RANDOMIZED SMOOTHING WITH MASKED INFERENCE FOR ADVERSARIALLY ROBUST NLP SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale pre-trained language models have shown outstanding performance in a variety of NLP tasks. However, they are also known to be significantly brittle against specifically crafted adversarial examples, leading to increasing interest in probing the adversarial robustness of NLP systems. We introduce RSMI, a novel two-stage framework that combines randomized smoothing (RS) with masked inference (MI) to improve the adversarial robustness of NLP systems. RS transforms a classifier into a smoothed classifier to obtain robust representations, whereas MI forces a model to exploit the surrounding context of a masked token in an input sequence. RSMI improves adversarial robustness by **2 to 3 times** over existing state-of-the-art methods on benchmark datasets. We also perform in-depth qualitative analysis to validate the effectiveness of the different stages of RSMI and probe the impact of its components through extensive ablations. By empirically proving the stability of RSMI, we put it forward as a practical method to robustly train large-scale NLP models. Our code and datasets are available at `https://anonymous.4open.science/r/RSMI`.

1 INTRODUCTION

Large-scale pre-trained language models have achieved veritable success on virtually any task in NLP (Devlin et al., 2019; Liu et al., 2019). However, they are also known to be considerably vulnerable to specifically crafted examples, called adversarial examples (Ebrahimi et al., 2018; Jin et al., 2020). The severity of the problem has triggered a variety of methods dedicated to address the adversarial robustness of NLP systems (Goyal et al., 2022; Ye et al., 2020; Zhou et al., 2021; Dong et al., 2021). Formulating an adversarially robust system typically involves a saddle point problem that consists of maximization and minimization of a system's loss function (Madry et al., 2018). The maximization component models a potential threat or adversary, while the minimization component models robustness to the adversary. Existing defense approaches in NLP typically tackle the maximization problem by substituting a target word with its synonym through a heuristic search in a predefined synonym set. This is followed by optimizing the minimization component, which aims at training the system on the perturbed examples until it achieves a certain level of robustness (Dong et al., 2021; Zhou et al., 2021; Ye et al., 2020). However, as shown recently by Li et al. (2021), such synonym-substitution-based defense algorithms generally show a significant performance drop when they have no access to the perturbation sets of the potential attacks. Assuming access to the potential perturbation sets is often unrealistic in practical settings with a deployed NLP system.

To address the adversarial vulnerability of NLP systems, we propose RSMI, a novel two-stage framework that leverages randomized smoothing (RS) and masked inference (MI). *Randomized smoothing* is a generic class of methods that transform a classifier into a smoothed classifier via a randomized input perturbation process (Cohen et al., 2019; Lécuyer et al., 2019). It has come recently into the spotlight due to its simplicity and theoretical guarantee that ensures certifiable robustness within a ball around an input point (Cohen et al., 2019; Salman et al., 2019). Moreover, its robustness enhancement is highly scalable to modern large-scale deep learning setups (Cohen et al., 2019; Lécuyer et al., 2019). These properties render it a promising research direction towards robust and reliable deployment of deep learning systems. However, there exists a non-trivial challenge in introducing RS to NLP systems due to the discrete nature of the text. In our work, we sidestep the issue and adapt RS to NLP problems by injecting noise at the hidden layers of the deep neural model.

The RS stage is followed by a gradient-guided masked inference (MI) to further reinforce the smoothing effect of RS. MI draws an inference on an input sequence via a noise reduction process that masks adversarially "salient" tokens in the input that are potentially perturbed by an attack algorithm. The adversarially salient tokens are achieved via a gradient-based feature attribution analysis rather than random selections as commonly done in pre-training language models (Devlin et al., 2019). The effectiveness of our novel MI can be attributed to several aspects: First, it is a natural regularization for forcing the model to make a prediction based on the surrounding contexts of a masked token in the input (Moon et al., 2021). Second, it works without any prior assumption about potential attacks, which renders it an attack-agnostic defense approach and is more practical in that it requires no sub-module for synonym-substitution. Finally, it has a close theoretical connection to the synonym-substitution-based approaches, as MI can be regarded as a special case of the weighted ensemble over multiple transformed inputs, as we show later in §3.2.

We evaluate the performance of RSMI through comprehensive experimental studies on large-scale pre-trained models trained with three benchmark datasets against TextFooler (Jin et al., 2020), PWWS (Ren et al., 2019) and BAE (Garg & Ramakrishnan, 2020) adversarial attacks. Our empirical observations show that RSMI obtains improvements of **2 to 3 times** against strong adversarial attacks in terms of robustness metrics over state-of-the-art defense methods (§5.1). We also conduct theoretical analysis to validate the effectiveness of our adapted RS (§3.1) and MI (§3.2). We further analyze the scalability of RSMI, the influence of hyperparameters, its impact on the latent representation of the system and its stochastic stability (§5.2). Our theoretical and empirical analyses validate the effectiveness of RSMI presenting it as a potential practical method for training adversarially robust scalable NLP systems.

The main contribution of this work can be summarized as follows:

- We propose a novel defense algorithm RSMI based on randomized smoothing and masked inference, which requires no assumption on potential attacks and no sub-modules for identifying synonyms.
- Our theoretical analyses validate the effectiveness of RSMI (*c.f.,* §3.1 and §3.2).
- We conduct large-scale empirical studies to validate the effectiveness of RSMI and its sub-stages RS and MI, respectively (*c.f.,* §5.1 and §5.2).

2 Background and Related Work

Defense algorithms Defense methods against adversarial attacks typically involve solving a min-max optimization problem, consisting of an *inner maximization* that seeks the worst (adversarial) example and an *outer minimization* that aims to minimize a system's loss over such examples (Madry et al., 2018). The solution to the inner maximization problem can be obtained through iterative optimization algorithms such as stochastic gradient descent. In NLP, the gradients can be computed with respect to word embeddings of an input sequence as done in (Miyato et al., 2016; Zhu et al., 2020; Wang et al., 2021a). An interesting result is that embedding-level adversarial training approaches for text classification tasks often show improvements in standard test accuracy (Zhu et al., 2020; Wang et al., 2021a; Ghaddar et al., 2021). Another prevailing approach is to substitute input words with their synonyms sampled from a pre-defined synonym set. Since many textual attack algorithms perturb input texts at a word level (Ren et al., 2019; Alzantot et al., 2018; Jin et al., 2020), synonym-based defense (SDA) algorithms have emerged as a prominent defense approach (Ye et al., 2020; Zhou et al., 2021; Dong et al., 2021). However, Li et al. (2021) recently pointed out that they tend to **show a significant brittleness** if they have no access to the perturbation sets of the potential attacks.

Textual adversarial attacks In computer vision, iterative optimization-based attacks such as projected gradient descent (PGD) have been considered as the standard attack algorithms for evaluating the robustness of defense algorithms (Athalye et al., 2018a; Madry et al., 2018). However, there is still no general consensus about the standard attack algorithms in NLP. Due to the absence of such standards, defense algorithms in NLP are generally evaluated against multiple attacks based on an assumption that a security hole in a benchmarked defense algorithm can be found via different attack algorithms. Nevertheless, TextFooler (Jin et al., 2020) and Probability Weighted Word Saliency or PWWS (Ren et al., 2019) are the two attack algorithms that have been widely adopted in a variety of defense methods (Dong et al., 2021; Zhou et al., 2021; Wang et al., 2021a; Li et al., 2021). They construct perturbation sets through two different synonym sets (Mrkšić et al., 2016; Fellbaum, 1998)

and identify target token positions via deletion and replacement, respectively. Their popularity can be attributed to their high attack success rate and high quality adversarial examples given a limited number of query budget (Hauser et al., 2021; Li et al., 2021). Another interesting attack approach is to construct a perturbation set via a large-scale pre-trained language model such as BERT (Devlin et al., 2019). BERT-based Adversarial Examples (BAE) (Garg & Ramakrishnan, 2020) aims to generate an adversarial example with better retention of semantics and grammaticality through BERT's perturbation candidate selections.

Randomized smoothing Randomized perturbation has been studied at different granularity (Srivastava et al., 2014; Bishop, 1995). They often provide a better generalization performance by acting as a regularization technique (Ghaddar et al., 2021; Vasiljevic et al., 2016). Especially, it has been shown that training a model with samples perturbed by Gaussian noise is equivalent to regularizing a model with *Tikhonov regularization* (Bishop, 1995; Tikhonov & Arsenin, 1977). In an effort towards devising a scalable robustness improvement algorithm, randomized smoothing has been introduced to modern deep learning setups in some prior studies (Lécuyer et al., 2019; Cohen et al., 2019). Despite its simplicity, it provides certifiable robustness within a radius of a ball around an input point (Lécuyer et al., 2019; Cohen et al., 2019). Its simplicity and theoretical guarantee lead many to extend it to the text domain. However, due to the discrete nature of texts, it had to be re-framed to fit it in the NLP systems. Previous methods (Zhou et al., 2021; Ye et al., 2020) propose to randomly perturb raw input sequences or their latent representation. But, they significantly underperform without an access to the perturbation sets of potential attacks as shown by Li et al. (2021).

3 Randomized Smoothing with Masked Inference (RSMI)

We consider a standard text classification task with a probabilistic model $F_{\boldsymbol{\theta}} : \mathbb{R}^d \rightarrow \mathcal{P}(\mathcal{Y})$, where $\mathcal{P}(\mathcal{Y})$ is the set of possible probability distributions over class labels $\mathcal{Y} = \{1, \ldots, C\}$, and $\boldsymbol{\theta} \in \mathbb{R}^p$ denotes the parameters of the model $F_{\boldsymbol{\theta}}$ (or F for simplicity). The model F is trained to fit a data distribution $\mathcal{D}$ over pairs of an input sequence $s = (s_1, \ldots, s_T)$ of T tokens and its corresponding class label $y \in \mathcal{Y}$. The distributed representation of s (or word embedding) is represented as $x = [x_1, \ldots, x_T]$. We assume the model is trained with a loss function $\mathcal{L}$ such as cross-entropy. We denote the final prediction of the model as $\hat{y} = \arg\max_i F(s)_i$ and the ground truth label as y^*.

3.1 Randomized Smoothing via Noise Layers

Given the model F, our method exploits a *randomized smoothing* (Lécuyer et al., 2019; Cohen et al., 2019) approach to obtain a smoothed version of it, denoted by $G : \mathbb{R}^d \rightarrow \mathcal{P}(\mathcal{Y})$, which is provably robust under isotropic Gaussian noise perturbation δ at an input query u (*e.g.,* an image). This can be expressed as:

Definition 1. *Given an original probabilistic neural network classifier F, the associated smoothed classifier G at a query u can be denoted as (*a.k.a. *Weierstrauss Transform (Ahmed I, 1996)):*

$$G(u) = (F * \mathcal{N}(0, \sigma^2 I))(u) = \mathbb{E}_{\delta \sim \mathcal{N}(0, \sigma^2 I)}[F(u + \delta)] \,. \tag{1}$$

The standard deviation of the Gaussian noise σ is a hyperparameter that controls the robustness/accuracy tradeoff of the resulting smoothed model G. The higher the noise level is, the more robust it will be, while the prediction accuracy may decrease. The asterisk $*$ denotes a convolution operation (Oppenheim et al., 1996) which, for any two functions h and ψ, can be defined as: $h * \psi(x) = \int_{\mathbb{R}^d} h(t)\psi(x - t)dt$. In practice, $G(u)$ can be estimated via Monte-Carlo sampling (Cohen et al., 2019; Salman et al., 2019).

Cohen et al. (2019) showed that the smoothed model G is robust around a query point u within a L_2 radius R given by:

$$R = \frac{\sigma}{2}(\Phi^{-1}(p_a) - \Phi^{-1}(p_b)) \,, \tag{2}$$

where Φ^{-1} is the inverse of the standard Gaussian CDF, p_a and p_b are the probabilities of the two most likely classes a and b, denoted as: $a = \arg\max_{y \in \mathcal{Y}} G(x)_y$ and $b = \arg\max_{y \in \mathcal{Y} \setminus a} G(x)_y$.

As per Eq. (1), a simple approach to obtain G is to perturb the input u by the noise δ and train with it. However, for a textual input, the token sequence cannot be directly perturbed by δ due to the its

discrete nature. To deviate from the issue, we inject noise at the hidden layers of the model to achieve stronger smoothness. For a given layer f_l, a noise layer f_l^δ draws a noise $\delta \sim \mathcal{N}(0, \sigma^2 I)$ and adds it to the output of f_l in every forward pass of the model. The stronger smoothness resulting from the multi-layer noise is provably guaranteed by the following theorem:

Theorem 1. *Let $F : \mathbb{R}^d \to \mathcal{P}(\mathcal{Y})$ be any soft classifier which can be decomposed as $F = f_1 \circ f_2 \circ \cdots \circ f_L$ and $G = g_1 \circ g_2 \circ \cdots \circ g_L$ be its associated smoothed classifier, where $g_l(x) = (f_l * N(0, \sigma_l^2 I))(x)$ with $1 \leq l \leq L$ and $\sigma_l > 0$. Let $a = \arg\max_{y \in \mathcal{Y}} G(x)_y$ and $b = \arg\max_{y \in \mathcal{Y} \setminus a} G(x)_y$ be two most likely classes for x according to G. Then, we have that $\arg\max_{y \in \mathcal{Y}} G(x')_y = a$ for x' satisfying*

$$\|x' - x\|_2 \leq \frac{1}{2\sigma_1} \prod_{l=2}^{L} (1 + \sigma_l^2)(\Phi^{-1}(p_a) - \Phi^{-1}(p_b)) \,.$$

We provide a proof of the theorem with Lipschitz continuity in Appendix A considering a case when the noise is injected into the word embeddings.

3.2 Gradient-guided Masked Inference

For an input sequence s, our method attempts to "denoise" its adversarially perturbed counterpart s' by attributing saliency of input tokens through a simple gradient-based attribution analysis. Due to the discrete nature of tokens, we compute the gradients of the loss function $\mathcal{L}$ with respect to the word embeddings x_t. The loss is computed with respect to the labels y, which is set to be the ground-truth labels y^* during training and model predictions $\hat{y}$ during inference. Formally, the gradients $\mathbf{g}_t$ for a token $s_t \in s$ (correspondingly $x_t \in x$) can be computed as follows:

$$\mathbf{g}_t = \nabla_{x_t} \mathcal{L}(G(x), y) \approx -\nabla_{x_t} \left(\log \left(\frac{1}{\nu} \sum_{i=1}^{\nu} G(x + \delta_i) \right) \right). \tag{3}$$

Eq. (3) exploits a Monte-Carlo approximation to estimate the gradient $\mathbf{g}_t$ as done in (Salman et al., 2019). Subsequently, the amount of stimulus of the input tokens toward the model prediction is measured by computing the L_2-norm of $\mathbf{g}_t$, *i.e.,* $||\mathbf{g}_t||_2$. The stimulus is regarded as the *saliency score* of the tokens and they are sorted in descending order of the magnitude (Li et al., 2016). Then, we sample M tokens from the top-N tokens in s, and mask them to generate a masked input sequence $m = [s_1, \ldots, m_t, \ldots, s_T]$, where t is the position of a salient token and m_t is the mask token, [MASK]. In the training stage, we mask the top-M positions (i.e., $N = M$), while the mask token selection procedure is switched to a sampling-based approach during inference as detailed later in §3.3. Finally, the gradients $\mathbf{g}_t$ computed for generating the masked sequence is repurposed for perturbing the word embeddings x_t (i.e., $\delta = \mathbf{g}_t$) to obtain robust embeddings as shown previously in (Zhu et al., 2020; Wang et al., 2021a; Miyato et al., 2017).

Our gradient-guided masked inference offers several advantages. First, it yields a natural regularization for forcing the model to exploit surrounding contexts of a masked token in the input (Moon et al., 2021). Second, the masking process can provide a better smoothing effect by masking 'salient' tokens that are potentially adversarial perturbations generated by an attack algorithm. In such cases, it works as a denoising process for reducing the strength of an attacks. In Appendix A.6, we discuss the denoising effect of the gradient-guided masked inference in terms of Lipschitiz continuity of a soft classifier.

Connection to synonym-based defense methods: Another interesting interpretation is that the masked inference has a close connection to the synonym-based defense methods (Wang et al., 2021b; Ye et al., 2020; Wang & Wang, 2020; Zhou et al., 2021). Assuming only position in s is masked and treating the mask as a latent variable $\tilde{s}_t$ that could take any token from the vocabulary V, we can express the masked inference as:

$$p(y|m) = \sum_{\tilde{s}_t \in V} p(y, \tilde{s}_t|m) = \sum_{\tilde{s}_t \in V} p(y|m, \tilde{s}_t) p(\tilde{s}_t|m) \approx \sum_{\tilde{s}_t \in V_t} p(y|m, \tilde{s}_t) p(\tilde{s}_t|m) \,, \tag{4}$$

where $|V_t| \ll |V|$ is the number of words to be at position t with a high probability mass. As shown in the equation, the masked inference can be factorized into a classification objective and a masked language modeling objective, which can be further approximated into a weighted ensembled inference

Algorithm 1 Training and prediction procedure of RSMI.

Initialize. s, M, N, σ, ν, k_0, k_1, α, step size η, and a gradient scale parameter β.
Compute $L := \left[||\mathbf{g}_1||_2, \cdots, ||\mathbf{g}_T||_2\right]$ via Eq. (3). ▷ Gradients *w.r.t.* word embeddings
Sort L in descending order and keep top-N items
Get a masked sequence m by masking top-M tokens based on L.
if Training **then**
    $x := x + \beta(\mathbf{g}_1, \cdots, \mathbf{g}_T)$ ▷ Noise to word embeddings
    $\boldsymbol{\theta} := \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(G(x), y^*)$
else if Prediction **then**
    $\boldsymbol{\phi}(m)_0 = \sum_{i=1}^{k_0} [\mathbb{I}(\hat{y}^{(i)}(m) = y_1), \cdots, \mathbb{I}(\hat{y}^{(i)}(m) = y_c)]$ ▷ First vote
    $n_a = \max \boldsymbol{\phi}(m)_0$
    p-value = BINOMTEST($n_a, k_0, 0.5$,one-tail)
    if p-value $> \alpha$ **then**
        Return $\arg\max_{y \in \mathcal{Y}} \boldsymbol{\phi}(m)_0$
    else
        $[m^{(1)}, \cdots, m^{(k_1)}] \sim$ RANDGRADMASK(k_1, L)
        $\boldsymbol{\phi}(m)_1 = \sum_{i=1}^{k_1} [\mathbb{I}(\hat{y}(m^{(i)}) = y_1), \cdots, \mathbb{I}(\hat{y}(m^{(i)}) = y_c)]$ ▷ Second vote
        Return $\arg\max_{y \in \mathcal{Y}} \boldsymbol{\phi}(m)_1$
    end if
end if

with $|V_t|$ substitutions of s with the highly probable tokens (e.g., synonyms) corresponding to the original word s_t. If we assume $p(\tilde{s}_t|m)$ to be a probability of sampling uniformly from a synonym set such as the one from the WordNet (Fellbaum, 1998), then the masked inference is reduced to a synonym-substitution based defense approach with no necessity of an explicit synonym set.

3.3 TWO-STEP MONTE-CARLO SAMPLING FOR EFFICIENT INFERENCE

The prediction procedure of RSMI involves averaging predictions of k Monte-Carlo samples of $G(m)$ to deal with the variations that come from the noise layers. A large number of k is typically required for a robust prediction but the computational cost increases proportionally as k gets larger. To alleviate the computational cost, we propose a two-step sampling-based inference (Alg. 1).

In the first step, we make k_0 predictions by estimating $G(m)$ for k_0 times (k_0 forward passes). We then make an initial guess about the label of the masked sample m by taking a majority vote of the predictions. Following Cohen et al. (2019), this initial guess is then tested by a one-tailed binomial test with a significance level of α. If the guess passes the test, we return the most probable class based on the vote result. If it fails, then we attempt to make a second guess with a set of k_1 masked input sequences $\mathcal{M} = [m^{(1)}, \cdots, m^{(k_1)}]$. Note that the masked input m used in the first step is generated by masking the top-M tokens from the top-N candidates as we do during training. However, in the second step, we randomly sample M masking positions from the N candidates to create each masked sequence $m^{(i)}$ of $\mathcal{M}$ in order to maximize variations in the predictions; this step is denoted as RANDGRADMASK in Alg. 1.

Our two-step sampling based inference is based on an assumption that textual adversarial examples are liable to fail to achieve consensus from the RSMI's predictions compared to clean samples. In other words, it is considerably harder for adversarial examples to estimate the optimal perturbation direction towards the decision boundary of a stochastic network (Däubener & Fischer, 2022; Athalye et al., 2018a; Cohen et al., 2019).

4 EXPERIMENT SETUP

Datasets Table 1 summarizes the statics of benchmarking datasets adopted in our experiments. We evaluate RSMI on two conventional NLP tasks: text CLaSsification (CLS) and Natural Language Inference (NLI). We adopt IMDB (Maas et al., 2011) and

Table 1: A summary of the datasets.

Dataset	Train	Dev	Test	# Classes
IMDb	22.5k	2.5k	25k	2
AG	108k	12k	7.6k	4
QNLI	105k	5.5k	5.5k	2

AG'S NEWS (Zhang et al., 2015) datasets for the classification task. The IMDB contains movie reviews labeled with positive or negative sentiments. The AG'S NEWS (AG) dataset consists of news articles collected from more than 2K news sources and the samples are grouped into four coarse-grained topic classes. For NLI, we compare the defense algorithms on the Question-answering NLI (QNLI) dataset, which is a part of the GLEU benchmark (Wang et al., 2018). In QNLI, a model is asked to determine an entailment relationship between a pair of a question and a paragraph that may contain the answers. We build development sets for IMDB, AG, and QNLI by randomly drawing 10% samples from each training set via a stratified sampling strategy. For QNLI, we use the original dev set for evaluataions.

Evaluation metrics The performance of defense algorithms is evaluated in terms of four different metrics as proposed in (Li et al., 2021): (*i*) Standard accuracy (SAcc) is the model's accuracy on clean samples. (*ii*) Robust accuracy (RAcc) measures the model's robustness against adversarial attacks. (*iii*) Attack success rate (ASR) is the ratio of the inputs that successfully fool the victim models. (*iv*) Finally, the average number of queries (AvgQ) needed to generate the adversarial examples.

Baselines We select FreeLB (Zhu et al., 2020), InfoBERT (Wang et al., 2021a), and SAFER (Ye et al., 2020) as baselines and apply them over BERT-base (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019) models. We train the baseline algorithms according to their default configurations mentioned in the respective papers. We run them three times with a different random initialization and choose the best model to reproduce the robustness evaluation results in (Li et al., 2021).

4.1 TEXTUAL ADVERSARIAL ATTACKS

We generate adversarial examples via TextFooler (TF) (Jin et al., 2020), Probability Weighted Word Saliency (PWWS) (Ren et al., 2019) and BERT-based Adversarial Examples (BAE) (Garg & Ramakrishnan, 2020). These attack algorithms are widely adopted in a variety of defense work as adversarial robustness benchmarking algorithms (Yoo & Qi, 2021; Dong et al., 2021). TF identifies a target token via cosine similarity between counter-fitting word embeddings (Mrkšić et al., 2016) of candidate tokens. PWWS is a greedy attack algorithm that samples a candidate perturbation from WordNet (Fellbaum, 1998). BAE adopts a pre-trained BERT for perturbing target tokens. For our experiments, we adopted a replacement-based attack model called BAE-R[1].

We randomly draw 1,000 samples from each test set following (Dong et al., 2021; Li et al., 2021; Ye et al., 2020) and perturb them via an attack to generate the corresponding adversarial examples for all experiments unless stated otherwise. We implement all attacks through the publicly available TextAttack library (Morris et al., 2020) and use their default configurations.

For robustness evaluation of RSMI against the attacks, we modify the second step of the two-step inference to make a final decision by averaging logit scores of k_1 Monte-Carlo samples instead of the majority voting approach in Alg. 1. We do this to prevent obfuscating the perturbation processes of TF and PWWS that are devised to identify target tokens via the change of the model's confidence, which can give a false impression about the robustness of RSMI. Nonetheless, we investigate the effectiveness of majority voting based inference as a practical defense method in §5.2. Further details about the attack algorithms and parameter settings of the algorithms are provided in Appendix B.

5 RESULTS AND ANALYSIS

5.1 ADVERSARIAL ROBUSTNESS COMPARISON

We empirically compare the performance of RSMI with the baselines in Table 2. We observe that the standard fine-tuned models are extremely vulnerable to adversarial attacks, as also shown in previous work (Li et al., 2021). Overall, we observe that **RSMI** outperforms all the baselines by quite a large margin across the majority of the metrics such as RAcc, ASR and AvgQ. In particular, it achieves about **2 to 3 times** improvements against strong attack algorithms (*e.g.,* TextFooler and PWWS) in terms of ASR and RAcc, which are key metrics for evaluating the robustness of defense algorithms. RSMI also outperforms other existing methods such as TAVAT (Li & Qiu, 2020), MixADA (Si et al.,

[1]We observe that BAE significantly changes the semantics of the original input compared to TF and PWWS.

Table 2: Performance comparison of adversarial robustness of RSMI with the baselines. RSMI-NoMask excludes masking during inference time. Avg. stands for an average of evaluation results.

Dataset	Model	SAcc (↑)	RAcc (↑)				ASR (↓)				AvgQ (↑)			
			TF	PWWS	BAE	Avg.	TF	PWWS	BAE	Avg.	TF	PWWS	BAE	Avg.
IMDb	BERT-base													
	+ FineTuned	90.60	5.90	0.60	27.30	11.27	93.49	99.34	69.87	87.57	440	1227	377	681
	+ FreeLB (Zhu et al., 2020)	92.90	10.50	14.22	45.30	23.34	88.70	84.71	51.24	74.88	909	1400	442	917
	+ InfoBERT (Wang et al., 2021a)	**92.90**	26.40	26.80	50.00	34.40	71.58	71.15	46.18	62.97	1079	1477	458	1005
	+ SAFER (Ye et al., 2020)	91.80	23.80	30.90	38.20	30.97	74.08	66.34	58.39	66.27	1090	1504	618	1071
	+ RSMI-NoMask (Our)	91.00	34.90	57.00	58.40	50.10	61.65	37.36	35.83	44.95	1395	1733	817	1315
	+ RSMI (Our)	92.20	**56.40**	**58.70**	**80.20**	**65.10**	**38.83**	**36.34**	**13.02**	**29.40**	**1651**	**1764**	**1287**	**1567**
	RoBERTa-base													
	+ FineTuned	93.10	0.50	1.10	22.60	8.07	99.46	98.82	75.73	91.34	588	1248	398	745
	+ FreeLB (Zhu et al., 2020)	93.20	17.30	21.20	49.50	29.33	81.44	77.25	46.89	68.53	999	1433	461	964
	+ InfoBERT (Wang et al., 2021a)	**94.00**	7.60	13.20	36.10	18.97	91.92	85.96	61.60	79.83	855	1388	418	887
	+ SAFER (Ye et al., 2020)	93.20	31.80	39.20	45.40	38.80	65.88	57.94	51.29	58.37	1276	1575	678	1176
	+ RSMI-NoMask (Our)	93.30	47.00	54.00	52.10	51.03	49.63	42.12	44.16	45.30	1455	1684	764	1301
	+ RSMI (Our)	93.00	**73.40**	**76.20**	**83.00**	**77.53**	**21.08**	**18.07**	**10.75**	**16.63**	**1917**	**1863**	**1314**	**1698**
AGNews	BERT-base													
	+ FineTuned	93.90	16.80	34.00	81.00	43.93	82.11	63.79	13.74	53.21	330	352	124	269
	+ FreeLB (Zhu et al., 2020)	**95.00**	24.40	48.20	84.10	52.23	74.32	49.26	11.47	45.02	383	367	131	294
	+ InfoBERT (Wang et al., 2021a)	94.81	19.90	40.90	84.90	48.57	79.01	56.86	10.45	48.77	371	365	126	287
	+ SAFER (Ye et al., 2020)	93.70	46.30	64.00	80.00	63.43	50.59	31.70	14.62	32.30	447	379	170	332
	+ RSMI-NoMask (Our)	92.60	60.40	75.30	77.90	71.20	34.77	18.68	15.88	23.11	497	395	203	365
	+ RSMI (Our)	92.70	**63.20**	**76.10**	**86.10**	**75.13**	**31.82**	**17.91**	**7.12**	**18.95**	**503**	**397**	**573**	**491**
	RoBERTa-base													
	+ FineTuned	93.91	23.90	49.30	80.00	51.07	74.55	47.50	14.80	45.62	353	367	130	283
	+ FreeLB (Zhu et al., 2020)	**95.11**	23.90	48.20	83.00	51.70	74.87	49.32	12.73	45.64	393	374	127	298
	+ InfoBERT (Wang et al., 2021a)	94.00	30.20	52.30	79.80	54.10	67.87	44.36	15.11	42.45	396	374	134	301
	+ SAFER (Ye et al., 2020)	93.60	49.30	68.90	81.60	66.60	47.33	26.39	12.82	28.85	452	386	172	337
	+ RSMI-NoMask (Our)	94.10	66.40	79.00	82.80	76.07	29.44	16.05	12.01	19.17	504	396	213	371
	+ RSMI (Our)	94.30	**74.10**	**81.90**	**88.60**	**81.53**	**21.42**	**13.15**	**6.04**	**13.54**	**530**	**401**	**576**	**502**
QNLI	BERT-base													
	+ FineTuned	79.50	13.20	26.40	52.10	30.57	83.40	66.79	34.47	61.55	189	218	97	168
	+ FreeLB (Zhu et al., 2020)	85.60	13.10	28.00	54.50	31.87	84.70	67.29	36.33	62.77	191	221	96	169
	+ InfoBERT (Wang et al., 2021a)	84.40	18.50	30.50	54.60	34.53	78.08	63.86	35.31	59.08	201	219	97	172
	+ SAFER (Ye et al., 2020)	88.30	33.70	43.00	52.00	42.90	61.83	51.14	41.11	51.36	229	221	109	186
	+ RSMI-NoMask (Our)	**90.60**	30.60	42.40	53.40	42.13	66.23	53.20	41.06	53.50	222	223	117	187
	+ RSMI (Our)	90.57	**41.20**	**54.20**	**60.10**	**51.83**	**54.51**	**40.15**	**33.52**	**47.73**	**256**	**230**	**120**	**202**
	RoBERTa-base													
	+ FineTuned	91.90	19.80	34.00	51.20	35.00	78.48	62.93	44.35	61.92	189	217	91	166
	+ FreeLB (Zhu et al., 2020)	**92.10**	27.30	37.70	55.70	40.23	70.36	59.07	39.52	56.32	215	223	95	178
	+ InfoBERT (Wang et al., 2021a)	91.60	23.00	36.50	53.80	37.77	74.89	60.15	41.27	58.77	204	221	92	172
	+ SAFER (Ye et al., 2020)	90.80	33.80	45.50	49.70	43.00	62.82	49.67	45.26	52.58	232	227	109	189
	+ RSMI-NoMask (Our)	91.50	34.10	46.80	50.50	43.80	62.73	48.73	45.86	52.44	218	225	113	185
	+ RSMI (Our)	91.81	**49.00**	**60.10**	**60.60**	**56.57**	**46.63**	**34.54**	**34.05**	**38.41**	**266**	**240**	**330**	**279**

2020), A2T (Yoo & Qi, 2021), and ASCC (Dong et al., 2021) which we do not report in Table 2 as they show lower performance than our baselines, *c.f.,* Li et al. (2021).[2]

The strong performance of RSMI can be attributed to four factors: (*i*) A provable robustness guarantee by the randomized smoothing approach helps attain higher robustness. To support this claim, we evaluate the robustness of RSMI without the proposed masking process during inference and the results are reported as RSMI-NoMask in Table 2. As we can see, RSMI-NoMask outperforms the baselines in most experiment scenarios. (*ii*) The randomized smoothing denoises adversarial perturbations in the latent space of systems. Our experiments in Appendix C bolster this claim by showing the significant noise reduction in hidden representations of RSMI. (*iii*) The gradient-guided masked inference leads to a reduction in the noise of the adversarial perturbations. This claim can be strongly supported again by comparing the results of RSMI with and without the masking strategy during inference (*c.f.,* RSMI-NoMask and RSMI in Table 2). (*iv*) The two-step sampling-based inference makes it harder for the attack algorithms to estimate the optimal perturbation direction to fool the model for an input sample. In Appendix F, we show the effectiveness of the two-step sampling by comparing the performance of RSMI with and without the two-step sampling.

5.2 Closer Analysis

Majority voting-based inference The inference of RSMI involves a combination of individual predictions. During evaluations in §5.1, RSMI is modified to draw a final decision about an input

[2]Also Li et al. (2021) put constraints to make the attack algorithms weaker which we did not do in our work.

Table 3: Performance of the majority-voting based inference of RSMI. The round brackets next to each number denote the change of score compared to logit averaging based inference.

Dataset	Model	SAcc (↑)	RAcc (↑)		ASR (↓)		AvgQ (↑)	
			TF	PWWS	TF	PWWS	TF	PWWS
IMDb	BERT-base	91.70(-0.50)	77.20(+20.80)	77.80(+19.10)	15.81(-23.02)	15.16(-21.18)	1989(+338)	1877(+113)
	RoBERTa-base	94.30(+1.3)	81.90(+8.50)	82.70(+6.50)	13.15(-7.93)	12.30(-5.77)	2031(+114)	1916(+53)
AGNews	BERT-base	92.90(+0.20)	85.40(+22.20)	87.20(+11.10)	8.07(-23.75)	6.14(-11.77)	572(+69)	408(+11)
	RoBERTa-base	94.10(-0.20)	86.10(+12.00)	88.40(+6.50)	8.50(-12.91)	6.06(-7.08)	571(+41)	407(+6)
QNLI	BERT-base	90.00(-0.57)	63.60(+22.40)	71.30(+17.10)	29.33(-25.18)	20.78(-19.37)	321(+65)	246(+16)
	RoBERTa-base	91.89(+0.08)	68.00(+19.00)	76.40(+16.30)	26.00(-20.63)	16.86(-17.68)	329(+63)	251(+11)

Table 4: Performance comparison of adversarial robustness of RSMI on large-scale pre-trained masked language models. The round brackets next to each number denote the change of score compared to its base-model.

Dataset	Model	SAcc (↑)	RAcc (↑)	ASR (↓)	AvgQ (↑)
IMDb	BERT-large + FineTuned	92.50(+1.90)	29.50(+23.60)	68.11(-25.38)	1130(+690)
	BERT-large + RSMI (Our)	93.16(+0.96)	79.30(+22.90)	14.88(-23.95)	1980(+329)
	RoBERTa-large + FineTuned	94.30(+1.20)	21.20(+20.70)	77.52(-21.94)	1034(+446)
	RoBERTa-large + RSMI (Our)	95.06(+2.06)	87.40(+14.00)	8.06 (-13.02)	2092(+175)
AGNews	BERT-large + FineTuned	95.30(+1.40)	20.60(+3.80)	78.38(-3.73)	358 (+28)
	BERT-large + RSMI (Our)	94.60(+1.90)	85.70(+22.50)	9.41 (-22.41)	568 (+65)
	RoBERTa-large + FineTuned	94.06(+0.15)	41.50(+17.60)	55.88(-18.67)	433 (+80)
	RoBERTa-large + RSMI (Our)	94.60(+0.30)	88.10(+14.00)	6.87 (-14.55)	577 (+47)

sequence by averaging logit scores of multiple Monte-Carlo samples for a fair comparison, because the majority voting obfuscates the perturbation processes of attack algorithms by hiding model prediction scores. However, the majority voting-based inference (*c.f.,* Alg.1 and §3.3) can be a practical defense method against adversarial attacks that require access to a victim model's prediction probabilities for their perturbation process since most attack algorithms require the prediction information (*e.g.,* TextFooler, PWWS, and BAE). To validate the effectiveness of the majority vote, we conduct additional experiments. As shown in Table 3, the majority voting-based inference significantly outperforms the logit averaging approaches.

Large scale parameterization and adversarial robustness We investigate the scalability of RSMI and the impact of model size on robustness. To this end, we conduct experiments on large-scale pre-trained masked language models. During the experiments, RoBERTa-large and BERT-large models are adopted for each task and adversarial examples are generated via TextFooler attack algorithm. Table 4 summarizes our experiment results. The fine-tuned large-scale models show significantly improved robustness and standard task performance compared to their smaller version, RoBERTa-base and BERT-base. We also observe the significant improvements of robustness and standard task performance in the large models with RSMI.

Impact of standard deviation for noise size, number of masks, and number of noise layers. Table 5 shows the impact of different hyperparameters of RSMI. As observed, the overall ASR tends to decrease as we inject more noises into models by increasing noise size (σ), replacing more input words with the mask token (M), and adding more noise layers (N_l). Specifically, we observe that ASR gradually decreases as we put more noise layers in the model. Also, ASR steadily declines as we increase the standard deviation of noise. Finally, we observe that the increased number of masks effectively decreases ASR. However, we observe that these improvements in ASR come at the cost of a decreasing SAcc.

Table 5: Study on noise size (σ), number of masks (M), and number of noise layers (N_l).

σ	M	N_l	ASR(↓)	SAcc(↑)
0.2	2	3	35.26	93.15
0.2	2	4	32.73	**93.26**
0.2	2	5	**29.90**	93.11
0.2	2	3	35.26	**93.15**
0.3	2	3	26.05	92.96
0.4	2	3	**24.17**	92.70
0.4	2	3	24.17	**92.70**
0.4	3	3	22.39	92.49
0.4	4	3	**20.99**	92.13

Effectiveness of gradient-guided masking We probe the effectiveness of the gradient-guided masking strategy by ablating the noise layers and the two-step sampling of RSMI. The resulting model, namely the gradient-guided masking (GM) is compared to a model trained on randomly masked inputs, namely the random masking model (RM). Note that GM predicts and masks inputs in a deterministic way due to the absence of noise layers. Table 6 summarizes our study of ASR changes of GM and RM over different number of mask tokens M in an input sequence as well as k randomly masked sequences drawn for estimating an expectation of RM prediction. RM tends to show a similar pattern that is observed in the experiment about the variation of SAFER in Appendix F. Specifically, it tends to achieve its best performance at $k = 1$, but shows vulnerability as k increases. On the other hand, ASR of GM tends to decrease as we increase M, **validating the effectiveness of gradient-guided masking for denoising adversarial perturbations** injected by attack algorithms.

Table 6: ASR of random masking (RM) and gradient-guided masking (GM) for combinations of M masked tokens and k masked sequences.

	Model	ASR(↓)			
		$k=1$	$k=5$	$k=10$	$k=50$
$M=1$	RM	86.65	97.95	99.68	100
	GM	96.55			
$M=2$	RM	76.57	90.84	93.11	96.27
	GM	83.12			
$M=3$	RM	70.05	86.34	90.08	92.76
	GM	90.81			
$M=4$	RM	65.34	78.86	79.74	82.64
	GM	81.34			
$M=5$	RM	68.35	86.31	90.70	96.26
	GM	80.72			

Run time analysis We compare the computation speed of RSMI with the baselines on the RoBERTa-base model fine-tuned on QNLI. All experiments are conducted on an Intel Xeon Gold 5218R CPU-2.10GHz processor with a single Quadro RTX 6000 GPU. For a fair comparison, the number of gradient computation steps of FreeLB and InfoBERT is set to 3 and other parameters are configured to the default settings provided by the original papers. Also, we do not include the preprocessing time of SAFER. As shown in Table 7, RSMI is approximately 1.9x slower than the Fine-Tuned model during training and 3.5x slower during inference. The latency of RSMI is mainly caused by the additional backpropagation and forward propagation for computing the gradients. The inference speed of RSMI can be improved by removing the masking step during inference, but there exist a trade-off between the inference speed and robustness as shown in Table 2.

Table 7: Run time comparison of RSMI with the baselines.

Dataset	Model	Train (↓)	Inference (↓)
QNLI	FineTuned	1.0	1.0
	FreeLB(Zhu et al., 2020)	×2.8	×1.0
	InfoBERT(Wang et al., 2021a)	×5.4	×1.0
	SAFER(Ye et al., 2020)	×1.0	×1.0
	RSMI NoMask (Our)	×1.9	×1.0
	RSMI (Our)	×1.9	×3.5

6 Conclusion, Limitations, and Future Work

We have proposed RSMI, a novel two-stage framework to tackle the issue of adversarial robustness of large-scale deep NLP systems. To this end, we first adapted the randomized smoothing (RS) strategy for discrete text inputs. We followed it up by devising a novel gradient-guided masked inference (MI) approach that reinforces the smoothing effect of RS. We have evaluated RSMI by applying it to large-scale pre-trained models on three benchmark datasets and obtain 2 to 3 times improvements against strong attacks in terms of robustness evaluation metrics over state-of-the-art defense methods. We have also studied the scalability of RSMI and performed extensive qualitative analyses to examine the effect of RSMI on the latent representations of the original and perturbed inputs as well as the change in its stability owing to its non-deterministic nature. Our thorough ablations and experiments validate the effectiveness of RSMI as a practical approach to train adversarially robust NLP systems.

A major component of RSMI has been developed with the concept of randomized smoothing which is known to be certifiably robust within a radius of a ball around an input point. Though we have proved the robustness for the perturbed samples within this given ball, there is no theoretical guarantee that a perturbed sample will always lie within the ball. Accordingly, our study is limited to empirical validation of the effectiveness of RSMI, although it has theoretical robustness within a L_2 norm ball as shown in §3. Nevertheless, certified robustness is a critical research direction for robust and reliable deployment of NLP systems. In our future work, we will explore the theoretical understanding of the certified-robustness of NLP systems and textual adversarial examples in-depth.

REFERENCES

Zayed Ahmed I. *Handbook of Function and Generalized Function Transformations*. CRC Press, London, 1996.

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

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018a.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293. PMLR, 10–15 Jul 2018b. URL https://proceedings.mlr.press/v80/athalye18b.html.

Chris M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995. doi: 10.1162/neco.1995.7.1.108.

Paul Bromiley. Products and convolutions of gaussian probability density functions. *Tina-Vision Memo*, 3(4):1, 2003.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cohen19c.html.

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

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*, 2021.

Sina Däubener and Asja Fischer. How sampling impacts the robustness of stochastic neural networks, 2022. URL https://arxiv.org/abs/2204.10839.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 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)*, pp. 31–36, Melbourne, Australia, July 2018. Association for Computational Linguistics.

Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

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

Abbas Ghaddar, Philippe Langlais, Ahmad Rashid, and Mehdi Rezagholizadeh. Context-aware Adversarial Training for Name Regularity Bias in Named Entity Recognition. *Transactions of the Association for Computational Linguistics*, 9:586–604, 07 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00386. URL https://doi.org/10.1162/tacl_a_00386.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. A survey in adversarial defences and robustness in nlp, 2022. URL https://arxiv.org/abs/2203.06414.

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

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/a2b15837edac15df90721968986f7f8e-Paper.pdf.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8018–8025, Apr. 2020. doi: 10.1609/aaai.v34i05.6311. URL https://ojs.aaai.org/index.php/AAAI/article/view/6311.

Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pp. 656–672. IEEE, 2019. doi: 10.1109/SP.2019.00044. URL https://doi.org/10.1109/SP.2019.00044.

Jerry Li. Cse 599-m, lecture notes of robustness in machine learning, November 2019. URL https://jerryzli.github.io/robust-ml-fall19/lec14.pdf.

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

Linyang Li and Xipeng Qiu. Tavat: Token-aware virtual adversarial training for language understanding, 2020. URL https://arxiv.org/abs/2004.14543.

Zongyi Li, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3137–3147, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.251. URL https://aclanthology.org/2021.emnlp-main.251.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification, 2016. URL http://arxiv.org/abs/1605.07725. cite arxiv:1605.07725Comment: Published as a conference paper at ICLR 2017.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. In *International Conference on Learning Representations*, 2017.

Seung Jun Moon, Sangwoo Mo, Kimin Lee, Jaeho Lee, and Jinwoo Shin. Masker: Masked keyword regularization for reliable text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13578–13586, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17601.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–148. Association for Computational Linguistics, June 2016.

Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & Systems*. Prentice-Hall, Inc., USA, 1996. ISBN 0138147574.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

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

Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sébastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. Better robustness by more coverage: Adversarial training with mixup augmentation for robust fine-tuning. *CoRR*, abs/2012.15699, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Charles M. Stein. Estimation of the Mean of a Multivariate Normal Distribution. *The Annals of Statistics*, 9(6):1135 – 1151, 1981. doi: 10.1214/aos/1176345632. URL https://doi.org/10.1214/aos/1176345632.

Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.

Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446.

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

Wenjie Wang, Pengfei Tang, Jian Lou, and Li Xiong. Certified robustness to word substitution attack with differential privacy. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1102–1112, Online, June 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.87. URL https://aclanthology.org/2021.naacl-main.87.

Zhaoyang Wang and Hongtao Wang. Defense of word-level adversarial attacks via random substitution encoding. In *Knowledge Science, Engineering and Management: 13th International Conference, KSEM 2020, Hangzhou, China, August 28–30, 2020, Proceedings, Part II*, pp. 312–324, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-55392-0. doi: 10.1007/978-3-030-55393-7_28. URL https://doi.org/10.1007/978-3-030-55393-7_28.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Mao Ye, Chengyue Gong, and Qiang Liu. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3465–3475, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.317. URL https://www.aclweb.org/anthology/2020.acl-main.317.

Jin Yong Yoo and Yanjun Qi. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 945–956, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.81. URL https://aclanthology.org/2021.findings-emnlp.81.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pp. 649–657. Curran Associates, Inc., 2015.

Yi Zhou, Xiaoqing Zheng, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. Defense against synonym substitution-based adversarial attacks via Dirichlet neighborhood ensemble. 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)*, pp. 5482–5492, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.426. URL https://aclanthology.org/2021.acl-long.426.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and JJ (Jingjing) Liu. Freelb: Enhanced adversarial training for natural language understanding. In *Eighth International Conference on Learning Representations (ICLR)*, April 2020. URL https://www.microsoft.com/en-us/research/publication/freelb-enhanced-adversarial-training-for-natural-language-understanding/.

A PROOF

This section provides a proof of Theorem 1. The sketch of the new theorem is as follows:

Consider a simple case where a noise is added to the output of a single intermediate layer and word embeddings of a soft neural network classifier with normalization layers. The network is denoted as $F : \mathbb{R}^d \to \mathcal{P}(\mathcal{Y})$ and word embeddings of an input sequence s are represented as x. Then, F can be deemed as a composite function as follows:

$$F = f_1 \circ f_2 = f_1(f_2(x)),$$

where f_1: $\mathbb{R}^{d'} \to \mathcal{P}(\mathcal{Y})$ and f_2: $\mathbb{R}^d \to \mathbb{R}^{d'}$. After injecting a noise, the new smoothed classifier can be represented as follows:

$$G = g_1 \circ g_2,$$

where g_i is the *Weierstrauss Transform* (Ahmed I, 1996) of f_i as stated in the following definition:

Definition 2. *Denote the original soft neural network classifier as f, the associated smooth classifier (Weierstrauss Transform (Ahmed I, 1996)) can be denoted as g:*

$$g(x) = (f * \mathcal{N}(0, \sigma^2 I))(x) = \mathop{\mathbb{E}}_{\delta \sim \mathcal{N}(0,\sigma^2 I)}[f(x + \delta)]. \tag{5}$$

In the following sections, we will prove L-Lipschtzness of g_1 and g_2. Subsequently, we will show that the output $\arg\max_{y \in \mathcal{Y}} G(x)_y$ does not change within a certain radius of input x (*c.f.,* Theorem 2). Eventually, we will generalize the simplified case towards a general case where multiple layers of activations are perturbed and its radius increases exponentially as we add more noise layers to the classifier (*c.f.,* Theorem 1).

We also provide justifications for the gradient-guide masking strategy (*i.e.,* MI) and show that it acts as a denoising process that enhances the smoothing effect of the proposed approach (Appendix A.6). Note that we adopt Lemma 1, Lemma 2, and Lemma 6 from Li (2019) and follow its proofs.

A.1 LIPSCHITZNESS OF THE SMOOTHED CLASSIFIERS

We will first show that g_1 is $\sqrt{\frac{2}{\pi\sigma^2}}$-Lipschitz in ℓ_2 norm. Note that g_1 is the *Weierstrauss Transform* (Ahmed I, 1996) of a classifier f_1 (*c.f.,* Eq. (5)).

Lemma 1. *Let $\sigma > 0$, let $h : \mathbb{R}^d \to [0, 1]^d$ be measurable, and let $H = h * \mathcal{N}(0, \sigma^2 I)$. Then H is $\sqrt{\frac{2}{\pi\sigma^2}}$-Lipschitz in ℓ_2.*

Proof. In ℓ_2, we have:

$$\begin{aligned}
\nabla H(x) &= \nabla \left(\frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \int_{\mathbb{R}^d} h(t) \exp\left(-\frac{1}{2\sigma^2} \|x - t\|_2^2 \right) dt \right) \\
&= \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \int_{\mathbb{R}^d} h(t) \frac{t - x}{\sigma^2} \exp(-\frac{1}{2\sigma^2} \|x - t\|_2^2) dt \,.
\end{aligned} \tag{6}$$

Let $v \in \mathbb{R}^d$ be a unit vector, the norm of $\nabla H(x)$ is bounded:

$$\begin{aligned}
|\langle v, \nabla H(x) \rangle| &= \left| \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \int_{\mathbb{R}^d} h(t) \left\langle v, \frac{t - x}{\sigma^2} \right\rangle \exp\left(-\frac{1}{2\sigma^2} \|x - t\|_2^2 \right) dt \right| \\
&\leq \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \left| \left\langle v, \frac{t - x}{\sigma^2} \right\rangle \right| \exp\left(-\frac{1}{2\sigma^2} \|x - t\|_2^2 \right) dt \\
&= \frac{1}{\sigma^2} \mathop{\mathbb{E}}_{Z \sim \mathcal{N}(0,\sigma^2)} [|Z|] = \sqrt{\frac{2}{\pi\sigma^2}} \,,
\end{aligned} \tag{7}$$

where the second line holds since $h : \mathbb{R}^d \to [0, 1]^d$. □

By Lemma 1, g_1 is $\sqrt{\frac{2}{\pi\sigma^2}}$-Lipschitz.

Lemma 2. *Let $\sigma > 0$, let $h : \mathbb{R}^d \to [0, 1]$, and let $H = h * \mathcal{N}(0, \sigma^2 I)$. Then the function $\Phi^{-1}(H(x))$ is σ-Lipschitz.*

Proof. Let's first consider a simple case where $\sigma = 1$. Then, we have that

$$\nabla \Phi^{-1}(H(x)) = \frac{\nabla H(x)}{\Phi'(\Phi^{-1}(H(x)))},$$

where Φ^{-1} is the inverse of the standard Gaussian CDF. Then, we need to show the following inequality holds for any unit vector v.

$$\langle v, \nabla H(x) \rangle \leq \Phi'(\Phi^{-1}(H(x))) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\Phi^{-1}(H(x))^2 \right).$$

By Stein's lemma (Stein, 1981), the LHS is equal to

$$\mathbb{E}_{X \sim \mathcal{N}(0,I)}[\langle v, X \rangle \cdot h(x + X)].$$

We need to bound the maximum of this quantity to have the constraint that $h(x) \in [0, 1]$ for all x and $\mathbb{E}_{x \sim \mathcal{N}(0,I)}[h(x + X)] = p$. Let $f(z) = h(z + x)$, the problem becomes:

$$\begin{aligned} &\max \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [\langle v, X \rangle \cdot f(X)] \\ &\text{s.t. } f(x) \in [0, 1] \quad \text{and} \quad \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [f(X)] = p\,. \end{aligned} \tag{8}$$

The solution of the optimization problem is given by the halfspace $\ell(z) = \mathbf{1}[\langle u, z \rangle > -\Phi^{-1}(p)]$ and it is a valid solution to the problem. To show its uniqueness, let f be any other possible solution and A be the support of ℓ. Then, by assumption, $\mathbb{E}_{X \sim \mathcal{N}(0,I)}[\ell(X) - f(X)] = 0$. In particular, we must have:

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [(\ell(X) - f(X))\mathbf{1}_A] = \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [(\ell(X) - f(X))\mathbf{1}_{A^C}]\,.$$

However, for any $z \in A$ and $z' \in A^C$, we have that $\langle v, z \rangle \geq \langle v, z' \rangle$. Hence,

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [\langle v, X \rangle (\ell(X) - f(X))\mathbf{1}_A] \geq \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [\langle v, X \rangle (\ell(X) - f(X))\mathbf{1}_{A^C}],$$

where this uses that $\ell(z) \geq f(z)$ if $z \in A$ and $f(z) \geq \ell(z)$ otherwise. Rearranging, yields

$$\mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [\langle v, X \rangle \ell(X)] \geq \mathop{\mathbb{E}}_{X \sim \mathcal{N}(0,I)} [\langle v, X \rangle f(X)]$$

as claimed. Now, we simply observe that

$$\begin{aligned} \mathop{\mathbb{E}}_{X \sim N(0,I)} [\langle v, X \rangle \ell(X)] &= \mathop{\mathbb{E}}_{Z \sim \mathcal{N}(0,1)} [Z \cdot \mathbf{1}_{Z > -\Phi^{-1}(p)}] \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\Phi^{-1}(p)}^{\infty} x e^{-x^2/2} dx \\ &= \exp\left(-\frac{1}{2}\Phi^{-1}(p)^2 \right) = \exp\left(-\frac{1}{2}\Phi^{-1}(H(x))^2 \right), \end{aligned} \tag{9}$$

as claimed.

To extend the simple case to a general σ, we can take the auxiliary function $\tilde{h}(z) = h(z/\sigma)$, and the corresponding smoothed function $\tilde{H} = \tilde{h} * N(0, 1)$. Then $\tilde{H}(\sigma x) = H(x)$. By the same proof as before, $\Phi^{-1} \circ \tilde{H}$ is 1-Lipschitz, and this immediately implies that $\Phi^{-1} \circ H$ is σ-Lipschitz. □

Therefore, $\Phi^{-1}(g_1(x))$ is a σ-Lipschitz smooth classifier.

A.2 LIPSCHITZNESS OF THE SMOOTHED INTERMEDIATE LAYERS

Lemma 3. *Let $h : \mathbb{R}^d \to \mathcal{N}(0, I^d)$ and $H = h * \mathcal{N}(0, \sigma^2 I^d)$. Then, they are L_h-Lipschitz and L_H-Lipschitz, respectively, where $L_h = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}}$ and $L_H = \frac{1}{1+\sigma^2} L_h$.*

Proof. In general, a Gaussian distribution $\phi \sim \mathcal{N}(\mu, \sigma^2 I)$ is $\frac{1}{\sigma^2\sqrt{2\pi}} e^{-\frac{1}{2}}$-Lipschitz, where $\phi(x)$ can be represented as follows:

$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(- \frac{(x-\mu)^T I (x-\mu)}{2\sigma^2} \bigg).$$

Then, the derivate of ϕ is given by

$$\phi'(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \frac{\mu - x}{\sigma^2} \exp\bigg(- \frac{(x-\mu)^T I (x-\mu)}{2\sigma^2} \bigg).$$

The maximum of $\|\phi'(x)\|$ can be obtained by taking the derivative of its square and set it to zero as follows:

$$\frac{d}{dx}\|\phi'(x)\|^2 = \frac{1}{2\pi\sigma^6} \bigg(- \frac{2\|x-\mu\|^2 (x-\mu)}{\sigma^2} \exp\bigg(- \frac{\|x-\mu\|^2}{\sigma^2} \bigg) + 2(x-\mu) \exp\bigg(- \frac{\|x-\mu\|^2}{\sigma^2} \bigg) \bigg) = 0\,.$$

Note that the square of the norm is a monotonic function as the norm is greater than 0. Thus, we have

$$\|x-\mu\|^2 = \sigma^2\,.$$

This equation implies that the maximum of ϕ' can be found at a distance of σ from μ. For any unit vector $v \in \mathbb{R}^d$, the maximum value of ϕ' occurs at:

$$x = \mu + \sigma v$$

Subsequently, the norm of the maximum gradient is given by:

$$\|\phi'(\mu + \sigma v)\| = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(- \frac{1}{2} \bigg).$$

Since $\|\phi'(x)\| \leq \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2})$, Lipschitz continuity of ϕ can be shown by the Mean Value Theorem:

$$\|\phi(x) - \phi(y)\| \leq \sup_{x \in \mathbb{R}^d} \|\phi'(x)\| \|x-y\| = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\bigg(- \frac{1}{2} \bigg) \|x-y\|\,.$$

Therefore, $\phi \sim \mathcal{N}(\mu, \sigma^2 I)$ is $\frac{1}{\sigma^2\sqrt{2\pi}} e^{-\frac{1}{2}}$-Lipschitz. Since h maps to the standard normal distribution, $h(t)$ is $\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}}$-Lipschitz.

To show the Lipschitz constraint of $H(x)$, we exploit the fact that the convolution of two Gaussian distributions $\phi_1 \sim (\mu_1, \sigma_1^2)$ and $\phi_1 \sim (\mu_2, \sigma_2^2)$ is another Gaussian distribution $\phi = \phi_1 * \phi_2 \sim (\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$, which could be extended to the standard multi-dimensional independent Gaussian variables with no covariance (Bromiley, 2003). This property leads to an equality of $H(x) = h * N(0, \sigma^2 I^d) = N(0, I^d) * N(0, \sigma^2 I^d) \sim N(0, (1+\sigma^2) I^d)$, which shows that $H(x)$ is $\frac{1}{\sqrt{2\pi}(1+\sigma^2)} e^{-\frac{1}{2} I}$-Lipschitz. Thus, $L_H = \frac{1}{1+\sigma^2} L_h$. □

Lemma 3 implies that randomized smoothing imposes a stronger smoothness of the function, since the Lipschitz bound of the original function will be reduced by a factor of $\frac{1}{1+\sigma^2}$ as $1 + \sigma^2 > 1$.

A.3 LIPSCHITZNESS OF THE OVERALL COMPOSITE FUNCTION

Lemma 4. *If f and g are L_1-Lipschitz and L_2-Lipschitz, respectively, then the composite function $f \circ g$ is L_1L_2-Lipschitz.*

Proof.

$$\begin{aligned} |f \circ g(x') - f \circ g(x)| &= |f(g(x')) - f(g(x))| \\ &\leq L_1 |g(x') - g(x)| \\ &\leq L_1 L_2 |x' - x| \,. \end{aligned} \tag{10}$$

□

Lemma 2, Lemma 3, and Lemma 4 lead to the following lemma:

Lemma 5. *$\Phi^{-1}(G) = \Phi^{-1} \circ g_1 \circ g_2$ is $\left(\frac{\sigma_1}{1+\sigma_2^2}\right)$-Lipschitz when*

$$g_1(x) = (f_1 * \mathcal{N}(0, \sigma_1^2 I))(x) = \mathop{\mathbb{E}}_{\delta \sim \mathcal{N}(0, \sigma_1^2 I)} [f_1(x + \delta)]$$

and

$$g_2(x) = (f_2 * \mathcal{N}(0, \sigma_2^2 I))(x) = \mathop{\mathbb{E}}_{\delta \sim \mathcal{N}(0, \sigma_2^2 I)} [f_2(x + \delta)]$$

Proof. If a noise is only added to the input x, then the inverse of the standard Gaussian CDF of the smoothed classifier , *i.e.,* $\Phi^{-1} \circ g(x)$, is σ_1-Lipschitz, as stated in Lemma 2. We can consider g as a composite function , where $g = g_1' \circ g_2'$, then $\Phi^{-1} \circ g_1' \circ g_2'(x)$ is still σ_1-Lipschitz.

Let the Lipschitz constant of Φ^{-1} be L_Φ. Note that $g_1'(x)$ is L_1-Lipschitz since gradients of g_1' is clipped during a training phase. Also g_2' is L_2-Lipschitz as it is a soft classifier. Lemma 4 leads to the following observation:

$$L_\Phi L_1 L_2 = \sigma_1$$

Subsequently, we consider a case where the intermediate outputs of the composite function are perturbed. In other words, we also apply Weierstrauss transform to the layer f_1 (*i.e.,*f_1 to g_1). Thus, g_1 is $\left(\frac{1}{1+\sigma_2^2} L_1\right)$-Lipschitz by Lemma 3. Then, Lemma 3 with Lemma 4 results in the new Lipschitzness constant for $\Phi^{-1}(G)$, which is given by

$$L_\Phi \frac{1}{1+\sigma_2^2} L_1 L_2 = \frac{\sigma_1}{1+\sigma_2^2} \,.$$

□

A.4 ROBUST RADIUS OF INPUT

Lemma 6. *Let $m : \mathbb{R} \to \mathbb{R}$ be a monotone, invertible function. Suppose that $F : \mathbb{R}^d \to \mathcal{P}(\mathcal{Y})$ is a soft classifier, and moreover, the function $x \mapsto m(F(x)_y)$ is L-Lipschitz in norm $||\cdot||$, for every $y \in Y$. Let a and b are the most likely classes which are denoted as $a = \arg\max_{y \in Y} G(x)_y$ and $b = \arg\max_{y \in \mathcal{Y} \setminus a} G(x)_y$, respectively, and their corresponding probabilities are p_a and p_b, then, we have that $\arg\max_{y \in \mathcal{Y}} F(x') = a$ for all x' so that $||x' - x|| < \frac{1}{2L}(m(p_a) - m(p_b))$.*

Proof. As $x \mapsto m(F(x)_y)$ is L-Lipschitz, we know that for any x' within ball $\frac{1}{2L}(m(p_a) - m(p_b))$, we have:

$$|m(F(x')_a) - m(F(x)_a)| = |m(F(x')_a) - m(p_a)| \leq L\|x' - x\| < \frac{1}{2}(m(p_a) - m(p_b))$$

In particular, this implies that $m(F(x')_a) > \frac{1}{2}(m(p_a) + m(p_b))$. However, for any $y \neq a$, by the same logic,

$$m(F(x')_y) < m(F(x)_y) + \frac{1}{2}(m(p_a) - m(p_b)) \leq \frac{1}{2}(m(p_a) + m(p_b)) < m(F(x')_a)$$

Hence, $\arg\max_{y \in \mathcal{Y}} F(x') = a$. □

Theorem 2. *Let $F : \mathbb{R}^d \to \mathcal{P}(\mathcal{Y})$ be any soft classifier, let $\sigma > 0$, and let G be its associated soft classifier, Let*

$$a = \arg\max_{y \in \mathcal{Y}} G(x)_y \quad \textit{and} \quad b = \arg\max_{y \in \mathcal{Y} \setminus a} G(x)_y$$

be two most likely classes for x according to G. Then, we have that $\arg\max_{y \in \mathcal{Y}} G(x')_y = a$ for x' satisfying

$$\|x' - x\|_2 \leq \frac{1 + \sigma_2^2}{2\sigma_1}(\Phi^{-1}(p_a) - \Phi^{-1}(p_b))$$

Proof. Follows from Lemma 5 and Lemma 6. □

Theorem 2 implies that a prediction of the smoothed network is robust around the input x within a radius of $\frac{1+\sigma_2^2}{2\sigma_1}(\Phi^{-1}(p_a) - \Phi^{-1}(p_b))$. The robust radius of the proposed randomized smoothing approach is scaled by a factor of $(1 + \sigma_2^2) > 1$.

A.5 Generalization to multi-layer

Theorem 2 can be generalized to a multi-layer perturbation approach for a multi-layer network $F = f_1 \circ f_2 \circ \cdots \circ f_L$ by repetitively applying Lemma 3. Then, the new smoothed classifier G is $\sigma_1 / \prod_{l=2}^{L}(1 + \sigma_l^2)$-Lipschitz and it yields Theorem 1 with Lemma 6 by iterative use of Lemma 3 with Lemma 6.

A.6 De-noising effect of the gradient-guided masked inference

The de-noising effect of the gradient-guided masked inference (MI) can be understood via its impact on the Lipschtiz continuity of a soft classifier $F : \mathbb{R}^d \to \mathcal{P}(\mathcal{Y})$. Let's consider a case where we have an input sequence s and its perturbed sequence s'. Then, their distributed representations are x and x', respecitvely. Subsequently, let L be the maximum gradient for masking, then a classifier F is L-Lipschitz, as the first derivatives is bounded by the max gradient L. The Lipschitz property is given by

$$|F(x') - F(x)| \leq L|x' - x|\,.$$

The proposed gradient-guided masking process lowers the Lipschtiz constant of F through masking a token under a guidance of max gradient signals. It implies that F will become L'-Lipschitz, where $L' < L$ and it can be represented as follows:

$$|F(\hat{x}) - F(x)| \leq L'|\hat{x} - x| < L|x' - x|\,,$$

where $\hat{x}$ is the max gradient-guided masked word embeddings. As shown in the above equation, MI can effectively lower the upper bound of the prediction change and increases a chance of pushing the model prediction to fall into the robustness radius derived by the randomized smoothing method.

B Experiment details

Experiment environment All of the experiments are conducted on an Intel Xeon Gold 5218R CPU-2.10GHz processor with a single Quadro RTX 6000 GPU under Python with PyTorch (Paszke et al., 2019).

Models used The models used in this work are pre-trained RoBERTa-base (Liu et al., 2019) and BERT-base (Devlin et al., 2019), both of which have 124 milion parameters. We adopt Huggingface library (Wolf et al., 2020) for training the models on the benchmark datasets.

Parameter settings of RSMI RSMI and the fine-tuned models are optimized by AdamW (Loshchilov & Hutter, 2019)with a linear adaptive learning rate scheduler. The maximum sequence length of input sequences is set to 256 during experiments. Further details are summarized in Table 8.

Table 8: Parameter settings of RSMI and the fine-tuned models. AL denotes the adaptive linear learning rate scheduler.

Model		IMDb		AGNews		QNLI	
		RSMI	Fine-Tuned	RSMI	Fine-Tuned	RSMI	Fine-Tuned
RoBERTa-base	Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
	Batch size	16	16	24	24	36	36
	Epochs	10	10	10	10	10	10
	Learning rate	10^-5	5×10^-5	10^-5	5×10^-5	10^-5	10^-5
	Learning rate scheduler	AL	AL	AL	AL	AL	AL
	Maximum sequence length	256	256	256	256	256	256
	M	4	-	4	-	2	-
	σ	0.4	-	0.4	-	0.2	-
	# Noise layers	3	-	3	-	3	-
	ν	1	-	1	-	1	-
	k_0	5	-	5	-	5	-
	k_1	50	-	50	-	50	-
	α	0.98	-	0.98	-	0.98	-
	β	1	-	1	-	1	-
BERT-base	Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
	Batch size	16	16	24	24	36	36
	Epochs	10	10	10	10	10	10
	Learning rate	10^-5	5×10^-5	10^-5	5×10^-5	10^-5	10^-5
	Learning rate scheduler	AL	AL	AL	AL	AL	AL
	Maximum sequence length	256	256	256	256	256	256
	M	3	-	2	-	2	-
	σ	0.3	-	0.2	-	0.2	-
	# Noise layers	4	-	3	-	3	-
	ν	1	-	1	-	1	-
	k_0	5	-	5	-	5	-
	k_1	50	-	50	-	50	-
	α	0.98	-	0.98	-	0.98	-
	β	1	-	1	-	1	-

Textual attack algorithm We employed the publicly available TextAttack library (Morris et al., 2020) for TextFooler (TF) (Jin et al., 2020), PWWS (Ren et al., 2019), and BAE (Garg & Ramakrishnan, 2020) attack algorithms. We follow the default settings of each algorithm. Note that TextAttack does not include the named entity (NE) adversarial swap constraint in its PWWS implementation to extend PWWS towards a practical scenario where NE labels of input sequences are not available. As a consequence, PWWS attack in TextAttack tends to show stronger attack success rates.

C Analysis of hidden representations of perturbed samples

We investigate the change in latent representation of input samples upon perturbation with respect to the original latent representations on using RSMI. We compare the L_2 distance and cosine similarity between a hidden representation of clean sample $h_l(s)$ and that of its corresponding adversarial example $h_l(s')$ for each layer l of the fine-tuned base models with that of RSMI. Fig. 1 (Right) shows that the changes of L_2 distance and cosine similarity of the fine-tuned RoBERTa and BERT stay quite constant until later 8. However, for subsequent layers, we observe a rapid increase in L_2 distance and a sudden fall in cosine similarity for these models. At the final layer ($l = 12$), we see the largest differences. In contrast, RSMI tends show abrupt changes at l_1, l_5, and l_9 with small standard deviations, probably due to the Gaussian noise perturbation processes for these layers. However, at the final layers, especially at $l = 12$ that is used for prediction, we observe very minor L_2 distance and high cosine similarity. This further validates RSMI's ability to more effectively represent the perturbed samples in the latent representation space in accordance with the original samples, eventually leading to higher adversarial robustness.

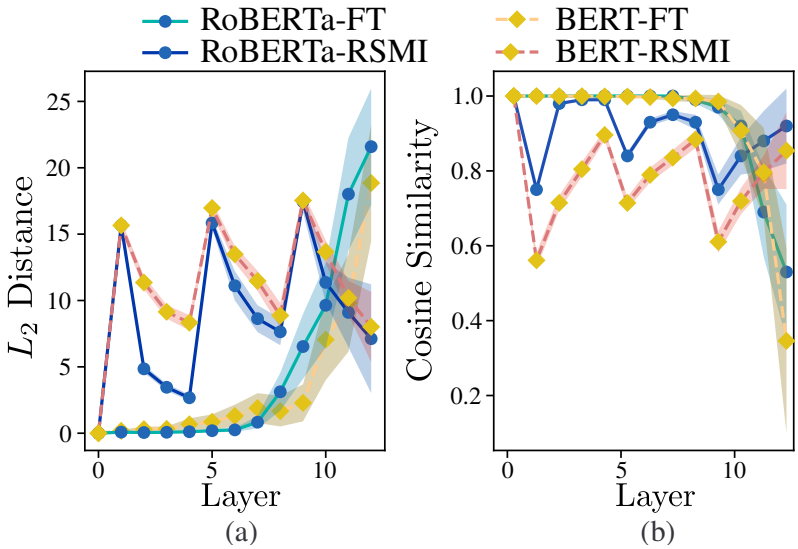

Figure 1: Analysis of hidden representations of RSMI.

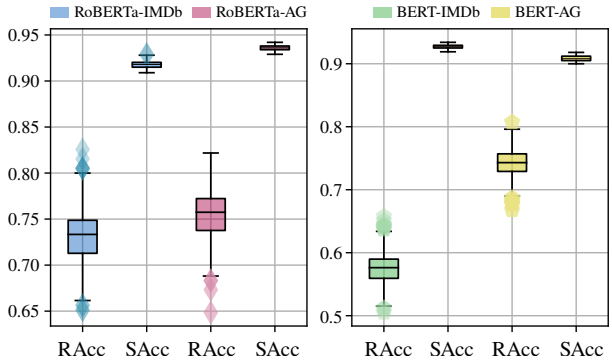

Figure 2: Stochastic stability of RSMI.

D STOCHASTIC STABILITY OF RSMI

As RSMI is stochastic in nature, we examine its stability in classifying clean as well as adversarial samples. We first randomly draw 1,000 clean examples and evaluate them with RSMI with 1,000 independent runs. Next, we perform inference repeatedly using adversarial examples that were successful in fooling RSMI with 1,000 independent runs. As all the evaluations are independent, each evaluation involves a different noise sampling and masking process. Fig. 2 shows that RSMI's evaluation on clean samples is significantly stable. On the other hand, most of the adversarial examples are correctly classified during each individual evaluation around the median of RSMI's RAcc. This shows that the attack success rate of adversarial attack algorithms become stochastic rather than deterministic due to RSMI's non-deterministic nature.

E EFFECT OF EARLY STOPPING DURING FINE-TUNING

We make some interesting observations while investigating the effect of early stopping during fine-tuning of models on their adversarial robustness. From Fig. 3, we observe that RoBERTa, without any fine-tuning on the datasets ("0" Iteration), inherently possesses robustness to adversarial attacks. We further observe that models fine-tuned for less number of steps possess stronger robustness to adversarial attacks. As we fine-tune them further, their standard accuracies rapidly improve at the loss of adversarial robustness. A potential explanation of this result may be related to the robustness

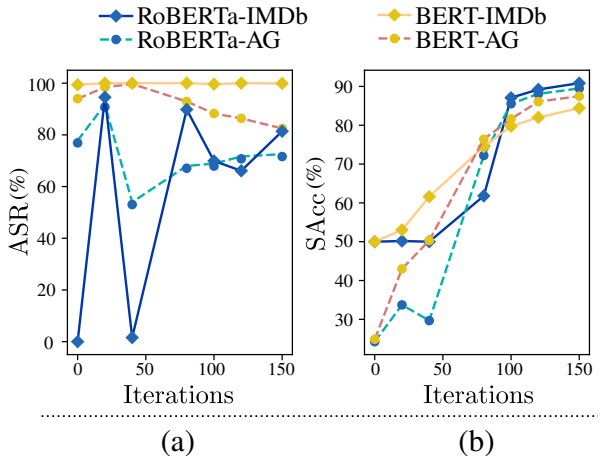

Figure 3: (a) and (b) respectively show changes in attack success rate (ASR) and standard accuracy (SAcc) of fine-tuned models over tuning iterations.

improvements from self-supervised training approaches (Hendrycks et al., 2019). However, their robustness is compromised by the fine-tuning process that only aims to optimize the training for a higher standard accuracy (Ilyas et al., 2019).

F VARIATION IN EXPECTATION OVER INPUT TRANSFORMATIONS

As SAFER and RSMI transform inputs in their prediction phase, we study the impact of their input transformation processes on their adversarial robustness through expectation over transformations (EOT) (Athalye et al., 2018b;a). Specifically, we take an average of models' predictions over multiple transformed inputs corresponding to a given input sample while varying the number of transformed samples, k. We also report RAcc of RSMI without conducting the second sampling step (*c.f.,* line 11-19 in Algorithm 1, denoted by w/o TS in Fig. 4) for a fair comparison with SAFER and understand the impact of the two-step sampling. Fig. 4 shows that RAcc of SAFER peaks at $k = 1$ and steadily decreases as k increases. We claim that this tendency might be attributed to an *attack obfuscation* issue that attempts to defend an NLP system via disturbing a perturbation process of an attack algorithm by an input transformation such as a random word-substitution. However, attack algorithms can deviate the obfuscation by approximating the optimal perturbation through a larger number of queries for estimating an expectation of perturbation directions over transformed samples. In contrast to SAFER, RAcc of RSMI remains stable or even increases steadily for both BERT and RoBERTa. Also, the two-step sampling significantly enhances the RAcc. We credit this robustness to the transform proceess of RSMI, which transforms an input text into a masked input and perturbs their hidden representation with Gaussian noise. This behaves as a mixture of two stochastic transformations and makes it harder for the attack algorithms to estimate the optimal perturbation direction.

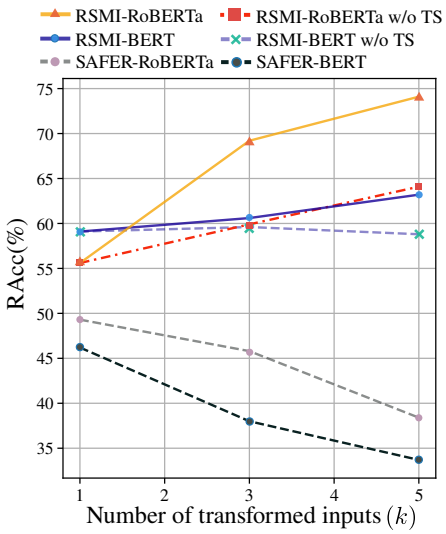

Figure 4: Variation of robust accuracy (RAcc) in expectation over input transformations.