $\int \left(\lambda d\Pi_{\mathbf{X}}(\mathbf{Z}) - d\Pi_{\mathbf{X}'}(\mathbf{Z})\right)_+$.

Lemma 2. *Given x, x', define $n_x = |P_x|$, $n_{x'} = |P_{x'}|$ and $n_{x,x'} = |P_x \cap P_{x'}|$. We have the following identity*

$$\begin{aligned}
&\int \left(\lambda d\Pi_{\boldsymbol{X}}(\mathbf{Z}) - d\Pi_{\boldsymbol{X}'}(\mathbf{Z})\right)_+ \\
=&\lambda\left[1 - \prod_{j\in[L], x_j\neq x'_j} \frac{n_{x_j,x'_j}}{n_{x_j}}\right] + \left[\prod_{j\in[L], x_j\neq x'_j} \frac{n_{x_j,x'_j}}{n_{x_j}}\right]\left[\lambda - \prod_{j\in[L], x_j\neq x'_j} \frac{n_{x_j}}{n_{x'_j}}\right]_+.
\end{aligned}$$

As a result, under the assumption that $n_x = |P_x| = |P_{x'}| = n_{x'}$ for every word x and its synonym $x' \in S_x$, we have

$$\int (\lambda d\Pi_{\boldsymbol{X}}(\mathbf{Z}) - d\Pi_{\boldsymbol{X}'}(\mathbf{Z}))_+ = \lambda \left[1 - \prod_{j\in[L], x_j \neq x'_j} \frac{n_{x_j,x'_j}}{n_{x_j}}\right] + \left[\prod_{j\in[L], x_j \neq x'_j} \frac{n_{x_j,x'_j}}{n_{x_j}}\right](\lambda - 1)_+ .$$

We now need to solve the optimization of $\max_{\mathbf{X}'\in S_{\mathbf{X}}} \int (\lambda d\Pi_{\mathbf{X}}(\mathbf{Z}) - d\Pi_{\mathbf{X}'}(\mathbf{Z}))_+$.

Lemma 3. *For any word x, define $\tilde{x}^* = \arg\min_{x'\in S_x} n_{x,x'}/n_x$. For a given sentence $\boldsymbol{X} = x_1, \ldots, x_L$, we define an ordering of the words $x_{\ell_1}, \ldots, x_{\ell_L}$ such that $n_{x_{\ell_i},\tilde{x}^*_{\ell_i}}/n_{x_{\ell_i}} \leq n_{x_{\ell_j},\tilde{x}^*_{\ell_j}}/n_{x_{\ell_j}}$ for any $i \leq j$. For a given $\boldsymbol{X}$ and R, we define an adversarial perturbed sentence $\boldsymbol{X}^* = x^*_1, \ldots, x^*_L$, where*

$$x^*_i = \begin{cases} \tilde{x}^*_i & \text{if } i \in [\ell_1, \ldots, \ell_R] \\ x_i & \text{if } i \notin [\ell_1, \ldots, \ell_R]. \end{cases}$$

Then for any $\lambda \geq 0$, we have that $\boldsymbol{X}^*$ is the optimal solution of $\max_{\boldsymbol{X}'\in S_{\boldsymbol{X}}} \int (\lambda d\Pi_{\boldsymbol{X}}(\mathbf{Z}) - d\Pi_{\boldsymbol{X}'}(\mathbf{Z}))_+$, that is,

$$\max_{\boldsymbol{X}'\in S_{\boldsymbol{X}}} \int (\lambda d\Pi_{\boldsymbol{X}}(\mathbf{Z}) - d\Pi_{\boldsymbol{X}'}(\mathbf{Z}))_+ = \int (\lambda d\Pi_{\boldsymbol{X}}(\mathbf{Z}) - d\Pi_{\boldsymbol{X}^*}(\mathbf{Z}))_+ .$$

Now by Lemma 3, the lower bound becomes

$$\begin{aligned} g^{\text{RS}}_{low}(\mathbf{X}, c) =& -\max_{\lambda\geq 0} \max_{\mathbf{X}'\in S_{\mathbf{X}}} \int (\lambda d\Pi_{\mathbf{X}}(\mathbf{Z}) - d\Pi_{\mathbf{X}'}(\mathbf{Z}))_+ + \lambda p \\ =& -\max_{\lambda\geq 0} \int (\lambda d\Pi_{\mathbf{X}}(\mathbf{Z}) - d\Pi_{\mathbf{X}^*}(\mathbf{Z}))_+ + \lambda p \\ =& \max_{\lambda\geq 0} (p - q_{\mathbf{X}})\lambda - (1 - q_{\mathbf{X}})(\lambda - 1)_+ \qquad (4) \\ =& \max(p - q_{\mathbf{X}},\ 0), \end{aligned}$$

where $q_{\mathbf{X}}$ is consistent with the definition in Theorem 1:

$$q_{\mathbf{X}} = 1 - \prod_{j\in[L], x_j \neq \tilde{x}^*_j} \frac{n_{x_j,\tilde{x}^*_j}}{n_{x_j}} = 1 - \prod_{j=1}^{R} q_{x_{\ell_j}}.$$

Here equation (4) is by calculation using the assumption of Theorem 1. The optimization of $\max_{\lambda\geq 0}$ in (4) is an elementary step: if $p \leq q$, we have $\lambda^* = 0$ with solution 0; if $p \geq q$, we have $\lambda^* = 1$ with solution $(p - q_{\mathbf{X}})$. This finishes the proof of the lower bound. The proof the upper bound follows similarly. □

Proof of Lemma 2 Notice that we have

$$\begin{aligned} \int (\lambda d\Pi_{\mathbf{X}}(\mathbf{Z}) - d\Pi_{\mathbf{X}'}(\mathbf{Z}))_+ =& \sum_{\mathbf{Z}\in S_{\mathbf{X}'}\cap S_{\mathbf{X}}} \left(\lambda |S_{\mathbf{X}}|^{-1} - |S_{\mathbf{X}'}|^{-1}\right)_+ + \lambda \sum_{\mathbf{Z}\in S_{\mathbf{X}} - S_{\mathbf{X}'}} |S_{\mathbf{X}}|^{-1} \\ =& |S_{\mathbf{X}'} \cap S_{\mathbf{X}}| \left(\lambda |S_{\mathbf{X}}|^{-1} - |S_{\mathbf{X}'}|^{-1}\right)_+ + \lambda |S_{\mathbf{X}} - S_{\mathbf{X}'}| |S_{\mathbf{X}}|^{-1} . \end{aligned}$$

Also notice that $|S_{\mathbf{X}}| = \prod_{j=1}^{L} n_{x_j}$; $|S_{\mathbf{X}'}| = \prod_{j=1}^{L} n_{x'_j}$; $|S_{\mathbf{X}'} \cap S_{\mathbf{X}}| = \prod_{j=1}^{L} n_{x_j,x'_j}$ and $|S_{\mathbf{X}} - S_{\mathbf{X}'}| = \prod_{j=1}^{L} n_{x_j} - \prod_{j=1}^{L} n_{x_j,x'_j}$. Plugging in the above value, we have

$$\begin{aligned} |S_{\mathbf{X}} - S_{\mathbf{X}'}| |S_{\mathbf{X}}|^{-1} =& \frac{\prod_{j=1}^{L} n_{x_j} - \prod_{j=1}^{L} n_{x_j,x'_j}}{\prod_{j=1}^{L} n_{x_j}} \\ =& 1 - \prod_{j=1}^{L} \frac{n_{x_j,x'_j}}{n_{x_j}} \\ =& 1 - \prod_{j\in[L], x_j \neq x'_j} \frac{n_{x_j,x'_j}}{n_{x_j}}. \end{aligned}$$

And also,

$$\begin{aligned}\left(\lambda\left|S_{\mathbf{X}}\right|^{-1}-\left|S_{\mathbf{X}'}\right|^{-1}\right)_{+} &= \left(\lambda\prod_{j=1}^{L} n_{x_j}^{-1}-\prod_{j=1}^{L} n_{x_j'}^{-1}\right)_{+} \\ &= \left(\lambda\prod_{j\in[L],x_j=x_j'} n_{x_j}^{-1}\prod_{j\in[L],x_j\neq x_j'} n_{x_j}^{-1}-\prod_{j\in[L],x_j=x_j'} n_{x_j}^{-1}\prod_{j\in[L],x_j\neq x_j'} n_{x_j'}^{-1}\right)_{+} \\ &= \prod_{j\in[L],x_j=x_j'} n_{x_j}^{-1}\left(\lambda\prod_{j\in[L],x_j\neq x_j'} n_{x_j}^{-1}-\prod_{j\in[L],x_j\neq x_j'} n_{x_j'}^{-1}\right)_{+}.\end{aligned}$$

Plugging in the above value, we have

$$\begin{aligned}\left|S_{\mathbf{X}'}\cap S_{\mathbf{X}}\right|\left(\lambda\left|S_{\mathbf{X}}\right|^{-1}-\left|S_{\mathbf{X}'}\right|^{-1}\right)_{+} &= \prod_{j=1}^{L} n_{x_j,x_j'}\left(\lambda\left|S_{\mathbf{X}}\right|^{-1}-\left|S_{\mathbf{X}'}\right|^{-1}\right)_{+} \\ &= \prod_{j\in[L],x_j=x_j'} n_{x_j}\prod_{j\in[L],x_j\neq x_j'} n_{x_j,x_j'}\left(\lambda\left|S_{\mathbf{X}}\right|^{-1}-\left|S_{\mathbf{X}'}\right|^{-1}\right)_{+} \\ &= \prod_{j\in[L],x_j\neq x_j'} n_{x_j,x_j'}\left(\lambda\prod_{j\in[L],x_j\neq x_j'} n_{x_j}^{-1}-\prod_{j\in[L],x_j\neq x_j'} n_{x_j'}^{-1}\right)_{+} \\ &= \prod_{j\in[L],x_j\neq x_j'} n_{x_j,x_j'}\prod_{j\in[L],x_j\neq x_j'} n_{x_j}^{-1}\left(\lambda-\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j}}{n_{x_j'}}\right)_{+} \\ &= \left(\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j,x_j'}}{n_{x_j}}\right)\left(\lambda-\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j}}{n_{x_j'}}\right)_{+}.\end{aligned}$$

Combining all the calculation, we get

$$\begin{aligned}&\int\left(\lambda d\Pi_{\mathbf{X}}(\mathbf{Z})-d\Pi_{\mathbf{X}'}(\mathbf{Z})\right)_{+} \\ =&\lambda\left[1-\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j,x_j'}}{n_{x_j}}\right]+\left[\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j,x_j'}}{n_{x_j}}\right]\left[\lambda-\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j}}{n_{x_j'}}\right]_{+}.\end{aligned}$$

Proof of Lemma 3 It is sufficient to proof that, for any $\mathbf{X}'\neq\mathbf{X}^*$, we have

$$\int\left(\lambda d\Pi_{\mathbf{X}}(\mathbf{Z})-d\Pi_{\mathbf{X}^*}(\mathbf{Z})\right)_{+}\geq\int\left(\lambda d\Pi_{\mathbf{X}}(\mathbf{Z})-d\Pi_{\mathbf{X}'}(\mathbf{Z})\right)_{+}.$$

Notice that for any $\lambda\geq 0$, define

$$Q(\mathbf{X},\mathbf{X}'')=\lambda\left[1-\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j,x_j''}}{n_{x_j}}\right]+\left[\prod_{j\in[L],x_j\neq x_j'}\frac{n_{x_j,x_j''}}{n_{x_j}}\right](\lambda-1)_{+}.$$

Given any $\mathbf{X}$, we can view $Q(\mathbf{X},\mathbf{X}'')$ as the function of $n_{x_i,x_i''}/n_{x_i}$, $i\in[L]$. And $Q(\mathbf{X},\mathbf{X}'')$ is a decreasing function of $n_{x_i,x_i''}/n_{x_i}$ for any $i\in[L]$ when fixing $\frac{n_{x_j,x_j''}}{n_{x_j}}$ for all other $j\neq i$. Suppose $\tilde{r}_k$ is the k-th smallest quantities of $n_{x_i,\tilde{x}_i^*}/n_{x_i}$, $i\in[L]$ and r_k' is the k-th smallest quantities of $n_{x_j,\tilde{x}_j^*}/n_{x_i}$, $i\in[L]$. By the construction of $\mathbf{X}^*$, we have $\tilde{r}_k\leq r_k'$ for any $k\in[L]$. This implies that

$$Q(\mathbf{X},\mathbf{X}^*)\geq Q(\mathbf{X},\mathbf{X}').$$

A.2.3 Proof of Theorem 2

We denote $g^{\text{RS}}(\mathbf{X}, y) = p_A$, $g^{\text{RS}}(\mathbf{X}, y_B) = p_B$ and $q = q_{\mathbf{X}}$ in this proof for simplicity. The $\mathbf{X}^*$ below is the one defined in the proof of Lemme 3. Our proof is based on constructing a randomized smoothing classifier that satisfies the desired property we want to prove.

Case 1 $p_A \geq q$ **and** $p_B + q \leq 1$ Note that in this case $|S_{\mathbf{X}} \cap S_{\mathbf{X}^*}| / |S_{\mathbf{X}}| = 1 - q \geq (p_A - q) + p_B$, where the inequality is due to $p_A + p_B \leq 1$. Therefore, we can choose set U_1 and U_2 such that $U_1 \subseteq S_{\mathbf{X}} \cap S_{\mathbf{X}^*}$; $U_2 \subseteq S_{\mathbf{X}} \cap S_{\mathbf{X}^*}$; $U_1 \cap U_2 = \emptyset$; $|U_1| / |S_{\mathbf{X}}| = p_A - q$ and $|U_2| / |S_{\mathbf{X}}| = p_B$. We define the classifier:

$$f^*(\mathbf{Z}) = \begin{cases} y & \text{if } \mathbf{Z} \in (S_{\mathbf{X}} - S_{\mathbf{X}^*}) \cap U_1 \\ y_B & \text{if } \mathbf{Z} \in (S_{\mathbf{X}^*} - S_{\mathbf{X}}) \cup U_2 \\ \text{other class } (c \neq y \text{ or } y_B) & \text{if } \mathbf{Z} \in S_{\mathbf{X}} \cap S_{\mathbf{X}^*} - (U_1 \cup U_2) \\ \text{any class } (c \in \mathcal{Y}) & \text{otherwise} \end{cases}$$

This classifier is well defined for binary classification because $S_{\mathbf{X}} \cap S_{\mathbf{X}^*} - (U_1 \cup U_2) = \emptyset$.

Case 2 $p_A < q$ **and** $p_B + q \leq 1$ In this case, we can choose set U_1 and U_2 such that $U_1 \subseteq S_{\mathbf{X}} - S_{\mathbf{X}^*}$; $U_2 \subseteq S_{\mathbf{X}} \cap S_{\mathbf{X}^*}$; $|U_1| / |S_{\mathbf{X}}| = p_A$ and $|U_2| / |S_{\mathbf{X}}| = p_B$. We define the classifier:

$$f^*(\mathbf{Z}) = \begin{cases} y & \text{if } \mathbf{Z} \in U_1 \\ y_B & \text{if } \mathbf{Z} \in U_2 \cup (S_{\mathbf{X}^*} - S_{\mathbf{X}}) \\ \text{other class } (c \neq y \text{ or } y_B) & \text{if } \mathbf{Z} \in S_{\mathbf{X}} - (U_1 \cup U_2) \\ \text{any class } (c \in \mathcal{Y}) & \text{otherwise} \end{cases}$$

This classifier is well defined for binary classification because $S_{\mathbf{X}} - (U_1 \cup U_2) = \emptyset$.

Case 3 $p_A \geq q$ **and** $p_B + q > 1$ This case does not exist since we would have $p_A + p_B > 1$.

Case 4 $p_A < q$ **and** $p_B + q > 1$ We choose set U_1 and U_2 such that $U_1 \subseteq S_{\mathbf{X}} - S_{\mathbf{X}^*}$; $U_2 \in S_{\mathbf{X}} - S_{\mathbf{X}^*}$; $U_1 \cap U_2 = \emptyset$; $|U_1| / |S_{\mathbf{X}}| = p_A$ and $|U_2| / |S_{\mathbf{X}}| = p_B - (1 - q)$. Notice that the intersect of U_1 and U_2 can be empty as $|U_1| / |S_{\mathbf{X}}| + |U_2| / |S_{\mathbf{X}}| = p_A + p_B - (1 - q) \leq 1 - (1 - q) = q = |S_{\mathbf{X}} - S_{\mathbf{X}^*}| / |S_{\mathbf{X}}|$. We define the classifier:

$$f^*(\mathbf{Z}) = \begin{cases} y & \text{if } \mathbf{Z} \in U_1 \\ y_B & \text{if } \mathbf{Z} \in U_2 \cup S_{\mathbf{X}^*} \\ \text{other class } (c \neq y \text{ or } y_B) & \text{if } \mathbf{Z} \in (S_{\mathbf{X}} - S_{\mathbf{X}^*}) - (U_1 \cup U_2) \\ \text{any class } (c \in \mathcal{Y}) & \text{otherwise} \end{cases}$$

This classifier is well defined for binary classification because $S_{\mathbf{X}} - S_{\mathbf{X}^*} - (U_1 \cup U_2) = \emptyset$.

It can be easily verified that for each case, the defined classifier satisfies all the conditions in Theorem 2.

B Additional Experiment Details

We set $R = L$ in adversarial attacking, that is, all words in the sentence can be perturbed simultaneously by the attacker. We use 5,000 random draws in the Monte Carlo estimation of $\Delta_{\mathbf{X}}$, and use the same method in Jia et al. (2019) to tune the hyper-parameters when training the base models e.g. learning rate, batch size and the schedule of loss function. For the IMDB dataset, we train the IBP models and ours for 60 and 10 epochs, respectively. For the Amazon dataset, we train the IBP models and ours for 100 and 20 epochs, respectively.

We test our algorithm on two different datasets, IMDB and Amazon. The IMDB movie review dataset (Maas et al., 2011) is a sentiment classification dataset. It consists of 50,000 movie review comments with binary sentiment labels. The Amazon review dataset (McAuley, 2013) is an extremely large dataset that contains 34,686,770 reviews with 5 different types of labels. Similar to Cohen et al. (2019), we test the models on randomly selected subsets of the test set with 1,250 and 6,500 examples for IMDB and Amazon dataset, respectively.