Robustness-Aware Word Embedding Improves Certified Robustness to Adversarial Word Substitutions

Yibin Wang[1*], **Yichen Yang**[1*], **Di He**[2] and **Kun He**[1†]

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

{yibinwang, yangyc, brooklet60}@hust.edu.cn

[2] School of Intelligence Science and Technology, Peking University, Beijing, China

dihe@pku.edu.cn

Abstract

Natural Language Processing (NLP) models have gained great success on clean texts, but they are known to be vulnerable to adversarial examples typically crafted by synonym substitutions. In this paper, we target to solve this problem and find that word embedding is important to the certified robustness of NLP models. Given the findings, we propose the Embedding Interval Bound Constraint (EIBC) triplet loss to train robustness-aware word embeddings for better certified robustness. We optimize the EIBC triplet loss to reduce distances between synonyms in the embedding space, which is theoretically proven to make the verification boundary tighter. Meanwhile, we enlarge distances among non-synonyms, maintaining the semantic representation of word embeddings. Our method is conceptually simple and componentized. It can be easily combined with IBP training and improves the certified robust accuracy from 76.73% to 84.78% on the IMDB dataset. Experiments demonstrate that our method outperforms various state-of-the-art certified defense baselines and generalizes well to unseen substitutions. The code is available at https://github.com/JHL-HUST/EIBC-IBP/.

1 Introduction

Deep neural networks have achieved impressive performance on many NLP tasks (Devlin et al., 2019; Kim, 2014). However, they are known to be brittle to adversarial examples: the model performance could dramatically drop when applying imperceptible crafted perturbations, especially synonym substitutions, into the input text. These phenomena have been observed in a wide range of practical applications (Alzantot et al., 2018; Ren et al., 2019; Wallace et al., 2019; Zang et al., 2020; Maheshwary et al., 2021; Meng and Wattenhofer, 2020; Yu et al., 2022).

To mitigate the vulnerability of NLP models, many adversarial defense methods have been proposed to boost the model robustness from various perspectives, such as adversarial training (Wang et al., 2021b; Dong et al., 2021; Li et al., 2021; Si et al., 2021), advanced training strategy (Liu et al., 2022), input transformation (Wang et al., 2021a), and robust word embedding (Yang et al., 2022). However, these methods could only provide empirical robustness, *i.e.*, the robust accuracy of these models varies depending on the heuristic search used in the attacks. In contrast, certified robustness guarantees that a model is robust to all adversarial perturbations of a given input, regardless of the attacks for evaluation. Certified robustness provides a lower bound on the robust accuracy of a model in the face of various adversarial attacks.

In this work, we aim to design better training methods for certified robustness. In particular, our algorithm is mainly based on Interval Bound Propagation (IBP). IBP is initially designed for images (Gowal et al., 2019) and is also utilized to provide certified robustness in NLP models (Huang et al., 2019; Jia et al., 2019). In the first step, we compute the interval of embedding of all possible texts perturbed on the current input by word substitutions, where the embedding layer is fixed using the commonly used word embeddings, such as GloVe (Pennington et al., 2014). Then, in the second step, given the pre-computed interval, IBP is used to estimate the upper and lower bounds of the output layer by layer and minimize the worst-case performance to achieve certified robustness.

However, previous works of IBP method (Huang et al., 2019; Jia et al., 2019) use *fixed* word embeddings and we argue that may not be good enough for certified robustness. As shown in the experiments of Huang et al. (2019), the embedding space significantly impacts the IBP bounds and the effectiveness of IBP training. Though the close neighbor words in the embedding space are selected for

[*] The first two authors contribute equally.

[†] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 673–687

the synonym set, the volume of the convex hull constructed by them is still large for IBP training, which will lead to loose bounds through propagating and a poor robustness guarantee. Inspired by the above observation, in this work, we develop a new loss to train robustness-aware word embeddings for higher certified robustness.

We first decompose certified robust accuracy into robustness and standard accuracy. We optimize for robustness from the perspective of embedding constraint and optimize for standard accuracy by training the model normally. It can be proved that the upper bound of certified robustness can be optimized by reducing the interval of the convex hull constructed by synonyms in the embedding space. Therefore, we propose a new loss called Embedding Interval Bound Constraint (EIBC) triplet loss. Specifically, given a word, on each dimension in the embedding space, we aim to reduce the maximum distance between each word and its synonyms, which is actually to make a smaller interval of the convex hull formed by synonyms. Then, we freeze the embedding layer after training the word embeddings by EIBC triplet loss, and train the model by normal training or IBP training to achieve higher certified robust accuracy.

Extensive experiments on several benchmark datasets demonstrate that EIBC could boost the certified robust accuracy of models. Especially when EIBC is combined with IBP training, we could achieve SOTA performance among advanced certified defense methods. For instance, on IMDB dataset, EIBC combined with IBP training achieves 84.78% certified robust accuracy, surpassing IBP by about 8%, which indicates that constraining the embedding interval bound will significantly boost the performance of IBP. Our main contributions are as follows.

- We prove theoretically that the upper bound of certified robustness can be optimized through reducing the interval of the convex hull formed by synonyms in the embedding space.
- We propose a new loss of EIBC constraining the word embeddings. EIBC is plug-and-play and could combine with normal training or IBP training to boost certified robust accuracy.
- Extensive experiments demonstrate that EIBC combined IBP training significantly promotes the certified robustness of the model across multiple datasets. EIBC also exhibits good generalization to unseen word substitutions.

2 Related Work

There are many adversarial defense methods to boost the model's robustness to adversarial word substitutions. Adversarial Training (AT), one of the most popular defense approaches, crafts adversarial examples during the training and injects them into the training set (Alzantot et al., 2018; Ren et al., 2019; Ivgi and Berant, 2021; Si et al., 2021). A stream of work aims to improve the effectiveness and efficiency of textual adversarial training by adversary generation based on gradient optimization (Wang et al., 2021b; Dong et al., 2021; Li et al., 2021). To eliminate the differences between clean samples and adversarial examples, Wang et al. (2021a) insert a synonym encoder before the input layer, and Yang et al. (2022) propose Fast Triplet Metric Learning (FTML) to train robust word embeddings. Liu et al. (2022) leverage the Flooding training method (Ishida et al., 2020) to guide the model into a smooth parameter landscape that leads to better adversarial robustness. Besides, adversarial detection methods detect the adversarial examples before feeding the input samples to models by training a classifier (Zhou et al., 2019) or randomized substitution (Wang et al., 2022). However, these methods can only provide empirical robustness, which is unstable for attacks based on different heuristic searches.

Certified robustness is proposed to guarantee that a model is robust to all adversarial perturbations of any given input. Interval Bound Propagation (IBP) calculates the input interval involving all possible word substitutions and propagates the upper and lower bounds through the network, then minimizes the worst-case loss that any combination of the word substitutions may cause (Jia et al., 2019; Huang et al., 2019). Randomized smoothing methods, such as SAFER (Ye et al., 2020) and RanMASK (Zeng et al., 2021), mask a random portion of the words in the input text to construct an ensemble and utilize the statistical properties of the ensemble to predict the output. Zhao et al. (2022) propose Causal Intervention by Semantic Smoothing (CISS), which associates causal intervention with randomized smoothing in latent semantic space to make provably robust predictions.

Most previous works do not attach importance to word embeddings concerning certified robustness. Our work introduces EIBC triplet loss to

achieve certified robustness through constraining word embeddings and incorporates it into IBP to boost certified robustness.

In the field of adversarial images, Shi et al. (2021) improve the IBP training method by mitigating the issues of exploded bounds at initialization and the imbalance in ReLU activation states. It is worth noting that our work differs from Shi et al. (2021). We particularly focus on reducing the difference between the upper and lower bounds of initial inputs by fine-tuning the embeddings. The reduction of bounds interval provably causes the tightening of bounds in following propagation.

3 Preliminaries

For the text classification task, a model $f : \mathcal{X} \to \mathcal{Y}$ predicts label $y \in \mathcal{Y}$ given a textual input $\mathbf{x} \in \mathcal{X}$, where $\mathbf{x} = \langle x_1, x_2, \cdots, x_N \rangle$ is a sequence consisting of N words, and the output space $\mathcal{Y} = \{y_1, y_2, \cdots, y_C\}$ contains C classes. In this paper, we focus on an adversarial scenario in which any word in the textual input can be arbitrarily replaced by its synonyms so as to change the model's prediction. Formally, we use $\mathcal{S}(x_i)$ to denote the synonym set of the i^{th} word x_i of input $\mathbf{x}$. Then, we formulate the set consisting of all the adversarial examples with allowed perturbations of $\mathbf{x}$:

$$\mathcal{B}_{adv}(\mathbf{x}) = \{\langle x'_1, x'_2, \cdots, x'_N \rangle, x'_i \in \mathcal{S}(x_i) \cup \{x_i\}\}. \tag{1}$$

Our goal is to defend against the adversarial word substitutions and train models with certified robustness, *i.e.*,

$$\forall\, \mathbf{x}' \in \mathcal{B}_{adv}(\mathbf{x}), \quad f(\mathbf{x}') = f(\mathbf{x}) = y. \tag{2}$$

If Eq. (2) holds and the model classifies the instance correctly, that is, $y = y_{true}$, then we call the model prediction on input $\mathbf{x}$ is certified.

We can easily decompose certified robust accuracy into *robustness* and *standard accuracy*. Robustness cares about whether the model prediction is consistent under perturbations. Clearly, achieving robustness is a necessary condition for obtaining models with high certified robust accuracy. We then illustrate the conditions to be satisfied for robustness in terms of interval bound. For a K-layer neural network, assuming we can calculate the interval bound of the output logits $\mathbf{z}^K$: $\underline{\mathbf{z}}^K \le \mathbf{z}^K \le \overline{\mathbf{z}}^K$ of all the perturbed inputs $\mathbf{x}' \in \mathcal{B}_{adv}(\mathbf{x})$, the model with robustness satisfies that the lower bound of the model's largest logit $\mathbf{z}^K_{y_{max}}$ is greater than the upper bound of other logits, *i.e.*,

$$\underline{\mathbf{z}}^K_{y_{max}} \ge \overline{\mathbf{z}}^K_y, \quad \forall y \in \mathcal{Y}, y \ne y_{max}. \tag{3}$$

To evaluate the model's certified robust accuracy, we just need to replace the model's largest logit $\mathbf{z}^K_{y_{max}}$ with the logit of the true class $\mathbf{z}^K_{y_{true}}$ in Eq. (3).

Interval Bound Propagation IBP provides the solution to estimate the interval bound layer by layer. We could represent a K-layer neural network model as a series of transformations f_k (*e.g.*, linear transformation, ReLU activation function):

$$\mathbf{z}^k = f_k(\mathbf{z}^{k-1}), \quad k = 1, \cdots, K, \tag{4}$$

where $\mathbf{z}^k$ is the vector of activations in the k^{th} layer. To calculate the interval bound of the output logits, we need to construct the interval bound of the input vector and propagate it through the network. Let $\varphi(x_i) \in \mathbb{R}^D$ denote the embedding word vector of word x_i with D dimensions. The word vector input is $\mathbf{z}^0 = \langle \varphi(x_0), \varphi(x_1), \cdots, \varphi(x_N) \rangle$. We obtain the interval bounds of the word vector input $\mathbf{z}^0$ by constructing the convex hull of $\mathcal{S}(x_i)$ in the embedding space:

$$\begin{aligned} \underline{z}^0_{ij} &= \min_{x_i \in \mathcal{S}(x_i) \cup \{x_i\}} \varphi(x_i)_j, \\ \overline{z}^0_{ij} &= \max_{x_i \in \mathcal{S}(x_i) \cup \{x_i\}} \varphi(x_i)_j, \end{aligned} \tag{5}$$

where $\varphi(x_i)_j$ is the j^{th} element of the word vector of word x_i. $\underline{\mathbf{z}}^0$ and $\overline{\mathbf{z}}^0$ are the lower and upper bounds of $\mathbf{z}^0$, respectively.

Similarly, for subsequent layers $k > 0$, we denote the lower and upper bounds of activations in the k^{th} layer as $\underline{\mathbf{z}}^k$ and $\overline{\mathbf{z}}^k$, respectively. The bounds on the $\mathbf{z}^k$ can be obtained from the bounds of previous layer $\mathbf{z}^{k-1}$:

$$\begin{aligned} \underline{z}^k_i &= \min_{\underline{\mathbf{z}}^{k-1} \le \mathbf{z}^{k-1} \le \overline{\mathbf{z}}^{k-1}} \mathbf{e}_i^\top f_k(\mathbf{z}^{k-1}), \\ \overline{z}^k_i &= \max_{\underline{\mathbf{z}}^{k-1} \le \mathbf{z}^{k-1} \le \overline{\mathbf{z}}^{k-1}} \mathbf{e}_i^\top f_k(\mathbf{z}^{k-1}), \end{aligned} \tag{6}$$

where $\mathbf{e}_i$ is the one-hot vector with 1 in the i^{th} position. Interval Bound Propagation (IBP) (Gowal et al., 2018) gives a simple way to solve the above problems for affine layers and monotonic activation functions as described in Appendix B.

4 Methodology

In this section, we first theoretically demonstrate the influence of word embedding on the model robustness and then introduce the proposed EIBC triplet loss to optimize the word embedding. Finally, we describe how to incorporate EIBC into the training process.

4.1 Word Embedding Matters Robustness

Previous works on the IBP method (Huang et al., 2019; Jia et al., 2019) use fixed word embeddings. As illustrated in Figure 1, IBP constructs an axis-aligned box around the convex hull constructed by synonyms in the embedding space. As stated in Huang et al. (2019), since synonyms may be far away from each other, the interval of the axis-aligned box can be large. Through propagating the interval bounds in the network, the interval bounds become too loose to satisfy the certified conditions.

To be concrete, based on Eq. (3), training a model with certified robustness is an optimization problem formulated as follows:

$$\text{minimize } \overline{\mathbf{z}}_y^K - \underline{\mathbf{z}}_{y_{max}}^K, \quad \forall y \in \mathcal{Y}, y \neq y_{max}. \tag{7}$$

We propose the following theorem to demonstrate that minimizing the objective in Eq. (7) could be converted to an optimization objective with respect to the word embeddings by backpropagating the interval bounds through the network. We provide the proof in Appendix A.

Theorem 1 *The upper bound on the solution of Eq. (7) is*

$$\textit{minimize} \quad \max_{x_i \in \mathbf{x}} \Big(\max_{x_i' \in \mathcal{S}(x_i)} \big(\big| \varphi(x_i) - \varphi(x_i') \big| \big) \Big). \tag{8}$$

where $\max(\cdot)$ *and* $|\cdot|$ *are the element-wise operators.*

Theorem 1 inspires us that we could approach certified robustness by reducing the interval of the convex hull constructed by synonyms in the embedding space.

4.2 Robustness-Aware Word Embedding

Based on Theorem 1, we attach importance to word embeddings and propose the Embedding Interval Bound Constraint (EIBC) triplet loss to train robustness-aware word embeddings to achieve higher certified robustness while maintaining their representation capability for classification.

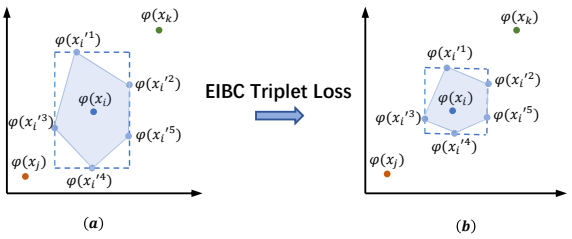


Figure 1: EIBC triplet loss reduces the area of the axis-aligned box formed by synonyms and meanwhile holds the distance between the word and its non-synonyms in the embedding space. x_i is a word with $x_i'^1, x_i'^2, ..., x_i'^5 \in \mathcal{S}(x_i)$ as its synonyms and x_j, x_k as its non-synonyms. The dashed line represents the bound of the convex hull constructed by synonyms.

We measure the interval of the convex hull constructed by synonyms of word x_i in the embedding space by:

$$d_{bound}(x_i, \mathcal{S}(x_i)) = \| \max_{x_i' \in \mathcal{S}(x_i)} \big| \varphi(x_i) - \varphi(x_i') \big| \|_p, \tag{9}$$

where $\| \cdot \|_p$ indicates p-norm. According to Theorem 1, the certified robustness can be optimized by minimizing $d_{bound}(x_i, \mathcal{S}(x_i))$ for each word x_i in the input squence $\mathbf{x}$.

Meanwhile, non-synonyms may be connected by multiple synonym pairs, and simply reducing the distance between synonyms will also reduce the distance between non-synonyms. To prevent all words from being drawn close to each other and hurting semantic representation, we also control the distances between words and their non-synonyms. Inspired by FTML (Yang et al., 2022), we adopt the triplet metric learning to reduce the interval of convex hull constructed by synonyms and increase the distance between words and their non-synonyms simultaneously. Consistent with Eq. (9), we also use the p-norm distance of word vectors in the embedding space as the distance metric between two words x_a and x_b:

$$d(x_a, x_b) = \|\varphi(x_a) - \varphi(x_b)\|_p. \tag{10}$$

In this work, we adopt the Manhattan distance, *i.e.*, $p = 1$ and provide analysis on different p-norms in Section 5.7.

Finally, we design the EBIC triplet loss for each

word x_i as follows:

$$\mathcal{L}_{EIBC}(x_i, \mathcal{S}(x_i), \mathcal{N}(M)) = d_{bound}(x_i, \mathcal{S}(x_i)) - \frac{1}{M} \sum_{\tilde{x_i} \in \mathcal{N}(M)} min(d(x_i, \tilde{x_i}), \alpha) + \alpha, \tag{11}$$

where $\mathcal{S}(x_i)$ denotes the synonym set of word x_i, and $\mathcal{N}(M)$ denotes the set containing M words randomly sampled from the vocabulary. We set M to be the same as the maximum size of the synonym set of a word to maintain the duality of the maximization and minimization problem. Note that the purpose of increasing the distance between words and their non-synonyms is to prevent them from getting too close and losing semantic representations, without constantly increasing their distance. Thus we set a scalar hyperparameter α to control that they would no longer be pushed away once the distance exceeds α.

We minimize $\mathcal{L}_{EIBC}(x_i, \mathcal{S}(x_i), \mathcal{N}(M))$ to reduce the interval of convex hull shaped by word x_i and its synonyms (positive samples) and maintain the distances between x_i and its non-synonyms (negative samples) in the embedding space.

Figure 1 illustrates the effect of EIBC triplet loss. In the embedding space, the interval of the convex hull constructed by synonyms of word x_i is reduced, while distances between x_i and its non-synonyms x_j, x_k are maintained.

4.3 Overall Training Process

As described in Section 3, we decompose certified robust accuracy into two parts: certified robustness and standard accuracy. We utilize the proposed EIBC triplet loss to achieve certified robustness from the perspective of word embeddings, and optimize for standard accuracy by training the model normally.

In the first part, we use EIBC triplet loss to fine-tune the pretrained word embeddings, *e.g.*, GloVe word embeddings (Pennington et al., 2014) to get robust word embeddings. To employ the $\mathcal{L}_{EIBC}$ to each word of input $\mathbf{x}$ in the embedding space, we sum up $\mathcal{L}_{EIBC}$ of each word and take the mean value as our final loss $\mathcal{L}_{emb}$ to train the word embeddings:

$$\mathcal{L}_{emb} = \frac{1}{|\mathbf{x}|} \sum_{x_i \in \mathbf{x}} \mathcal{L}_{EIBC}(x_i, \mathcal{S}(x_i), \mathcal{N}(M)). \tag{12}$$

In the second part, since our BIEC method merely provides the word embedding with certified robustness, which is componentized, we could combine it with various training methods to boost the certified robust accuracy. Specifically, we freeze the embedding layer trained by EIBC triplet loss and train the model with normal cross-entropy loss or with IBP training method (Jia et al., 2019) towards higher certified robust accuracy.

The loss of IBP training is as follows:

$$\mathcal{L}_{model} = (1 - \beta) \cdot \mathcal{L}_{CE} + \beta \cdot \mathcal{L}_{IBP}(\epsilon), \tag{13}$$

where $\mathcal{L}_{CE}$ denotes the normal cross-entropy loss and $\mathcal{L}_{IBP}$ denotes the IBP loss (we give a brief description of the IBP loss in Appendix B). Scalar hyperparameter β governs the relative weight between the robustness and standard accuracy. The IBP loss uses ϵ to control the perturbation space size, and $\epsilon = 1$ means the original size. To maintain the balance between robustness and standard accuracy during training, the IBP training method gradually increases β and ϵ from 0 to 1. With the help of EIBC, we could reduce the training epochs to half of the original IBP training method.

5 Experiments

This section evaluates the proposed method with three advanced certified defense methods on three benchmark datasets. In addition, we further study EIBC on the generalization to unseen word substitutions, the empirical robustness, the trade-off between clean and robust accuracy, the training procedure, and the robustness with different distance metrics.

5.1 Experimental Setup

Tasks and Datasets We focus on evaluating certified robustness against adversarial word substitutions. Aligned with previous works (Jia et al., 2019; Ye et al., 2020; Zhao et al., 2022), we evaluate the proposed method on three benchmark datasets for the text classification task, including IMDB (Maas et al., 2011), YELP (Shen et al., 2017), and SST-2 (Wang et al., 2019).

Baselines We compare our proposed method with IBP (Jia et al., 2019), SAFER (Ye et al., 2020) and CISS (Zhao et al., 2022). We use the models with the best results for baselines. We also make our own implementation of IBP method on the TextCNN model (Kim, 2014). In our implementation of IBP, we tune and choose the best training

Method	Model	IMDB	YELP	SST-2
IBP Training (Jia et al., 2019)	CNN	67.83	85.94	66.17
IBP Training*	TextCNN	76.73	88.72	69.15
SAFER (Ye et al., 2020)†	BERT	69.20	80.63	-
CISS (Zhao et al., 2022)†	BERT	75.25	90.47	-
EIBC+Normal Training	TextCNN	72.37	89.51	66.86
EIBC+IBP Training*	TextCNN	**84.78**	**93.66**	**76.95**

* Our implementation.
† Results are obtained from Zhao et al., 2022

Table 1: The certified robust accuracy (%) against word substitutions on the IMDB, YELP and SST-2 datasets. All models are trained and evaluated using the word substitutions from Jia et al. (2019) as the perturbations for a fair comparison. Ye et al. (2020) and Zhao et al. (2022) do not report their results on the SST-2 dataset.

schedule and hyperparameters depending on certified robust accuracy, and the performance is better than that reported in Jia et al. (2019).

Perturbation Setting Following previous work, we use the same synonym substitutions as in Jia et al. (2019) and Zhao et al. (2022), which are initially defined in Alzantot et al. (2018). The synonyms of each word are defined as the $n = 8$ nearest neighbors satisfying the cosine similarity ≥ 0.8 in the GloVe embedding space (Pennington et al., 2014) processed by counter-fitting (Mrksic et al., 2016).

Model Setting Jia et al. (2019) adopt a simple CNN model with the filter size of 3 and 100 as the hidden size, termed CNN in the experiments. We adopt a TextCNN model (Kim, 2014) with three filter sizes (2, 3, 4) and 200 as the hidden size, termed TextCNN. Following Jia et al. (2019), we set a linear layer before the CNN layers of the models to further control the shape of the convex hull constructed by synonyms. We study the impact of different architectures in Appendix C.3.

Implementation Details We use the default train/test split for IMDB and YELP datasets. For SST-2, we use the default training set and take the development set as the testing set. For the generalization of EIBC, we set the hyperparameter $\alpha = 10.0$ in Eq. (11) for all experiments. Analyses of the impact of α are discussed in Section 5.5.

For the EIBC+Normal training method, we first use our EIBC triplet loss to train the word embeddings for 20 epochs, then we use cross-entropy loss to train the model with only 1 epoch, because further unconstrained normal training will lead to a decline in certified accuracy as shown in Section 5.6. For the EIBC+IBP training method, we use EIBC triplet loss to train the word embeddings and the IBP training method to train the model simultaneously, with half epochs of the original IBP method. We provide more implementation details in Appendix C.

5.2 Main Results

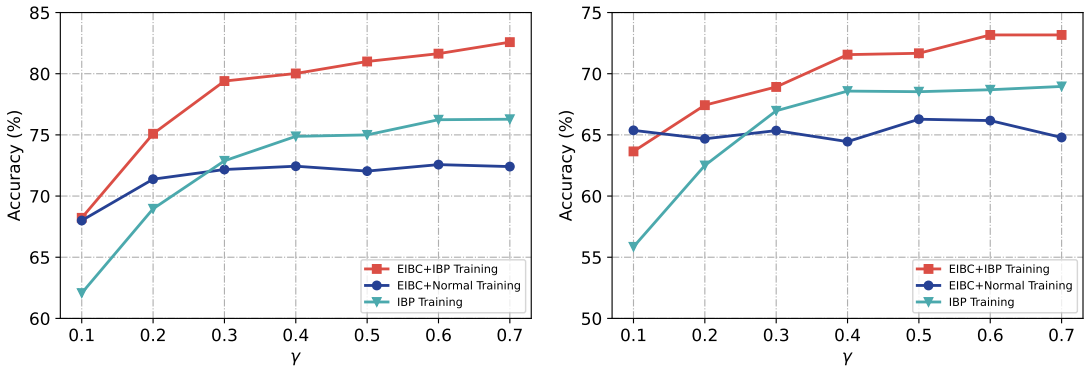
We combine the proposed EIBC with normal training and IBP training, respectively, to boost the certified robustness. Then, we compare them with three state-of-the-art baselines, IBP, SAFER, and CISS, in terms of certified robust accuracy against word substitutions.

As seen from Table 1, EIBC incorporated with normal training already achieves certified robustness to a certain extent without any other defense technique. Especially on the YELP dataset, it gains 89.51% certified robust accuracy, which performs significantly better than SAFER and IBP. Also, EIBC combined with IBP training achieves dominant certified robustness on all datasets with clear margins. For instance, it achieves 84.78% certified robust accuracy on the IMDB dataset, surpassing the original IBP on the TextCNN model by about 8%. This indicates that the tight embedding bounds benefiting from EIBC will considerably boost the performance of IBP.

It is worth noting that though EIBC combined with IBP training is implemented on simple CNN architectures, it achieves higher certified robust accuracy than SAFER and CISS based on large-scale pre-trained BERT models (Devlin et al., 2019), suggesting the superiority and lightness of our approach.

5.3 Generalization to Unseen Substitutions

The defense methods generally assume that the synonym lists used by attackers are known, which is



(a) IMDB (b) SST-2

Figure 2: The certified robust accuracy (%) against unseen word substitutions on IMDB and SST-2 datasets with different γ. The methods are implemented on TextCNN models.

under the ideal assumption. To study the generalization of our method to unseen word substitutions, we only use part of the word substitutions to train the model and all the word substitutions for robust evaluation.

Specifically, for each word with n synonyms, we randomly select its $\lceil \gamma n \rceil$ synonyms ($0 < \gamma \leq 1$) for training, where γ controls the proportion of the seen word substitutions during training. We observe the certified robust accuracy under the word substitutions based on the entire synonyms.

Figure 2 shows the certified robust accuracy with different γ. The performance of IBP decreases rapidly with the decline of γ, but the EIBC combined with normal training is relatively stable, indicating that EIBC has a remarkable generalization to unseen word substitutions. It also suggests that the improvement benefiting from the word embeddings is more generalized than that from other parts of the model under unseen word substitutions. Furthermore, EIBC combined with IBP training achieves the best certified robust accuracy in most cases.

5.4 Empirical Robustness

We utilize the Genetic Attack (GA) (Alzantot et al., 2018) to investigate the empirical robustness of our method. GA generates a population of perturbed texts by random substitutions, then searches and updates the population by the genetic algorithm. Following Jia et al. (2019), we set the population size as 60 and run 40 search iterations on 1,000 testing data randomly sampled from each dataset.

As shown in Table 2, without any defense technique, the genetic attack can dramatically mislead the normally trained model and degrade its accuracy to 8.0% on the IMDB dataset and 40.5% on the YELP dataset. Among all the defense baselines, our proposed method exhibits better performance with a clear margin under GA.

Method	IMDB	YELP
Normal Training	8.00	40.50
IBP Training*	74.90	87.50
EIBC+Normal Training	77.10	90.40
EIBC+IBP Training*	**86.10**	**93.40**

* Our implementation.

Table 2: The empirical robust accuracy (%) against genetic attack on IMDB and YELP datasets. The methods are implemented on TextCNN models.

5.5 Clean Accuracy versus Robust Accuracy

In Eq. (11), our EIBC triplet loss uses hyperparameter α to control the distance between words and their non-synonyms to hold the semantic representation capability of the word embeddings. We use *clean accuracy* to denote the accuracy (%) on clean testing data without any perturbation, and *robust accuracy* to denote the certified robust accuracy (%) against word substitutions.

We observe the trade-off between clean accuracy and robust accuracy controlled by α. As depicted in Figure 3, when α is low, the distances among any words are close, which harms the semantic representation of word vectors and leads to low clean accuracy. Meanwhile, the interval of convex hull constructed by synonyms is also small. Thus, the output bounds are tight, and the gap between robust accuracy and clean accuracy is reduced. Further, when α approaches 0, the term pushing away the non-synonyms in EIBC triplet loss tends to be invalid. The shape decline in clean accuracy in this

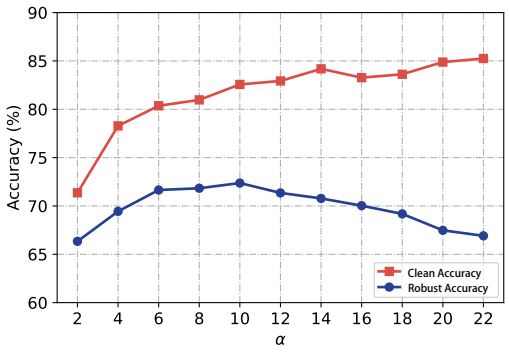


Figure 3: The impact of hyperparameter α on the trade-off between clean accuracy and robust accuracy of EIBC with normal training on the IMDB dataset.

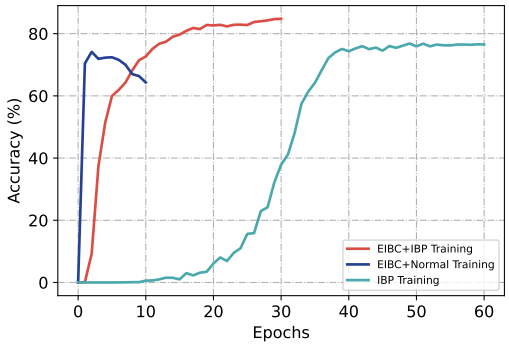


Figure 4: The curve of certified robust accuracy (%) in the training procedure of different methods.

Distance Metric	IMDB	YELP
$p = 1$	84.78	93.66
$p = 2$	81.47	92.68
$p = \infty$	60.60	82.26

Table 3: The certified robust accuracy (%) of models trained by EIBC+IBP Training method using different distance metrics on the IMDB and YELP datasets.

case demonstrates the importance of pushing away non-synonyms. As α grows, the distance between words and their non-synonyms gradually increases, thus ensuring better semantic representation and higher clean accuracy. However, the further increase of α leads to the enlargement of the interval of convex hull formed by synonyms and hinders the robust accuracy.

5.6 Training Procedure

To investigate how the word embeddings pre-trained by EIBC help improve the training process, in Figure 4, we illustrate the changing curve of the certified robust accuracy in the training procedure for IBP, EIBC with normal training, and EIBC with IBP training.

With loose interval bounds, the certified robust accuracy of IBP increases slowly during the training procedure, finally achieving a relatively low certified guarantee. For EIBC combined with normal training, since the word embeddings trained by EIBC have provided the model with initial certified robustness, the model only normally trains one epoch to achieve a certified robust accuracy slightly lower than IBP. However, further normal training without constraint leads to a decline in certified robust accuracy. We could combine EIBC with IBP training to achieve the best certified robust accuracy with half epochs of IBP. These results suggest that tightening word embeddings with EIBC can boost the certified robustness and accelerate the training process of IBP.

5.7 Analysis on Distance Metric

We explore the effect of different l_p-norm distance metrics in Eqs. (9) and (10), such as Manhattan distance ($p = 1$), Euclidean distance ($p = 2$), and Chebyshev distance ($p = \infty$). Table 3 shows the results of models trained by EIBC combined with IBP training on the IMDB and YELP datasets. EIBC with Euclidean distance achieves competitive robustness to EIBC with Manhattan distance. The performance of Euclidean distance and Manhattan distance is relatively close on the two datasets because they can constrain the bound on each dimension in the embedding space. In contrast, the effectiveness of Chebyshev distance is the worst as it can only constrain one dimension, which is inefficient.

6 Conclusion

In this work, we attach importance to word embeddings and prove that the certified robustness can be improved by reducing the interval of the convex hull constructed by synonyms in the embedding space. We introduce a novel loss termed the Embedding Interval Bound Constraint (EIBC) triplet loss to constrain the convex hull. Since EIBC merely provides word embeddings with certified robustness, which is componentized, we could incorporate EIBC into the normal training or IBP training to boost the certified robust accuracy. Experiments on three benchmark datasets show that EIBC combined with IBP training achieves much higher certified robust accuracy than various state-of-the-art defense methods. EIBC also exhibits good generalization to unseen word substitutions. We will further study how to incorporate EIBC

with other certified defense methods in future work. Moreover, we will apply the proposed method in transformer-based models and extend the research to defend against character-level or sentence-level perturbations.

An essential difference between image and text data is that text data is discrete and needs to be transformed into continuous word vectors by word embeddings. Tightened bounds of word embeddings benefiting from EIBC could boost the certified robustness of IBP, which is a typical example to indicate that word embeddings are vital to the robustness of NLP models. We hope our work could inspire more studies on the robustness of NLP models enhanced by word embeddings.

Limitations

As pointed out by Shi et al. (2020), applying IBP technologies to large-scale pre-trained BERT models is challenging because of the calculation of bound propagation on the attention layer is relatively loose. Since BERT is currently one of the most popular architectures in NLP, there is a limitation that the proposed method combined with IBP training could not generalize to BERT architectures. However, it is worth noting that the proposed method based on TextCNN architectures achieves better certified robustness than the advanced baselines, SAFER and CISS based on BERT. Besides, this paper focuses on enhancing the model's robustness to word substitutions, but not investigates the robustness to character-level or sentence-level perturbations.

Acknowledgments

This work is supported by National Natural Science Foundation (62076105,U22B2017).

References

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2021. Towards robustness against natural language word substitutions. In *9th International Conference on Learning Representations*.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2018. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Arthur Mann, and Pushmeet Kohli. 2019. Scalable verified training for provably robust image classification. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 4841–4850.

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 *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4081–4091.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. 2020. Do we need zero training loss after achieving zero training error? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 4604–4614.

Maor Ivgi and Jonathan Berant. 2021. Achieving model robustness through discrete adversarial training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1529–1544.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4129–4142.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

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

Qin Liu, Rui Zheng, Bao Rong, Jingyi Liu, Zhihua Liu, Zhanzhan Cheng, Liang Qiao, Tao Gui, Qi Zhang,

and Xuanjing Huang. 2022. Flooding-X: Improving bert's resistance to adversarial attacks via loss-restricted fine-tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5634–5644.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150.

Rishabh Maheshwary, Saket Maheshwary, and Vikram Pudi. 2021. Generating natural language attacks in a hard label black box setting. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 13525–13533.

Zhao Meng and Roger Wattenhofer. 2020. A geometry-inspired attack for generating natural language adversarial examples. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6679–6689.

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

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, pages 1532–1543.

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

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30, pages 6830–6841.

Zhouxing Shi, Yihan Wang, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2021. Fast certified robust training with short warmup. *Advances in Neural Information Processing Systems*, 34:18335–18349.

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

Chenglei Si, Zhengyan Zhang, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021. Better robustness by more coverage: Adversarial and mixup data augmentation for robust fine-tuning. In *Findings of the Association for Computational Linguistics*, pages 1569–1576.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2153–2162.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations*.

Xiaosen Wang, Jin Hao, Yichen Yang, and Kun He. 2021a. Natural language adversarial defense through synonym encoding. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 823–833.

Xiaosen Wang, Yifeng Xiong, and Kun He. 2022. Detecting textual adversarial examples through randomized substitution and vote. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 2056–2065.

Xiaosen Wang, Yichen Yang, Yihe Deng, and Kun He. 2021b. Adversarial training with fast gradient projection method against synonym substitution based text attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13997–14005.

Yichen Yang, Xiaosen Wang, and Kun He. 2022. Robust textual embedding against word-level adversarial attacks. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 2214–2224.

Mao Ye, Chengyue Gong, and Qiang Liu. 2020. 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*, pages 3465–3475.

Zhen Yu, Xiaosen Wang, Wanxiang Che, and Kun He. 2022. Learning-based hybrid local search for the hard-label textual attack. In *Findings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

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

Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. 2021. Certified robustness to text adversarial attacks by randomized [MASK]. *arXiv preprint arXiv:2105.03743*.

Haiteng Zhao, Chang Ma, Xinshuai Dong, Anh Tuan Luu, Zhi-Hong Deng, and Hanwang Zhang. 2022. Certified robustness against natural language attacks by causal intervention. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26958–26970.

Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. 2019. Learning to discriminate perturbations for blocking adversarial attacks in text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4903–4912.

A Proof of Theorem 1

In Theorem 1, minimizing the objective in Eq. (7) is converted to an optimization objective with respect to the word embeddings. We prove the theorem in two steps. Firstly, we prove the upper bound solution of the optimization objective in Eq. (7) is to minimize the maximum gap between the model's logits and its bound. Secondly, we convert the optimization of the gap to an optimization objective with respect to the word embeddings by back-propagating the interval bound.

Lemma 1 *The upper bound on the solution of Eq. (7) is*

$$\textit{minimize} \quad \max(\left|\underline{\mathbf{z}}^K - \mathbf{z}^K\right|) + \max(\left|\overline{\mathbf{z}}^K - \mathbf{z}^K\right|). \tag{14}$$

where $\max(\cdot)$ *and* $|\cdot|$ *are the element-wise operators.*

Proof of Lemma 1. For a fixed model, we have: $\mathbf{z}^K_{y_{max}} - \mathbf{z}^K_y = (\mathbf{z}^K_{y_{max}} - \underline{\mathbf{z}}^K_{y_{max}}) - (\overline{\mathbf{z}}^K_y - \underline{\mathbf{z}}^K_{y_{max}}) + (\overline{\mathbf{z}}^K_y - \mathbf{z}^K_y)$, which is a constant. Therefore, the optimization objective in Eq. (7) is equivalent to:

$$\begin{aligned} \text{minimize} \quad & (\mathbf{z}^K_{y_{max}} - \underline{\mathbf{z}}^K_{y_{max}}) + (\overline{\mathbf{z}}^K_y - \mathbf{z}^K_y), \\ & \forall y \in \mathcal{Y}, y \neq y_{max}. \end{aligned} \tag{15}$$

Besides, we have the following upper bound relationship:

$$\begin{aligned} (\mathbf{z}^K_{y_{max}} - \underline{\mathbf{z}}^K_{y_{max}}) &\leq \max(\left|\underline{\mathbf{z}}^K - \mathbf{z}^K\right|), \\ (\overline{\mathbf{z}}^K_y - \mathbf{z}^K_y) &\leq \max(\left|\overline{\mathbf{z}}^K - \mathbf{z}^K\right|), \\ & \forall y \in \mathcal{Y}, y \neq y_{max}. \end{aligned} \tag{16}$$

Then, based on Eq. (15) and Eq. (16), we can easily derive that Eq. (14) is the upper bound on the solution of Eq. (7).

□

Bound Backpropagation We back-propagate the interval bounds from the output logits to the embedding space through the network layer by layer. Assuming we have already obtained the interval bounds of layer $k+1$, we need to calculate the bound of the previous layer k. We mainly deal with two cases:

- For an affine transformation, denoted by $\mathbf{z}^{k+1} = \mathbf{W}\mathbf{z}^k + \mathbf{b}$, we have:

$$\begin{aligned} \left|\overline{\mathbf{z}}^{k+1} - \mathbf{z}^{k+1}\right| &= |\mathbf{W}| \left|\overline{\mathbf{z}}^k - \mathbf{z}^k\right|, \\ \left|\underline{\mathbf{z}}^{k+1} - \mathbf{z}^{k+1}\right| &= |\mathbf{W}| \left|\underline{\mathbf{z}}^k - \mathbf{z}^k\right|, \end{aligned} \tag{17}$$

  where $|\cdot|$ is the element-wise absolute value operator.

- For an element-wise monotonic activation function (*e.g.* ReLU, tanh, sigmoid), denoted by $\mathbf{z}^{k+1} = h(\mathbf{z}^k)$, we have:

$$\begin{aligned} \left|\overline{\mathbf{z}}^{k+1} - \mathbf{z}^{k+1}\right| &\leq C_a \left|\overline{\mathbf{z}}^k - \mathbf{z}^k\right|, \\ \left|\underline{\mathbf{z}}^{k+1} - \mathbf{z}^{k+1}\right| &\leq C_a \left|\underline{\mathbf{z}}^k - \mathbf{z}^k\right|, \end{aligned} \tag{18}$$

  where C_a is the Lipschitz constant of the activation function.

For $\mathbf{z}^0 \in \mathbb{R}^{N*D}$, we use $\max^*(\cdot)$ to denote the max operator over each dimension of the embedding space, and we have $\max^*(\mathbf{z}^0) \in \mathbb{R}^D$. With the bound backpropagation, we have:

$$\begin{aligned} \left|\overline{\mathbf{z}}^K - \mathbf{z}^K\right| &\leq \mathbf{C_1} \max^*(\left|\overline{\mathbf{z}}^0 - \mathbf{z}^0\right|), \\ \left|\underline{\mathbf{z}}^K - \mathbf{z}^K\right| &\leq \mathbf{C_2} \max^*(\left|\underline{\mathbf{z}}^0 - \mathbf{z}^0\right|), \end{aligned} \tag{19}$$

where $\mathbf{C}_1$ and $\mathbf{C_2}$ are calculated by interval bound backpropagation, and they are constant matrices for a fixed model. Then, we can derive the upper bound of the optimization objective in Eq. (14):

$$\text{minimize} \quad \max^*(\left|\underline{\mathbf{z}}^0 - \mathbf{z}^0\right|) + \max^*(\left|\overline{\mathbf{z}}^0 - \mathbf{z}^0\right|). \tag{20}$$

According to Eq. (5), we have:

$$\begin{aligned} \max^*(\left|\underline{\mathbf{z}}^0 - \mathbf{z}^0\right|) &\leq \max(\max_{x_i \in \mathbf{x}} (\max_{x_i' \in \mathcal{S}(x_i)} (\left|\varphi(x_i) - \varphi(x_i')\right|))), \\ \max^*(\left|\overline{\mathbf{z}}^0 - \mathbf{z}^0\right|) &\leq \max(\max_{x_i \in \mathbf{x}} (\max_{x_i' \in \mathcal{S}(x_i)} (\left|\varphi(x_i) - \varphi(x_i')\right|))), \end{aligned} \tag{21}$$

and then we can construct the upper bound on the solution of Eq. (20):

$$\text{minimize} \quad \max_{x_i \in \mathbf{x}} \left(\max_{x_i' \in \mathcal{S}(x_i)} \left(\left| \varphi(x_i) - \varphi(x_i') \right| \right) \right). \tag{22}$$

Based on Lemma 1, we can derive that Eq. (22) is the upper bound on the solution of Eq. (7). □

B Interval Bound Propagation

Here we give a brief description of Interval Bound Propagation (IBP) (Gowal et al., 2018; Jia et al., 2019) on its calculation of bound propagation and training loss.

Bound Propagation For Eq. (6), IBP provides corresponding calculation methods for affine layers and monotonic activation functions:

- For the affine transformation, denoted by $\mathbf{z}^{k+1} = \mathbf{W}\mathbf{z}^k + \mathbf{b}$, we have:

$$\begin{aligned} \mathbf{u}^{k+1} &= \frac{1}{2}\mathbf{W}(\overline{\mathbf{z}}^k + \underline{\mathbf{z}}^k) + \mathbf{b}, \\ \mathbf{r}^{k+1} &= \frac{1}{2}|\mathbf{W}|(\overline{\mathbf{z}}^k - \underline{\mathbf{z}}^k), \\ \overline{\mathbf{z}}^{k+1} &= \mathbf{u}^{k+1} + \mathbf{r}^{k+1}, \\ \underline{\mathbf{z}}^{k+1} &= \mathbf{u}^{k+1} - \mathbf{r}^{k+1}, \end{aligned} \tag{23}$$

  where $|\cdot|$ is the element-wise absolute value operator.

- For the element-wise monotonic activation function (*e.g.* ReLU, tanh, sigmoid), denoted by $\mathbf{z}^{k+1} = h(\mathbf{z}^k)$, we have:

$$\begin{aligned} \overline{\mathbf{z}}^{k+1} &= h(\overline{\mathbf{z}}^k), \\ \underline{\mathbf{z}}^{k+1} &= h(\underline{\mathbf{z}}^k). \end{aligned} \tag{24}$$

IBP Loss For the interval bounds calculated by Eq. (5), the IBP method scales them with scalar ϵ:

$$\begin{aligned} \underline{z}_{ij}^0(\epsilon) &= z_{ij}^0 - \epsilon(z_{ij}^0 - \underline{z}_{ij}^0), \\ \overline{z}_{ij}^0(\epsilon) &= z_{ij}^0 + \epsilon(\overline{z}_{ij}^0 - z_{ij}^0). \end{aligned} \tag{25}$$

Using bound propagation, we can get the lower bound and upper bound of logits with the scalar ϵ: $\underline{\mathbf{z}}^K(\epsilon)$ and $\overline{\mathbf{z}}^K(\epsilon)$, respectively. Similar to Eq. (3), we can get the worst-case logits and use them to construct the IBP loss:

$$\mathcal{L}_{IBP}(\epsilon) = \mathcal{L}_{CE}(\mathbf{z}_{worst}^K(\epsilon), y_{true}), \tag{26}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss and $\mathbf{z}_{worst}^K(\epsilon)$ is the worst-case logits:

$$\mathbf{z}_{worst}^K(\epsilon) = \begin{cases} \underline{\mathbf{z}}_{y_{true}}^K(\epsilon) & \text{if } y = y_{true}, \\ \overline{\mathbf{z}}_y^K(\epsilon) & \text{otherwise.} \end{cases} \tag{27}$$

Then, IBP loss can be combined with normal cross-entropy loss to train the model and boost the certified robust accuracy:

$$\mathcal{L}_{model} = (1-\beta)\mathcal{L}_{CE}(\mathbf{z}^K, y_{true}) + \beta\mathcal{L}_{IBP}(\epsilon). \tag{28}$$

C More Experimental Details

C.1 Dataset Statistics

IMDB is a binary sentiment classification dataset with 25,000 training data and 25,000 testing data. YELP is much larger, with 560,000 training data and 38,000 testing data. SST-2 is one of the classification tasks from GLUE (Wang et al., 2019) and contains 67,350 training data and 873 development data.

C.2 Detailed Setup

For the EIBC+Normal Training method, we divide the overall training process into two steps. In the first step, we use EIBC triplet loss to fine-tune the pretrained word embeddings, namely GloVe word embeddings (Pennington et al., 2014). We use the constant learning rate in the first e_{emb_1} epochs and the cosine decay learning rate schedule in the last e_{emb_2} epochs to decrease the learning rate to 0. In the second step, we freeze the embedding layer and use the normal cross-entropy loss to train the model with e_{model} epochs.

For the EIBC+IBP training method, we use EIBC triplet loss to train the word embeddings and the IBP training method to train the model simultaneously. We use the constant learning rate in the first e_1 epochs and the cosine decay learning rate schedule in the last e_2 epochs to decrease the learning rate to 0. For implementing the IBP training method, following Jia et al. (2019), we use a linear warmup over ϵ and β in the first e_1 epochs from ϵ_{start} to ϵ_{end} and β_{start} to β_{end}, respectively.

All the experiments are run for five times on a single NVIDIA-RTX 3090 GPU and the median of the results is reported. We provide the details of the EIBC+Normal training and EIBC+IBP training method in Table 4 and Table 5, respectively.

Dataset	IMDB	YELP	SST-2
Optimizer	Adam($\beta_1 = 0.9, \beta_2 = 0.999$)		
Batch size	32		
Learning rate	10^{-3}		
Weight decay	10^{-4}		10^{-3}
e_{emb_1}	15		
e_{emb_2}	5		
e_{model}	1		
α	10.0		
Total epochs	21		
GPU hours	0.2	0.5	0.2

Table 4: Training configuration and hyperparameters of EIBC+Normal training method. GPU hours are tested on a single NVIDIA-RTX 3090 GPU.

Dataset	IMDB	YELP	SST-2
Optimizer	Adam($\beta_1 = 0.9, \beta_2 = 0.999$)		
Batch size	32		
Learning rate	10^{-3}		
Weight decay	10^{-4}		10^{-3}
e_1	20	10	
e_2	10	5	
ϵ_{start}	0.0		
ϵ_{end}	1.0		
β_{start}	0.0		
β_{end}	1.0		
α	10.0		
Total epochs	30	15	
GPU hours	0.5	1.5	0.5

Table 5: Training configuration and hyperparameters of EIBC+IBP training method. GPU hours are tested on a single NVIDIA-RTX 3090 GPU.

Our implementation of the IBP training method follows the original settings described in Jia et al. (2019) except for a few differences below:

- We do not use early stopping but instead the cosine decay learning rate schedule to stabilize the training process.
- Jia et al. (2019) removes the words that are not in the vocabulary of the counter-fitted GloVe word embeddings space (Mrksic et al., 2016) from the input text data. However, some datasets, such as YELP, contain some short text samples, and such a pre-processing approach would result in no words existing. We retain all the words that appear in the vocabulary of the original GloVe word embeddings, which is a much larger vocabulary. We also show the model performance on the IMDB dataset under the two pre-processing approaches. The results are in Table 6.
- We set the β_{end} to 1.0 instead of 0.8 towards higher certified robust accuracy.

Method	**Vocab**	**IMDB**
IBP Training*	CF	76.16
	GloVe	76.73
EIBC+Normal Training	CF	69.54
	GloVe	72.37
EIBC+IBP Training*	CF	82.40
	GloVe	**84.78**

* Our implementation.

Table 6: The certified robust accuracy (%) against word substitutions on the IMDB dataset with different vocabulary. The methods are implemented on TextCNN models. CF means vocabulary of counter-fitted word embeddings.

Method	**Model**	**IMDB**
IBP Training*	CNN	76.00
	TextCNN	76.73
EIBC+Normal Training	CNN	72.22
	TextCNN	72.37
EIBC+IBP Training*	CNN	84.40
	TextCNN	**84.78**

* Our implementation.

Table 7: The certified robust accuracy (%) of models with different architectures and defense methods on the IMDB dataset.

C.3 Robustness on Different Architectures

We implement IBP, EIBC with normal training, and EIBC with IBP training on two architectures, *i.e.*, CNN and TextCNN. As shown in Table 7, using the same architectures, EIBC combined with IBP training performs better than IBP on both CNN and TextCNN models. Using the same training method, the models based on the TextCNN architecture perform better than that based on the CNN architecture, because TextCNN is more complicated.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Our work focuses on improving the robustness of NLP models without potential risks as far as we know.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

5

☑ B1. Did you cite the creators of artifacts you used?
5

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. We use publicly available and commonly used datasets for classification tasks.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
5

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. We use publicly available and commonly used datasets for classification tasks.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. We use publicly available and commonly used datasets for classification tasks.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
The number of examples, details of train/test/dev splits

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5, Appendix C

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5, Appendix C

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5, Appendix C

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
5, Appendix C

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.