CERT-RNN: Towards Certifying the Robustness of Recurrent Neural Networks

Tianyu Du
Zhejiang University
zjradty@zju.edu.cn

Shouling Ji*
Zhejiang University
Binjiang Institute of Zhejiang University
sji@zju.edu.cn

Lujia Shen
Zhejiang University
shen.lujia@zju.edu.cn

Yao Zhang
Zhejiang University
y.zhang@zju.edu.cn

Jinfeng Li
Zhejiang University
lijinfeng_0713@zju.edu.cn

Jie Shi
Huawei International, Singapore
shi.jie1@huawei.com

Chengfang Fang
Huawei International, Singapore
fang.chengfang@huawei.com

Jianwei Yin
Zhejiang University
Binjiang Institute of Zhejiang University
zjuyjw@zju.edu.cn

Raheem Beyah
Georgia Institute of Technology
rbeyah@ece.gatech.edu

Ting Wang
Pennsylvania State University
inbox.ting@gmail.com

ABSTRACT

Certifiable robustness, the functionality of verifying whether the given region surrounding a data point admits any adversarial example, provides guaranteed security for neural networks deployed in adversarial environments. A plethora of work has been proposed to certify the robustness of feed-forward networks, e.g., FCNs and CNNs. Yet, most existing methods cannot be directly applied to recurrent neural networks (RNNs), due to their sequential inputs and unique operations.

In this paper, we present CERT-RNN, a general framework for certifying the robustness of RNNs. Specifically, through detailed analysis for the intrinsic property of the unique function in different ranges, we exhaustively discuss different cases for the exact formula of bounding planes, based on which we design several precise and efficient abstract transformers for the unique calculations in RNNs. CERT-RNN significantly outperforms the state-of-the-art methods (e.g., POPQORN [25]) in terms of (i) effectiveness – it provides much tighter robustness bounds, and (ii) efficiency – it scales to much more complex models. Through extensive evaluation, we validate CERT-RNN's superior performance across various network architectures (e.g., vanilla RNN and LSTM) and applications (e.g., image classification, sentiment analysis, toxic comment detection, and malicious URL detection). For instance, for the RNN-2-32 model on the MNIST sequence dataset, the robustness bound certified by CERT-RNN is on average 1.86 times larger than that by POPQORN. Besides certifying the robustness of given RNNs, CERT-RNN also enables a range of practical applications including *evaluating the provable effectiveness for various defenses* (i.e., the defense with a larger robustness region is considered to be more robust), *improving the robustness of RNNs* (i.e., incorporating CERT-RNN with verified robust training) and *identifying sensitive words* (i.e., the word with the smallest certified robustness bound is considered to be the most sensitive word in a sentence), which helps build more robust and interpretable deep learning systems. We will open-source CERT-RNN for facilitating the DNN security research.

CCS CONCEPTS

• **Computing methodologies** → *Natural language processing*; • **Security and privacy** → *Formal methods and theory of security*.

KEYWORDS

deep learning, recurrent neural networks, robustness certification, natural language processing

ACM Reference Format:
Tianyu Du, Shouling Ji, Lujia Shen, Yao Zhang, Jinfeng Li, Jie Shi, Chengfang Fang, Jianwei Yin, Raheem Beyah, and Ting Wang. 2021. CERT-RNN: Towards Certifying the Robustness of Recurrent Neural Networks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3460120.3484538

*Shouling Ji is the corresponding author.

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8454-4/21/11…$15.00
https://doi.org/10.1145/3460120.3484538

1 INTRODUCTION

The recent advances in deep learning have achieved remarkable success in a large number of tasks, such as image classification [47, 48], natural language processing (NLP) [26, 31] and speech recognition [3, 5]. Nevertheless, it is now known that deep neural networks (DNNs) are fundamentally vulnerable to malicious manipulations, such as adversarial examples that force target DNNs to misbehave [45], which significantly hinders their application in security-sensitive domains.

Thus far intensive research has been devoted to improving the robustness of DNNs against adversarial attacks [19, 27, 28, 30, 32, 36, 50]. Unfortunately, most defenses are based on heuristics and thus lack any theoretical guarantee, which can often be defeated or circumvented by more powerful attacks. Aiming to end the constant arms race between adversarial attacks and defenses, the concept of *certifiable robustness* is proposed to provide guaranteed robustness by formally verifying whether a given region surrounding a data point admits any adversarial example [18, 24, 44].

There has been substantial work on robustness certification using techniques including satisfiability modulo theories (SMT) [13, 21, 24], mixed-integer linear programming (MILP) [1], convex polytope [50], and reachability analysis [49, 52]. However, designed for feed-forward networks such as fully connected networks (FCNs) and convolutional neural networks (CNNs), these methods cannot be directly extended to recurrent neural networks (RNNs). The major challenges stem from the sequential inputs of RNNs (while previous certification methods usually assume that the inputs were fed into the model at the bottom layer) and the operations unique to RNNs (e.g., multiplication of multiple variables). Until recently, some attempts have been made to certify the robustness of RNNs (e.g., POPQORN [25]). Unfortunately, the existing methods are limited in terms of both precision – relying on techniques such as interval arithmetic, they fail to capture the inter-variable correlation, resulting in overly loose robustness bounds; and efficiency – due to the expensive approximation of bounding planes, they fail to scale up to complex RNN models, making them unsuitable for many practical applications.

To address the above challenges, in this work, we propose Cert-RNN, a general robustness certification framework for RNNs. Specifically, leveraging *abstract interpretation* [10], Cert-RNN first maps all the possible inputs of a RNN to an *abstract domain* (e.g., zonotope), which retains the inter-variable correlation. Further, Cert-RNN adopts a set of precise and efficient abstract transformers for the operations unique to RNNs. Finally, by solving an optimization problem, Cert-RNN obtains tight robustness bounds with respect to given inputs. Thanks to its novel design, Cert-RNN significantly outperforms prior work in terms of both precision and efficiency; that is, it provides much tighter robustness bounds yet with much higher execution efficiency.

We conduct extensive evaluation to empirically validate Cert-RNN's performance across various network architectures (e.g., vanilla RNN and LSTM) and several security-sensitive applications with sequential inputs, including image classification, sentiment analysis, toxic comment detection, and malicious URL detection. Experimental results demonstrate that Cert-RNN estimates the robustness bound in a more precise and efficient manner compared with the state-of-the-art methods. For instance, on the MNIST sequence dataset, the Cert-RNN robustness bound is 1.86 times of that by the state-of-the-art POPQORN for the RNN-2-32 model, and for the LSTM-1-32 model, the running time of POPQORN is 46.78 minutes on average while Cert-RNN only consumes 2.66 minutes on average. To further demonstrate Cert-RNN's practical utility, we apply the robustness bounds suggested by Cert-RNN to certify the effectiveness of different adversarial defense methods as well as to identify sensitive words. We find that heuristic defense methods such as FGSM-AT [17] and PGD-AT [32] defend a DNN in a way that makes it only slightly more provably robust, while the provable defense IBP-VT [18] provides a significant increase in provable robustness. Furthermore, we incorporating Cert-RNN with verified robust training and demonstrate its superiority in improving the robustness of RNNs compared with the baselines. In addition, we find that the robustness bounds certified by Cert-RNN can be used to distinguish the importance of different words for RNN classification tasks, which is meanwhile consistent with human cognition and thus is very helpful for explaining the prediction of RNNs. Therefore, the proposed robustness certification framework can help users comprehensively evaluate the effectiveness of various defenses and build more robust intelligent systems, and can provide a meaningful quantitative metric for improving the interpretability of RNNs.

Our Contributions. Our main contributions are summarized as follows.

- We identify the primary limitations of existing RNN certification methods. Since their techniques cannot capture the inter-variable correlations in RNNs, they can only certify overly loose robustness bounds. In addition, they fail to scale up to complex RNN models, e.g., the popular LSTMs, due to their expensive computational cost for approximating the bounding planes.
- Leveraging abstract interpretation, we propose a novel certification framework for RNNs – Cert-RNN, which significantly outperforms prior work in terms of both precision and efficiency. Specifically, we use the abstract domain to retain the inter-variable correlation. Then, through detailed analysis of the intrinsic property of the function in different ranges, we exhaustively analyze different cases for all possible formulas of bounding planes and design a number of precise and efficient abstract transformers for the unique calculations in RNNs, which finally yields much more precise robustness bounds in a more efficient manner.
- We conduct extensive evaluation on four security-sensitive applications across various network architectures to empirically validate Cert-RNN's superiority. Experimental results confirm that Cert-RNN certifies the robustness bound in a more precise and efficient manner compared with the state-of-the-art methods.
- We show that the robustness bound certified by Cert-RNN can be practically used as a meaningful quantitative metric for evaluating both the interpretability of RNNs and the provable effectiveness of various defense methods. We also demonstrate Cert-RNN's superiority in improving the robustness of RNNs. In addition, we discuss Cert-RNN's further extension, such as supporting more kinds of norm-bounded attacks and more RNN types, as well as being incorporated with robust training to design

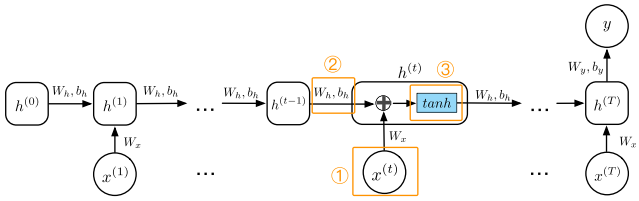


Figure 1: The architecture of a vanilla RNN.

a provably robust defense. We believe these extensions would further improve Cert-RNN's applicability and foster future research in AI security. We will open-source Cert-RNN for facilitating the DNN security research.

2 RELATED WORK

2.1 Adversarial Attacks & Defenses

Szegedy *et al.* firstly showed that DNNs are vulnerable to small perturbations on inputs [45]. Since then, a plethora of work has focused on constructing adversarial examples in various domains, including computer vision [8, 17, 32], natural language processing [16, 29, 39], and speech recognition [11, 38, 51]. The phenomenon of adversarial examples demonstrates the inherent lack of robustness of DNNs, which limits their use in security-critical applications. Meanwhile, many techniques have been proposed to defend against adversarial examples [30, 32, 36]. Unfortunately, most defenses are only effective in limited scenarios and would be defeated by later stronger adversarial attacks [4]. For instance, Ebrahimi *et al.* proposed to use adversarial training [17] against character-level perturbations [12]. Later, this defense was penetrated by a new attack, running a more expensive search procedure at the test time [29]. Therefore, defenses only showing empirical success against attacks [32], are difficult to be concluded robust. In order to guarantee the robustness of DNNs, provable defenses were proposed [18] with the expectation of providing certifiable robustness for DNNs.

2.2 Robustness Certification

Recently, disparate robustness certification methods have been proposed to certify the robustness bound of neural networks, i.e., within this bound, any possible perturbation would not impact the prediction of a neural network. These methods can be generally categorized as *exact certification* methods and *relaxed certification* methods. *Exact certification* methods are mostly based on satisfiability modulo theories (SMT) [13, 21, 24] or mixed-integer linear program (MILP) solvers [1]. Though these methods are able to certify the exact robustness bound, they are usually computationally expensive. Hence, it is difficult to scale them even to medium size networks. *Relaxed certification* methods include the convex polytope methods [50], reachability analysis methods [49, 52], and abstract interpretation methods [33, 44], etc. These methods are usually efficient but cannot provide precise bounds as exact certification methods do. Nevertheless, considering the expensive computational cost, relaxed certification methods are shown to be more promising in practical applications, especially for large networks. Another similar task is *verification of networks* [18, 41], which formally proves that a given input with any perturbation less than ϵ will not be misclassified by a neural network.

However, the aforementioned methods mainly focus on certifying networks with relatively simple architectures such as FCNs and CNNs, while few of them are able to handle complicated RNNs. The most relevant work to ours is POPQORN [25], which leverages interval arithmetic to approximate the non-linearities for RNNs. Our Cert-RNN differs from POPQORN mainly in two perspectives. First, POPQORN is imprecise as interval arithmetic does not keep the inter-variable correlation, while our method does. Therefore, our method can more precisely certifiy the robustness of RNNs than POPQORN, as the results shown in Section 5. Second, POPQORN's approximations for the non-linearities are very slow, which makes their approximations inefficient and thus unsuitable for practical applications. In contrast, leveraging our proposed abstract transformers for RNNs' non-linearities, we can certify the robustness of RNNs very efficiently (e.g., 23-times faster than POPQORN on the MNIST sequence dataset for the LSTM-1-32 model).

3 BACKGROUND

In this section, we first describe the types of RNNs we consider. Then, we introduce the basic concepts of abstract interpretation which form the building blocks of Cert-RNN.

3.1 Recurrent Neural Networks (RNNs)

Vanilla RNN. We first demonstrate the essential of our proposed method with a fundamental vanilla RNN as shown in Fig. 1. On an input sequence $\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(t-1)}, \boldsymbol{x}^{(t)}, \cdots, \boldsymbol{x}^{(T)}]$, at time step $t < T$, the vanilla RNN updates the hidden state with $\boldsymbol{h}^{(t)} = \tanh(\boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_x \boldsymbol{x}^{(t)} + \boldsymbol{b}_h)$, and at the last time step T, it computes output $\boldsymbol{y} = \boldsymbol{W}_y \boldsymbol{h}^{(T)} + \boldsymbol{b}_y$, where $\boldsymbol{W}.$ and $\boldsymbol{b}.$ are weight matrices and biases of the cell, respectively.

Long Short-Term Memory (LSTM) Networks. We then extend our proposed method to the more general and widely applied LSTM neural networks [20], as shown in Fig. 2, where each neuron is defined to be a gated cell with memory. LSTM networks operate by maintaining both a hidden state and memory at each time step, which effectively account for the temporal behavior by capturing the history from sequential information [37] and are less vulnerable to the vanishing or exploding gradient problems [7]. Thus, they are more popular models for sequential architectures. Without loss of generality, we consider the following definitions of updates in the LSTM unit: $\boldsymbol{f}^{(t)} = \left[\boldsymbol{x}^{(t)}, \boldsymbol{h}^{(t-1)}\right] \boldsymbol{W}_f + \boldsymbol{b}_f$, $\boldsymbol{o}^{(t)} = \left[\boldsymbol{x}^{(t)}, \boldsymbol{h}^{(t-1)}\right] \boldsymbol{W}_o + \boldsymbol{b}_o$, $\boldsymbol{i}^{(t)} = \left[\boldsymbol{x}^{(t)}, \boldsymbol{h}^{(t-1)}\right] \boldsymbol{W}_i + \boldsymbol{b}_i$, $\tilde{\boldsymbol{c}}^{(t)} = \left[\boldsymbol{x}^{(t)}, \boldsymbol{h}^{(t-1)}\right] \boldsymbol{W}_{\tilde{c}} + \boldsymbol{b}_{\tilde{c}}$, $\boldsymbol{c}^{(t)} = \sigma\left(\boldsymbol{f}^{(t)}\right) \odot \boldsymbol{c}^{(t-1)} + \sigma\left(\boldsymbol{i}^{(t)}\right) \odot \tanh\left(\tilde{\boldsymbol{c}}^{(t)}\right)$, $\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{o}^{(t)}\right) \odot \tanh\left(\boldsymbol{c}^{(t)}\right)$, and $\boldsymbol{y} = \boldsymbol{W}_y \boldsymbol{h}^{(T)} + \boldsymbol{b}_y$, where $[\cdot, \cdot]$ is the horizontal concatenation of two row vectors, σ is the sigmoid function, and $\odot$ represents the Hadamard product between two vectors. At time step t, $\boldsymbol{c}^{(t)}$ represents the cell state, $\tilde{\boldsymbol{c}}^{(t)}$ represents the pre-calculation of the cell state, and $\boldsymbol{f}^{(t)}$, $\boldsymbol{i}^{(t)}$, $\boldsymbol{o}^{(t)}$ represent pre-activations of the forget, input, and output gates, respectively.

3.2 Abstract Interpretation

Abstract interpretation is a general theory for approximating a potentially infinite set of behaviors with a finite representation [10]. A high-level illustration of abstract interpretation is shown in Fig. 3. Overall, any potential adversarial input (e.g., an input

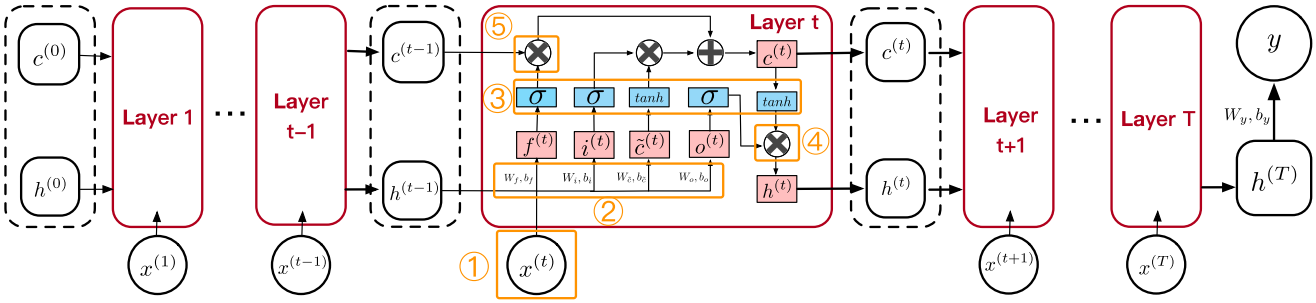


Figure 2: The architecture of an LSTM.

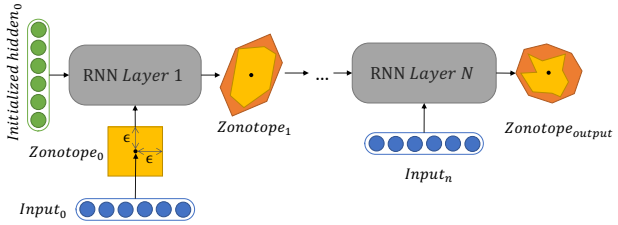


Figure 3: A high-level illustration of abstract interpretation.

that yields a different classification by adding a small perturbation with ℓ_p-norm no larger than ϵ) can be captured using an abstract polygon (e.g., the yellow $Zonotope_0$ in Fig. 3). Then, this abstract polygon is propagated through the given model and finally we can obtain the output abstract polygon (e.g., the orange $Zonotope_{output}$ in Fig. 3), which can be used to bound the possible outputs (i.e., the yellow region in $Zonotope_{output}$). Following this procedure, theoretically, we can certify the robustness for a neural network.

In this paper, we propose leveraging the idea of abstract interpretation to certify the robustness of RNNs. Although this idea has previously been applied to certify FCNs and CNNs [14, 33, 44], there are several fundamental challenges in applying it to RNNs, as discussed in Section 1. The basic idea of our approach is to propagate the possible perturbations (captured by a convex region in the abstract domain) through the operations of the entire RNN pipeline and use the final output region to certify the robustness space. We introduce several useful terms used in this paper below.

Abstract Domain. An abstract domain consists of shapes expressible as a set of logical constraints. A few popular abstract domains are: box, zonotope [15], etc. There are tradeoffs between precision versus scalability in choosing the abstract domain. For instance, box is faster than zonotope, whereas zonotope provides tighter bounds. In this work, we use zonotope to compute the bounds for certification considering the following two advantages: (i) zonotope can preserve the inter-variable correlation; (ii) the abstraction for affine functions (such as the transition function of a fully connected layer) will not lose any precision within the zonotope abstract domain. As shown in Fig. 3, the original input data and its all possible adversarial variants can be captured in $Zonotope_0$.

Abstract Transformer. An abstract transformer of a non-linear function works with abstract elements (drawn from the given abstract domain) and over-approximates the effect of the given function. For instance, as shown in Fig. 3, the abstract transformer will take input data in a zonotope (e.g., the yellow $Zonotope_0$) and output another zonotope (e.g., the orange $Zonotope_1$) which over-approximates the behavior of the model. Logically, we can certify a RNN by abstracting its affine functions and non-linear functions in zonotopes.

Property Certification. After obtaining the output zonotope from the last abstract transformer, we can verify various properties of interest that may hold for the output zonotope. In general, if a property (e.g., the confidence value of the correct label is always larger than that of any wrong label) holds for the output zonotope, we can deduce that the property holds for all possible perturbations of the input. Therefore, we can certify a robustness bound wherein there is no adversarial input. As an abstract transformer is an over-approximation, the certified bound is usually the lower bound of the exact robustness bound. The precision of the certified bound relies on the precision of the abstract transformer.

In summary, our method consists of the following steps: (1) find a suitable abstract domain; (2) construct abstract transformers as precise and efficient as possible; and (3) certify that the desired property holds for the output zonotope. In the next section, we will introduce a number of novel abstract transformers to abstract the non-linear operations used in RNNs, followed by our RNN robustness certification framework.

4 CERT-RNN: CERTIFYING THE ROBUSTNESS OF RNNS

In this section, we present CERT-RNN, a framework for robustness certification for RNNs based on abstract interpretation. As shown in Fig. 3, a zonotope abstract domain is first defined to capture all potential adversarial inputs. Then, CERT-RNN will verify the desired property by propagating the zonotope through all the layers of the target RNN. Specifically, an abstract transformer is created for each non-linear operation of the RNN, which takes a zonotope as input and outputs a new zonotope. The input zonotope represents an abstraction of the possible input of an operation while the output zonotope abstracts the possible output of the operation corresponding to the input zonotope. Finally, the output zonotope of the RNN's last layer is used to certify the robustness.

In this paper, we follow the same threat model in [8], where the attacker is able to add noise δ to the original input $\boldsymbol{x}$ so as to obtain a perturbed input $\boldsymbol{x'} = \boldsymbol{x}+\delta$. The ℓ_∞-norm of the noise δ is assumed not to be larger than constant ϵ defined by the threat model. In addition, we will discuss other norms of noise in Section 7.

4.1 Preliminaries

As illustrated in Section 3.2, our design is based on the zonotope abstract domain. We first introduce the formal definition of a zonotope as well as the preliminaries of zonotope approximation, leveraging on which we give the problem definition of finding the tightest bound for output zonotopes.

Definition 4.1. Given $\alpha_0, \alpha_i \in \mathbb{R}^K, \epsilon_i \in [-1, 1]$ for $1 \le i \le N$, a zonotope $\mathbf{z} \in \mathbb{R}^K$ building on the affine arithmetic is defined as:

$$\mathbf{z} := \alpha_0 + \sum_{i=1}^{N} \alpha_i \cdot \epsilon_i, \tag{1}$$

where ϵ_i is a set of error terms, and each element of $\mathbf{z}$ is denoted by $z_k = \alpha_{0k} + \sum_j \alpha_{jk}\epsilon_j$. Since the zonotope is a center-symmetric closed convex polyhedron, the coefficient α_0 represents the center of the zonotope, and α_i represents the partial deviations around the center.

Definition 4.2. Given a continuously differentiable non-linear function $f(x_1, x_2, \cdots)$ defined in a zonotope, the zonotope approximation for f consists of two parallel planes: the lower bounding plane Z^L and the upper bounding plane Z^U. We define Z^L and Z^U for any $(x_1, x_2, \cdots) \in \mathbf{z}$ as follows:

$$Z^L = C_1 + a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots,$$
$$Z^U = C_2 + a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots,$$

where $C_1, C_2, a_i \in \mathbb{R}$. Note that, when $a_i = 0$ $(i = 1, 2, \cdots)$, the zonotope approximation returns the interval range of f, i.e., $[C_1, C_2]$, which is also the case in [18, 23].

Problem Definition. Given a non-linear function f and its bounding planes Z^L, Z^U, its output region can be bounded by a zonotope $z_o = a_1 \cdot z_1 + a_2 \cdot z_2 + \cdots + \frac{C_2 - C_1}{2}\epsilon_{new}$, where ϵ_{new} is a *new error term* which is introduced from the zonotope approximation for f. Thus, the problem to find the tightest bound of z_o can be formalized as bellow:

$$\min \quad \frac{C_2 - C_1}{2}.$$

4.2 Warm-up: Vanilla RNN Certification

Now, we start the design of Cert-RNN. Since vanilla RNN is the most fundamental RNN model, we first certify its robustness bound. In the following, we introduce how to abstract the adversarial input region (① in Fig. 1) and the zonotope for intermediate operations (②, ③ in Fig. 1), based on which we can conduct robustness certification.

4.2.1 Adversarial Input Region Abstraction. Given an input sequence $\mathbf{X} = [\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}, \mathbf{x}^{(t+1)}, \cdots, \mathbf{x}^{(T)}]$, where $\mathbf{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \cdots, x_K^{(t)}]$ represents the t-th input frame. Based on Definition 4.1, the input frame $\mathbf{x}^{(t)}$ is mapped to the center coefficient α_0 of a zonotope $\mathbf{z}$ as shown in Fig. 3. For ℓ_∞-norm bounded attack, the adversarial perturbation of the j-th dimension of $\mathbf{x}^{(t)}$ is mapped to the coefficient α_{ij}. Then, the lower bound $\mathbf{l}_z \in \mathbb{R}^K$ and the upper bound $\mathbf{u}_z \in \mathbb{R}^K$ of $\mathbf{z}$ can be simply derived by computing the minimum and maximum values of $\mathbf{z}$, respectively.

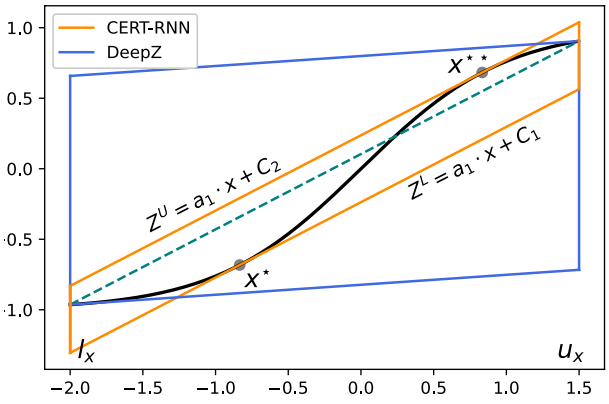


Figure 4: Comparison of zonotope approximations of tanh. The black curve is the tanh function, and the green dashed line connects the highest point and the lowest point of tanh.

4.2.2 Intermediate Operation Abstraction. After mapping the adversarial inputs to the abstract domain, we obtain a zonotope containing all possible adversarial inputs. Now, we discuss how to abstract various kinds of operations in a vanilla RNN.

Affine Transformation Abstraction. Following the RNN updating process ② shown in Fig. 1, we observe that pre-activations of hidden states and gates are affine transformations which can be exactly captured in our approximation, i.e., the zonotope approximation of an affine function preserves its precision [14, 33]. Therefore, given a zonotope $\mathbf{z} = \alpha_0 + \sum_i \alpha_i \epsilon_i$ and an affine function $\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}$, the output zonotope is exactly $\mathbf{W}\alpha_0 + \mathbf{b} + \mathbf{W}\alpha_i\epsilon_i$.

Tanh Function Abstract Transformer. For the **tanh** function (③ in Fig. 1), Singh *et al.* proposed a zonotope approximation method in *DeepZ* [43], whose graphic illustration is shown in Fig. 4. Specifically, the slope of the bounding lines of *DeepZ* is determined by min{*the slope of the highest point*, *the slope of the lowest point*}. Based on this slope, the upper bounding line is determined by the highest point, and the lower bounding line is determined by the lowest point. However, this method introduces a large ϵ_{new}, i.e., the new error term, which may be amplified through the following procedures and thus leads to imprecise certification results.

Comparatively, we propose a novel abstract transformer for the **tanh** function with a smaller ϵ_{new}, followed that we can obtain a more accurate approximation for the zonotope. Formally, our abstract transformer for the **tanh** function can be denoted as follows: given $y = tanh(x)$ where $x \in [l_x, u_x]$, based on Definition 4.2, the bounding planes of y are represented as $Z^L = C_1 + a_1 \cdot x$, and $Z^U = C_2 + a_1 \cdot x$, where

$$\begin{cases} a_1 = \dfrac{\tanh(u_x) - \tanh(l_x)}{u_x - l_x} \\ C_1 = \tanh(x^\star) - a_1 \cdot x^\star \\ C_2 = \tanh(x^{\star\star}) - a_1 \cdot x^{\star\star} \end{cases}, x^\star = \begin{cases} x', x' > l_x \\ l_x, x' \le l_x \end{cases}, x^{\star\star} = \begin{cases} x'', x'' < u_x \\ u_x, x'' \ge u_x \end{cases},$$

with x', x'' are two points of tangency and $x' < x''$. Based on the above formula, we show the graphic illustration of the proposed abstract transformer for the **tanh** function in Fig. 4, where the orange polygon represents our method. From Fig. 4, the slope of

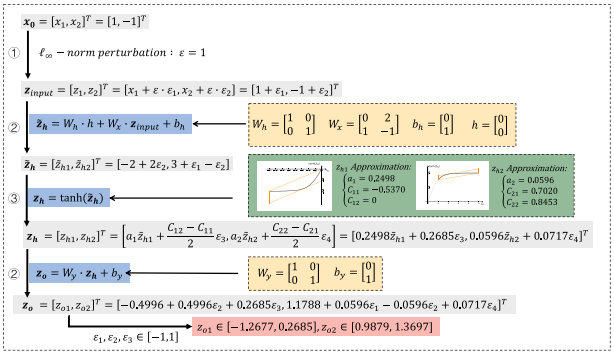


Figure 5: A toy example of how CERT-RNN certifying a vanilla RNN with 1 hidden layer and 2 hidden units. For simplicity, we omit the softmax layer, which just normalizes the output into a probability distribution and does not impact the final prediction results.

our two bounding lines are the tangent lines to **tanh**, whose slope is the same as that of the green dashed line, implying that our method for abstracting the **tanh** function is more accurate than *DeepZ*.

4.2.3 Robustness Verification. By analyzing the upper and lower bounds of an output zonotope, we can verify the robustness of a model for any input with a pre-defined ϵ based on the above approximation process. Taking a classification model as an example, if the lower bound of the correct label is larger than the upper bounds of all the other labels, its robustness with respect to this input is verified. It means that, the model is guaranteed to make consistent prediction for this input when the ℓ_p-norm perturbation is not larger than ϵ. The detailed verification algorithm will be given in Section 4.4.

4.2.4 A Toy Example. To better illustrate the workflow of CERT-RNN, we provide a toy example of how it works on a simple vanilla RNN model $\mathcal{F}$ which has 1 hidden layer and 2 hidden units as shown in Fig. 5. This model has two possible outputs y_1 and y_2, with the learned weights and biases shown in the yellow block in Fig. 5.

Suppose we have an input $\mathbf{X}_0 = [x_1, x_2]^T = [1, -1]^T$, and the model predicts $\mathbf{X}_0$ as label y_2, i.e., $\mathcal{F}(\mathbf{X}_0) = y_2$. Given the size of the ℓ_∞-norm perturbation as $\epsilon = 1$, we first map all possible adversarial inputs to the zonotope abstract domain as introduced in Section 4.2.1 and obtain an input zonotope $\mathbf{z}_{input}$. Then, we apply an affine transformation to $\mathbf{z}_{input}$ (shown in a blue block) and obtain a pre-activation zonotope $\tilde{\mathbf{z}}_h$, which does not lose any precision. After that, we apply the **tanh** activation function to $\tilde{\mathbf{z}}_h$, and then use the approximation method (shown in a green block) proposed in Section 4.2.2 to get a zonotope of the hidden state $\mathbf{z}_h$. Finally, we again apply an affine transformation to get the output zonotope of the entire network $\mathbf{z}_o$. Then, the confidence range of each label is obtained, i.e., the confidence value of y_1, y_2 lies in $[-1.2677, 0.2685]$ and $[0.9879, 1.3697]$, respectively. Since the lower bound of y_2 is larger than the upper bound of y_1, the robustness for $\mathbf{X}_0$ is verified. Further, we can leverage the algorithm in Section 4.4 to compute the bound for $\mathbf{X}_0$, which we do not detail here.

4.3 LSTM Certification

Based on the above procedure, we can certify the robustness bound of vanilla RNNs. However, the problem of vanishing/exploding gradient might occur in vanilla RNNs, thus an improved RNN – LSTM – is adopted more frequently in reality. Since LSTMs introduce different gates as shown in Fig. 2, they are much more complex than vanilla RNNs. In order to make CERT-RNN generally applicable, we certify the widely adopted LSTM below.

The whole certification process for LSTM is similar to that of vanilla RNNs (i.e., ①, ②, ③[1] in Fig. 2), except for two intermediate operations: the Hadamard product between a **sigmoid** function and a **tanh** function (④ in Fig. 2), and the Hadamard product between a **sigmoid** function and an **identity** function (⑤ in Fig. 2). In the following, we take the same design for input abstraction and robustness verification as that in the vanilla RNN certification, while focusing on studying how to abstract these intermediate operations.

4.3.1 Sigmoid ⊙ Tanh Abstract Transformer. Different from the abstraction of activation functions, which takes a one-dimensional zonotope as input, the abstraction of **sigmoid** ⊙ **tanh** receives a two-dimensional zonotope as input. A straightforward approach to handling such transforms is to approximate the zonotope to an interval and perform multiplication using intervals [22]. However, this approach would lose the relation information among perturbations and incur precision loss in further operations. Therefore, we propose a new abstract transformer, specifically tailored to handle the element-wise product in the update of the recurrent unit. To conveniently compute the upper and lower planes of the zonotope approximation, we first extend the two-dimensional zonotope to a rectangle encompassing the zontope.

Coarse-grained Abstract Transformer (CERT-RNN-Pre). We first propose a coarse-grained abstract transformer for the element-wise multiplication of **sigmoid** and **tanh**, leveraging which we can obtain an efficient coarse-grained abstract transformer for a zonotope, denoted by CERT-RNN-Pre.

THEOREM 4.1. *Let $x = \sigma(\cdot), y = tanh(\cdot)$ and $z = x \cdot y$, where $(x, y) \in \mathcal{Z} \subseteq [l_x, u_x] \times [l_y, u_y]$. Then, the coarse-grained zonotope approximation planes in $\mathcal{Z}$ are:*

$$Z^L = C_1 + Ax + By$$
$$Z^U = C_2 + Ax + By$$

where $A = (l_y + u_y)/2$, $B = (l_x + u_x)/2$, $C_1 = \min\{-(l_x \cdot l_y + u_x \cdot u_y)/2, -(u_x \cdot l_y + l_x \cdot u_y)/2\}$, and $C_2 = \max\{-(l_x \cdot l_y + u_x \cdot u_y)/2, -(u_x \cdot l_y + l_x \cdot u_y)/2\}$.

Due to the space limitation, the proof of Theorem 4.1 is deferred to Appendix A. From Theorem 4.1, we can efficiently obtain the coarse-grained upper and lower bounds for $f_{\sigma \cdot \tanh}(x, y) = \sigma(x)\tanh(y)$. However, as shown in Section 4.1, both the non-linear activation function abstraction and the multiplication operation abstraction yield new perturbations. Specifically, the amount of error will be tripled after an operation of **sigmoid** ⊙ **tanh** or doubled after

[1] As **sigmoid** is a rescaled **tanh** function, its zonotope approximation can be deducted in the same way as **tanh**.

an **sigmoid** $\odot$ **x** operation. Therefore, to eliminate the amplification of error, we propose the following fine-grained approximation method.

Fine-grained Abstract Transformer (Cert-RNN). In this abstract transformer, instead of separately computing the approximation of the multiplication operation as in Cert-RNN-Pre, we directly consider approximating the function of $f_{\sigma\cdot\tanh}(x,y) = \sigma(x)\tanh(y)$, as shown in Theorem 4.2, followed by obtaining a more accurate approximation.

Theorem 4.2. *Let $z = \sigma(x)\cdot\tanh(y)$, where $(x,y) \in \mathcal{Z} \subseteq [l_x, u_x]\times[l_y, u_y]$. Then, the fine-grained zonotope approximation planes in $\mathcal{Z}$ are:*

$$Z^L = C_1 + Ax + By$$
$$Z^U = C_2 + Ax + By$$

where A, B, C_1, C_2 have nine different cases as shown in Tab. 8 (deferred to Appendix B) according to the value of l_x, u_x, l_y and u_y.

Due to the space limitation, the proof of Theorem 4.2 is deferred to Appendix B. Based on Theorem 4.2, we can obtain more accurate upper and lower bounds for $f_{\sigma\cdot\tanh}(x,y) = \sigma(x)\tanh(y)$ than Cert-RNN-Pre.

4.3.2 Sigmoid $\odot$ Identity Abstract Transformer. For the function $f_{x\cdot\sigma}(x,y) = x\cdot\sigma(y)$, since $\sigma(y) \geq 0$, we consider three different cases, as shown in Theorem 4.3, according to the lower and upper bounds of x, denoted by l_x and u_x, respectively.

Theorem 4.3. *Let $z = x\cdot\sigma(y)$, where $(x,y) \in \mathcal{Z} \subseteq [l_x, u_x]\times[l_y, u_y]$. Then, the zonotope approximation planes in $\mathcal{Z}$ are:*

$$Z^L = C_1 + Ax + By$$
$$Z^U = C_2 + Ax + By$$

where A, B, C_1, C_2 have three different cases as shown in Tab. 9 (deferred to Appendix C) according to the value of l_x, u_x.

Due to the space limitation, the proof of Theorem 4.3 is deferred to Appendix C. Based on Theorem 4.3, we can efficiently obtain the accurate upper and lower bounds for $f_{x\cdot\sigma}(x,y) = x\cdot\sigma(y)$.

4.4 Certifying the Robustness Bound

Given a trained vanilla RNN or LSTM model $\mathcal{F}$, an input sequence $\mathbf{X}_0 \in \mathbb{R}^{T\times K}$, and the ℓ_∞-norm perturbation ϵ, now we can obtain an output zonotope $\mathbf{z}_o = \alpha_{i0} + \sum_{j=1}^{p}\alpha_{ij}\cdot\epsilon_j$, where $i \in \{1,\cdots,C\}$. Suppose the label of the input sequence is c. We aim at combining Cert-RNN and a binary search procedure to find the maximal robustness bound against any adversarial attack. Specifically, finding the largest robustness bound ϵ_c for the input sequence with true label c can be formalized as the following optimization problem:

$$\max \quad \epsilon_c$$
$$s.t. \quad \alpha_{0c} - \sum_{j=1}^{p}|\alpha_{jc}\cdot\epsilon_j| \geq \alpha_{0i} + \sum_{j=1}^{p}|\alpha_{ji}\cdot\epsilon_j|, \quad \forall i \neq c$$

To address the above optimization problem, we propose a binary search-based algorithm as shown in Alg. 1. According to Alg. 1, for the frame t in $\mathbf{X}_0$, we first initialize its bound as $\epsilon^{(t)} = 0.5$ (line 2), and then run Cert-RNN to verify the robustness of $\mathbf{X}_0$ under the temporary bound $\epsilon^{(t)}$ (line 4). If $\epsilon^{(t)}$ is verified, we then increase $\epsilon^{(t)}$ (line 6), else we decrease $\epsilon^{(t)}$ (line 8). For an input sequence $\mathbf{X}_0$ with T frames, we obtain the robustness bound for each frame and finally take the smallest $\epsilon^{(t)}$ as the largest possible robustness bound for $\mathbf{X}_0$ (line 9). With l starting from 2 to 13, we can obtain the robustness bound with precision of 0.0001.

Algorithm 1: Computing the robustness bound.

Result: Certified robustness bound ϵ_c
Data: model $\mathcal{F}$, input sequence $\mathbf{X}_0$, true label c
1 **for** t *in* T **do**
2 $\epsilon^{(t)} = 0.5$
3 **for** $l = 2$ *to* 13 **do**
4 $\mathbf{z}_o$ = Cert-RNN$(t, \mathcal{F}, \mathbf{X}_0, \epsilon^{(t)})$;
5 **if** $\alpha_{c0} - \sum_{j=1}^{p}|\alpha_{c_j}\cdot\epsilon_j| \geq \alpha_{i0} + \sum_{j=1}^{p}|\alpha_{ij}\cdot\epsilon_j|$ **then**
6 $\epsilon^{(t)} = \epsilon^{(t)} + 0.5^l$;
7 **else**
8 $\epsilon^{(t)} = \epsilon^{(t)} - 0.5^l$;
9 $\epsilon_c = min(\epsilon^{(1)}, \epsilon^{(2)}, \cdots, \epsilon^{(T)})$

5 EVALUATION OF CERT-RNN

In this section, we evaluate the performance of Cert-RNN on four tasks with sequential inputs and compare it with the state-of-the-art certification method.

5.1 Experimental Settings

5.1.1 Evaluation Scenarios. We consider four evaluation scenarios. (1) **Image Classification [47]**, aims to classify an image according to its visual content. It is important for many applications, e.g., autonomous driving, face recognition, etc. (2) **Sentiment Analysis [35]**, refers to identifying the sentiment orientation of the given text data. It also has many applications, e.g., analyzing ideological bias, monitoring online conversations, etc. (3) **Toxic Content Detection [46]**, aims to apply NLP, statistics, and/or machine learning methods to detect illegal or toxic-related (e.g., racism, pornography, terrorism, and riots) content for online systems. Toxic content detection is widly applied in many applications, including helping moderators improve the online conversation environment. (4) **Malicious URL Detection [54]**, aims to detect illegal websites that attempt to perform malicious behaviors, such as installing a malware onto a device, which is useful for various applications, including preventing users from viruses and hacking.

5.1.2 Datasets. We evaluate Cert-RNN and the state-of-the-art RNN robustness certification method POPQORN, which certifies RNNs by propagating linear bounds [25], on the following datasets corresponding to the above four scenarios, whose statistics (training, validation and testing) are shown in Tab. 1. (1) **MNIST sequence dataset.**[2] Different from the MNIST dataset, MNIST sequence records handwritten numbers as sequential data of line segment sequences. We use this dataset for the image classification evaluation. In our experiment, we split the original training

[2]https://edwin-de-jong.github.io/blog/mnist-sequence-data/

Table 1: Statistics of the four datasets.

Dataset	MNIST Sequence		Rotten Tomatoes			Toxic Comment Detection			Malicious URL Detection		
	# of Images	Size	Positive	Negative	Avg Length	Toxic	Normal	Avg Length	Malicious	Benign	Avg Length
Training	60,000	28×28	23,498	15,564	23 words	6,720	6,720	32 words	60,450	275,921	48 chars
Validation	/	/	3,362	1,562	23 words	1,280	1,280	32 words	7,567	34,479	48 chars
Testing	10,000	28×28	3,016	1,867	22 words	1,280	1,280	34 words	7,625	34,420	48 chars

Table 2: Evaluation results in the four scenarios, including model accuracy (Acc), mean value and standard deviation of the certified robustness bound (where a large mean implies a large robustness space), and running time.

Dataset	Model	Acc	POPQORN			Cert-RNN		
			Mean	Std	Time (min)	Mean	Std	Time (min)
MNIST Sequence	RNN-2-32	96.8%	0.0084	0.0037	0.13	0.0157	0.0077	0.61
	RNN-2-64	94.4%	0.0084	0.0033	0.12	0.0152	0.0076	0.63
	RNN-4-32	95.4%	0.0168	0.0058	0.30	0.0222	0.0074	1.72
	RNN-4-64	94.8%	0.0034	0.0018	0.40	0.0056	0.0032	1.70
	RNN-7-32	89.0%	0.0027	0.0016	0.64	0.0037	0.0025	4.01
	RNN-7-64	92.2%	0.0012	0.0012	0.60	0.0018	0.0012	4.21
	RNN-14-32	92.2%	0.0190	0.0064	1.44	0.0270	0.0075	13.44
	RNN-14-64	95.8%	0.0089	0.0030	2.31	0.0166	0.0044	14.38
	LSTM-1-32	98.0%	0.0152	0.0071	46.78	0.0187	0.0087	2.66
	LSTM-1-64	99.0%	0.0152	0.0064	53.09	0.0178	0.0075	4.92
	LSTM-1-128	98.0%	0.0143	0.0065	53.09	0.0184	0.0074	3.98
	LSTM-2-32	96.0%	0.0147	0.0062	150.00	0.0176	0.0080	8.42
	LSTM-2-64	98.0%	0.0145	0.0063	246.50	0.0167	0.0067	11.92
	LSTM-2-128	97.4%	0.0129	0.0052	192.77	0.0143	0.0056	12.77
	LSTM-4-32	95.0%	0.0093	0.0045	551.70	0.0095	0.0045	29.24
	LSTM-4-64	97.8%	0.0088	0.0040	593.31	0.0092	0.0039	37.13
	LSTM-7-32	96.6%	0.0054	0.0017	1522.77	0.0056	0.0015	90.99
RT	RNN	76.0%	0.0091	0.0049	1342.20	0.0207	0.0098	40.20
	LSTM	82.0%	-	-	-	0.0080	0.0026	2464.2
TC	RNN	90.0%	0.0190	0.0107	2070.60	0.0332	0.0243	98.40
	LSTM	93.0%	-	-	-	0.0117	0.0068	3903.60
MalURL	RNN	94.0%	0.0282	0.0132	2923.80	0.0361	0.0203	243.60
	LSTM	98.0%	-	-	-	0.0097	0.0044	9851.40

dataset into two parts, i.e., 50,000 samples for training set and 10,000 for validation set. (2) **Rotten Tomatoes Movie Review (RT) dataset.** This dataset is a benchmark corpus of movie reviews used for sentiment analysis, originally collected by Pang and Lee [34]. (3) **Toxic Comment (TC) dataset.** This dataset is provided by Kaggle[3]. Specifically, we consider six categories of toxicity (i.e., "toxic", "severe toxic", "obscene", "threat", "insult", and "identity hate") as toxic and perform binary classification in the evaluation. For more coherent comparisons, a balanced subset of this dataset is constructed by random sampling for evaluation. (4) **Malicious URL (MalURL) dataset.** This dataset is provided by Kaggle[4]. We use this dataset for the malicious URL detection evaluation.

5.1.3 Models. For MNIST sequence, we trained 8 vanilla RNNs and 9 LSTMs as listed in Tab. 2. The models are listed in the "network-layer-hidden units" format, e.g., LSTM-1-32 represents the LSTM with one layer and 32 hidden units. For the other three datasets, we each trained a vanilla RNN with 32 hidden units and an LSTM with 32 hidden units, respectively. All models were trained in a hold-out test strategy, whose accuracy is shown in Tab. 2.

5.1.4 Implementation Details. For MNIST sequence, we normalize the range of each pixel from $[0, 255]$ to $[0, 1]$ to be consistent with POPQORN [25]. For RT and TC, we evaluate on their word embeddings due to the extremely discrete property of the word space (theoretically, infinite). Specifically, we use the pretrained word embeddings from "glove.6B.100d"[5], and for the out-of-vocabulary words, we initialized them by randomly sampling from the uniform distribution in [-0.1, 0.1]. For MalURL, since the character level embedding can generalize to new URLs easily compared with the word level embedding (i.e., even if the given URL contains unseen words, the character level embedding can still represent these new words), we use a character level embedding [40] which contains 144 characters.

[3]https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
[4]https://www.kaggle.com/antonyj453/urldataset
[5]https://nlp.stanford.edu/projects/glove/

Table 3: Mann-Whitney U test results.

Model	RNN-2-32	RNN-4-32	RNN-7-32	RNN-14-32
p-value	6.93×10^{-9}	1.91×10^{-22}	2.10×10^{-29}	1.11×10^{-30}
Model	RNN-2-64	RNN-4-64	RNN-7-64	RNN-14-64
p-value	1.12×10^{-20}	1.83×10^{-12}	4.81×10^{-7}	2.76×10^{-12}

For adversarial attacks, adversaries usually try to perturb as less words/pixels as possible to be human imperceptible and meanwhile preserve more utility [29]. Therefore, we evaluate the robustness of perturbing one single frame instead of all frames (i.e., we fix all input frames but one and derive the certified bound for perturbations on that frame). For each sample, after calculating the robustness bounds for all frames, the minimal one is identified as the final bound of this sample.

In all experiments, we randomly select 1,000 correctly classified examples from the testing set to conduct the certification evaluation. We repeated each experiment 5 times and report the mean value. This replication is important because training is stochastic and thus may introduce variance in performance [53]. All experiments are conducted on a server with two Intel Xeon E5-2640 v4 CPUs running at 2.40GHz, 64 GB memory, 4TB HDD and a GeForce GTX 1080 Ti GPU card.

5.2 Results and Analysis

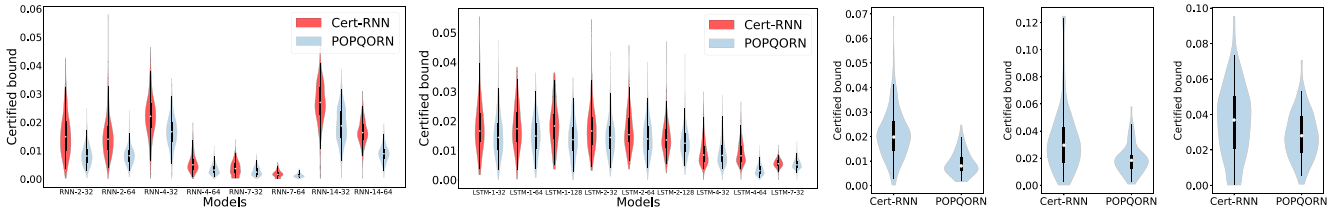
When reporting the results, we refer to the following quantities: (i) the **certified robustness bound** of a particular sample $\boldsymbol{x}$ is the maximum ϵ for which we can certify that the model $f(\boldsymbol{x}')$ will return the correct label, where $\boldsymbol{x}'$ is any adversarially perturbed version of $\boldsymbol{x}$ such that $||\boldsymbol{x} - \boldsymbol{x}'||_\infty \leq \epsilon$; (ii) the **verified accuracy** at ϵ of a dataset is the fraction of data items in the dataset with certified robustness bound of at least ϵ.

Certified Robustness Bound. The evaluation results in the four scenarios are shown in Tab. 2 and Fig. 6. From the results, we have the following observations.

- In all cases, Cert-RNN can obtain larger robustness bounds than that of POPQORN, i.e., the result of Cert-RNN is more accurate. For instance, in Tab. 2 and Fig. 6, for the RNN-2-32 model on MNIST sequence, the robustness bound of Cert-RNN is 1.86 times of that of POPQORN. Since an example in this model has 2 frames, the input space is $\frac{28^2}{2} = 392$-dimensional. Hence, the volume of the Cert-RNN robustness bound is 1.86^{392} times of that of POPQORN. The outstanding performance of Cert-RNN is mainly because the abstract interpretation used by Cert-RNN takes into account the inter-variable correlation while the interval arithmetic used by POPQORN does not. Thus, Cert-RNN's certification can be conducted at a fine-grained scale, followed by obtaining a more accurate robustness bound.
- We observe that when the number of hidden units is the same, LSTMs with less layers would be more robust. For instance, as shown in Tab. 2 and Fig. 6, LSTM-1-32, LSTM-2-32, LSTM-4-32, and LSTM-7-32 on MNIST sequence have the robustness bounds of 0.0187, 0.0176, 0.0095, and 0.0056, respectively. Meanwhile, when the number of layers is same, LSTMs with less hidden units would be more robust. For instance, LSTM-2-32, LSTM-2-64, and LSTM-2-128 on MNIST sequence have the robustness bounds of 0.0176, 0.0167, and 0.0143, respectively. This finding is also true for vanilla RNNs. Since our models are all in the "classical" regime [6], we speculate the reason is that too many hidden units may increase the attack surface and decrease the generalizability (i.e., have a high variance) of the model, which makes it less robust. This indicates that more complex models in the 'classical' regime are not more robust in reality, which is helpful for practitioners when building robust intelligent systems.
- As shown in Tab. 2 and Fig. 6, the certified robustness bound of vanilla RNN is larger than that of LSTM on the same dataset. For instance, on the RT dataset, the certified robustness bounds of vanilla RNN and LSTM are 0.0207 and 0.0080, respectively. Note that, for variable-length inputs, the number of layers of RNNs is determined by the input length. Therefore, we speculate the reason is that the approximation error for the multiplication operation is amplified through each layer of LSTM. Since vanilla RNN does not have the multiplication operation, it would be slightly affected.
- We conduct preliminary Mann-Whitney U test for 8 vanilla RNNs on MNIST sequence, and the results are shown in Tab. 3. For the Mann-Whitney U test, we define the results of POPQORN belong to the population X and the results of Cert-RNN belong to the population Y. The null hypothesis is the probability of X being greater than Y is equal to the probability of Y being greater than X. From Tab. 3, we can see that the p-values of all models are small enough to reject the null hypothesis, which demonstrates the superiority of Cert-RNN.

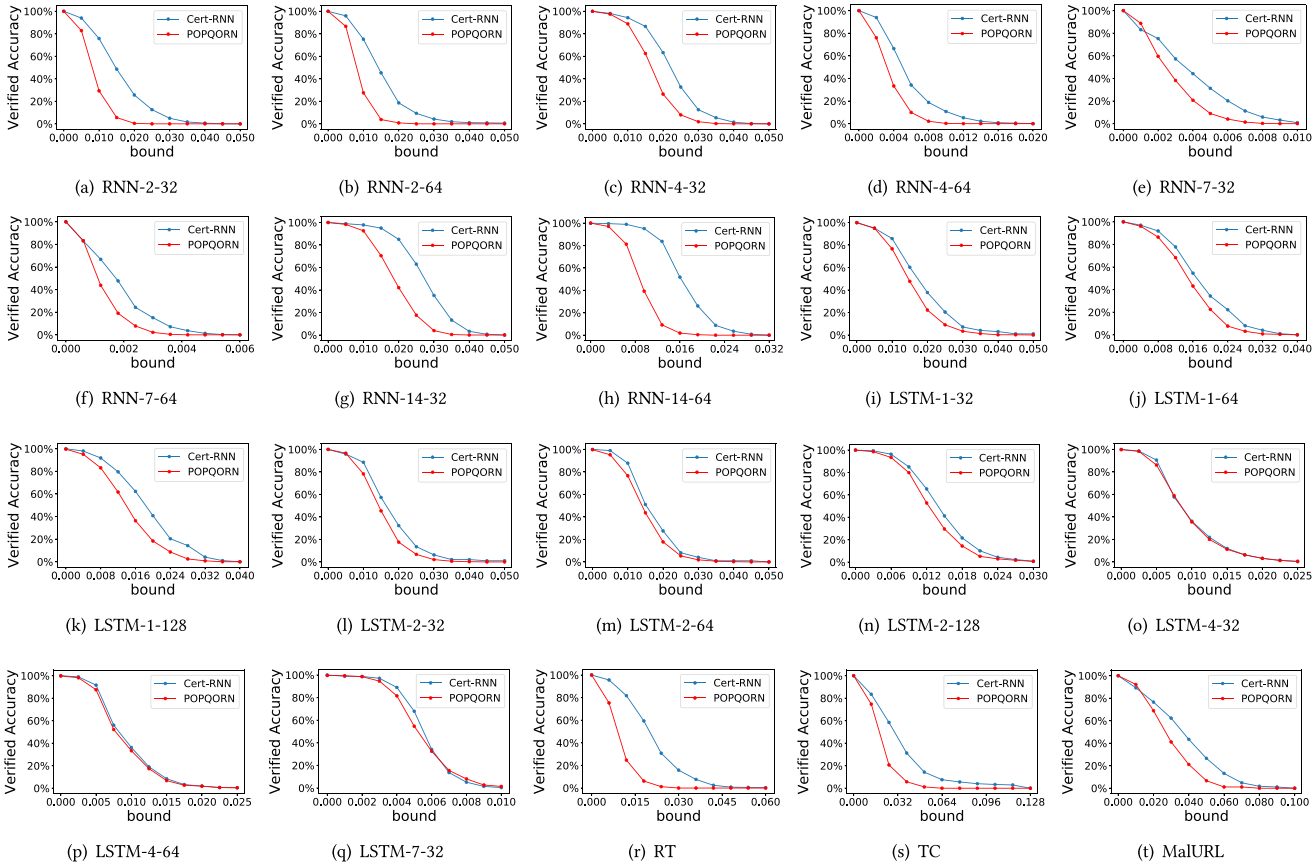
Efficiency. The running time results on vanilla RNNs and LSTMs of the four scenarios are shown in Tab. 2. Specifically, we can observe the following from the results.

- Cert-RNN is much more efficient than POPQORN in general, especially for large and complex networks. For instance, for LSTM-1-32, the running time of POPQORN is 46.78 minutes on average while Cert-RNN only consumes 2.66 minutes on average. In addition, for larger LSTMs such as LSTM-7-32, the running time of POPQORN increases significantly, i.e., taking about 25 hours, while Cert-RNN remains efficient, which takes about 1.5 hours. This is because the optimization approach of POPQORN needs training a new model to approximate the bounding planes, which is quite time-consuming. By contrast, Cert-RNN, which approximates the bounding planes based on efficient abstract transformers in Section 4, does not need to train such a model.
- Cert-RNN is also more efficient than POPQORN on RT, TC, and MalURL for vanilla RNNs and LSTMs. For instance, for vanilla RNN on RT, the running time of Cert-RNN and POPQORN are 0.67 hours and 22.37 hours, respectively. In addition, when evaluating POPQORN on RT, TC, and MalURL, we observe that POPQORN takes more than 24 hours for certifying LSTM even only for a single word. This further indicates that POPQORN is difficult to be scaled to large LSTMs. In contrast, Cert-RNN can be extend to these networks. Therefore, considering the limited computing resources, we do not extensively evaluate POPQORN for LSTMs in some scenarios that are extremely expensive for it.
- From Tab. 2, we can also see that both Cert-RNN and POPQORN can efficiently certify the robustness bound of vanilla RNNs on MNIST sequence, typically within a few minutes. Note that, for some simple vanilla RNN networks, POPQORN can even be a little faster than Cert-RNN. This is mainly because POPQORN



(a) vanilla RNNs on MNIST sequence (b) LSTMs on MNIST sequence (c) RT (d) TC (e) MalURL

Figure 6: Certified robustness bound in the four scenarios. The violin plot shows the data distribution shape and its probability density, which combines the features of box and density charts. The thick black bar in the middle indicates the quartile range, the thin black line extending from it represents the 95% confidence interval, and the white point is the median.



(a) RNN-2-32 (b) RNN-2-64 (c) RNN-4-32 (d) RNN-4-64 (e) RNN-7-32

(f) RNN-7-64 (g) RNN-14-32 (h) RNN-14-64 (i) LSTM-1-32 (j) LSTM-1-64

(k) LSTM-1-128 (l) LSTM-2-32 (m) LSTM-2-64 (n) LSTM-2-128 (o) LSTM-4-32

(p) LSTM-4-64 (q) LSTM-7-32 (r) RT (s) TC (t) MalURL

Figure 7: Verified accuracy of four datasets for each bound $\epsilon \in \Delta$(the x axis). The subfigures (a) to (q) are the results of the MNIST sequence dataset.

uses parallization implementation while Cert-RNN does not. Even though, Cert-RNN is more efficient than POPQORN in most cases, especially for large and complicated networks. This further demonstrates the efficient design of Cert-RNN.

Verified Accuracy. The verified accuracy of the four datasets versus $\epsilon \in \Delta$ (the x axis) is shown in Fig. 7, from which, we have the following observations.

- For both vanilla RNNs and LSTMs, the verified accuracy of Cert-RNN is much higher than that of POPQORN in most cases. For instance, for RNN-2-32 on MNIST sequence as shown in Fig. 7(a), when the robustness bound is 0.01, Cert-RNN can verify about 75% samples while POPQORN can only verify 25% samples. This demonstrates that Cert-RNN can verify RNNs with much wider robustness regions than POPQORN.
- For LSTM-4-32 and LSTM-4-64 as shown in Figs. 7(o) and 7(p), though both Cert-RNN and POPQORN verify roughly the same accuracy, their running time differs a lot as shown in Tab. 2: Cert-RNN is almost 19 times faster than POPQORN for LSTM-4-32 and 16 times faster for LSTM-4-64. Therefore, Cert-RNN is more promising in practical applications than POPQORN.

Table 4: Results for Cert-RNN-Pre on MNIST sequence.

Model	Acc	Cert-RNN-Pre		
		Mean	Std	Time (min)
LSTM-1-32	98.0%	0.0132	0.0072	0.90
LSTM-1-64	99.0%	0.0134	0.0070	0.98
LSTM-1-128	98.0%	0.0113	0.0071	1.00
LSTM-2-32	96.0%	0.0132	0.0056	2.47
LSTM-2-64	98.0%	0.0127	0.0056	3.16
LSTM-2-128	97.4%	0.0114	0.0045	2.50
LSTM-4-32	95.0%	0.0071	0.0038	7.61
LSTM-4-64	97.8%	0.0061	0.0032	9.61
LSTM-7-32	96.6%	0.0031	0.0021	20.82

Table 5: Model accuracy for the defended networks.

Dataset	FGSM-AT	PGD-AT	IBP-VT	Cert-RNN-VT
MNIST sequence	98.0%	98.0%	98.0%	-
RT	80.0%	79.0%	80.0%	79.0%
TC	87.0%	91.0%	91.0%	90.0%
MalURL	94.0%	93.0%	92.0%	91.0%

Table 6: Results for perturbing all frames on the MNIST sequence dataset.

	Cert-RNN		
	Mean	Std	Time (sec)
RNN-2-32	0.0126	0.0055	6.8420
RNN-2-64	0.0130	0.0056	9.0874
RNN-4-32	0.0044	0.0044	0.0044
RNN-4-64	0.0047	0.0023	14.3441
RNN-7-32	0.0044	0.0044	20.5882
RNN-7-64	0.0017	0.0009	15.4963
RNN-14-32	0.0127	0.0036	31.8162
RNN-14-64	0.0074	0.0020	34.0596

- From Fig. 7, we can also see that the value of the verified accuracy converges to zero as the robustness bound increases in all cases. This is expected, as larger input regions are more likely to contain more adversarial examples.

Cert-RNN vs Cert-RNN-Pre. Taking LSTMs on MNIST sequence for example, we further compare the performance of Cert-RNN and Cert-RNN-Pre, as shown in Tab. 4. Comparing Tab. 2 and Tab. 4, though Cert-RNN-Pre is more efficient than Cert-RNN (as expected, since Cert-RNN-Pre directly uses the interval range of two functions to approximate their multiplication), its certified robustness bound is smaller than Cert-RNN. This is because the amount of perturbation error will be tripled after an operation of **sigmoid** $\odot$ **tanh** or doubled after a **sigmoid** $\odot$ **x** operation in Cert-RNN-Pre. In contrast, directly approximating the function of $f_{\sigma\cdot\tanh}(x, y) = \sigma(x)\tanh(y)$, which is exactly Cert-RNN does, can mitigate the amplification of perturbation error. This indicates that Cert-RNN is more promising than Cert-RNN-Pre when being applied to security-sensitive applications, where the robustness bound should be certified as accurate as possible.

Perturbing All Frames. We conduct preliminary experiments for perturbing all frames for some models on the MNIST sequence dataset, and the results are shown in Tab. 6. Since POPQORN cannot handle this threat model, we only conduct expertiments for Cert-RNN. From Tab. 6, we can see that comparing with the threat model that only perturbing one frame, the robustness bounds for perturbing all frames decrease to some extent. This is reasonable since perturbing all frames enlarges the attack space for attackers.

Summary. In summary, from the above results and analysis, we can see that Cert-RNN outperforms POPQORN in the following aspects: (1) *accurate* – Cert-RNN can certify much tighter robustness bounds than POPQORN in all cases; (2) *efficient* – Cert-RNN is more efficient than POPQORN, especially for large and complicated networks; and (3) *scalable* – Cert-RNN can scale to larger models which are beyond the reach of POPQORN. These properties make Cert-RNN more promising in practical applications.

6 APPLICATIONS

The certified robustness bound has many important applications, e.g., certifying the effectiveness of different defenses [14], incorperated in the robust training procedure to design a provably robust defense [18, 33], and identifying sensitive words [25, 42]. In this section, we apply the certified robustness bound in the above three applications to further demonstrate its reasonability and benefits.

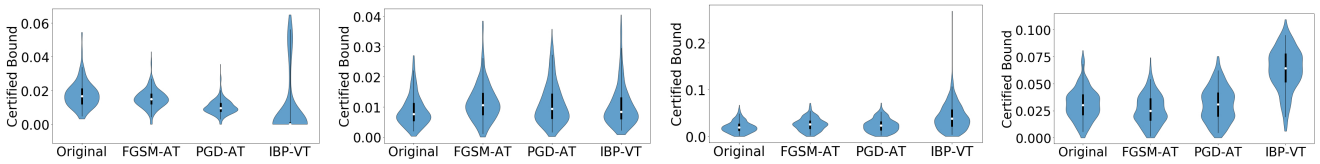
6.1 Certifying Adversarial Defenses

In this subsection, we demonstrate a practical application of Cert-RNN: certifying different adversarial defenses, which can help users build more robust intelligent systems.

Defense Methods. We trained the RNN being protected with each of the following defenses according to their published code, where each model's accuracy is shown in Tab. 5.

- **FGSM-AT** (Fast Gradient Sign Method-based Adversarial Training) [17]. FGSM is an adversarial attack that generates adversarial examples by increasing the loss of the model on input $\boldsymbol{X}$ as: $\boldsymbol{X}_{adv} = \boldsymbol{X} + \epsilon sign(\nabla_X J(\boldsymbol{X}, y_{true}))$, where ϵ represents the noise scale and $J(\cdot)$ represents the loss function (e.g., cross-entropy). FGSM-AT extends the loss funtion of the model to be protected with a regularization term encoding the FGSM attack.
- **PGD-AT** (Projected Gradient Descent-based Adversarial Training) [32]. PGD is an extension of FGSM that applies it multiple times with a small step size of perturbation and random starts. PGD-AT is designed to adversarially train a classifier using the PGD attack. Specifically, in each iteration, PGD is applied to generate a minibatch of adversarial samples to update the network.
- **IBP-VT** (Interval Bound Propagation-based Verified Training) [18] leverages interval bound propagation (IBP) to train provably robust models, which is shown outperforming the state-of-the-art in verified accuracy. The IBP technique is derived from interval arithmetic, which allows to define a loss to minimize the upper bound of the maximum difference between any pair of logits when the input is perturbed within an ℓ_∞ norm-bounded ball.

Implementation Details. In this experiment, we evaluate Cert-RNN on the LSTM-2-32 model for MNIST sequence due to its great robustness as shown in Fig. 6(b). In addition, we evaluate Cert-RNN on the vanilla RNN with 32 hidden units for RT, TC, and MalURL for consistency with Section 5. For comparison, each model will be trained with no defense and with FGSM-AT, PGD-AT, and IBP-VT, respectively. For measurement, we randomly select 500 examples



(a) MNIST sequence (b) Sentiment Analysis (c) Toxic Comment (d) Malicious URL

Figure 8: Violin plot of the certified bounds for the Original, and FGSM-AT, PGD-AT, and IBP-VT defended networks. The violin plot shows the data distribution shape and its probability density, which combines the features of box and density charts. The thick black bar in the middle indicates the quartile range, the thin black line extending from it represents the 95% confidence interval, and the white point is the median.

from each testing dataset to conduct the robustness certification. Other implementation details are the same with that in Section 5.

Results and Analysis. The evaluation results are shown in Fig. 8, from which we have the following observations.

- We observe that FGSM-AT and PGD-AT defend the network in a way that makes it only slightly more provably robust than the original RNN. For instance, on TC, the certified robustness bound of the original model is 0.0205, while the certified robustness bounds of FGSM-AT and PGD-AT defended models are 0.0258 and 0.0238, respectively. This finding is consistent with the empirical observations in [4, 19] that these defenses are insufficient.
- The certified bound for the IBP-VT defended model is larger than that of the original, FGSM-AT and PGD-AT defended models. For instance, on TC, the certified robustness bound of the IBP-VT defended model is 0.0416, which is 2.03 times of the original model, 1.61 times of the FGSM-AT defended model and 1.75 times of the PGD-AT defended model, respectively. This indicates that IBP-VT can provide a significant increase in provable robustness in this application scenario, and thus is superior to the other two defenses. We speculate the reason is that IBP-VT optimizes the worst-case adversarial loss with an adaptive regularizer that encourages robustness against all attacks, while heuristic defenses like FGSM-AT and PGD-AT are insufficient to ensure security.

In this application, we demonstrate that Cert-RNN can provide an accurate qualitative metric to evaluate the provable effectiveness of various defenses, which would be more reliable than previous empirical metrics, e.g., the attack success rate after applying a defense method. Therefore, we believe Cert-RNN is helpful to build more robust intelligent systems.

6.2 Improving RNN Robustness

Verified robust training [18] provides a general, principled mechanism to eliminate blind spots of adversarial examples by encouraging models to make correct predictions on all inputs within certain pre-defined adversarial regions. In this subsection, we demonstrate the application of incorporating Cert-RNN in verified robust training of RNNs (e.g., minimizing the upper bound on the worst-case loss) to improve the robustness of RNNs.

Implementation Details. In this experiment, we incorporating Cert-RNN in verified robust traininng of the vanilla RNN with 32 hidden units for RT, TC, and MalURL for consistency with Section 5. Our training follows [18, 33] – we perturb the input signal and propagate interval bounds obtained by Cert-RNN through the RNN stages. To train, we combine standard loss with the worst case loss obtained using interval propagation. For comparison, each model will also be trained with IBP-VT according to its published code. The trained models' accuracy are shown in Tab. 5, where Cert-RNN-VT denotes the models robustly trained with Cert-RNN. For measurement, we randomly select 500 examples from each testing dataset to conduct the robustness certification. Other implementation details are the same with that in Section 5.

Table 7: Certified robustness bounds for verified robustly trained RNNs.

Dataset	Original	IBP-VT	Cert-RNN-VT
RT	0.0207	0.0219	0.0224
TC	0.0332	0.0428	0.0436
MalURL	0.0361	0.0702	0.0730

Results and Analysis. The experimental results are shown in Tab. 7, from which we can see that the RNNs trained with Cert-RNN-VT achieve larger robustness bounds, outperforming the RNNs trained with IBP-VT on all three datasets. For instance, for MalURL, the RNN trained with Cert-RNN-VT achieves 0.0730 robustness bound, while the RNN trained with IBP-VT achieves 0.0702 robustness bound. This is because the interval bounds obtained by our approximation of the tanh function is more accurate than that obtained by the IBP method. Therefore, we believe Cert-RNN is helpful in improving the robustness of RNNs.

6.3 Identifying Sensitive Words

Identifying sensitive words is meaningful in many text-based machine learning tasks, including explaining the prediction of models, assisting sensitivity analysis, etc. In this subsection, we demonstrate the application of Cert-RNN in identifying sensitive words.

Experimental Settings. We demonstrate the application on the TC dataset as shown in Section 5.1.2, and the evaluated example model is the LSTM with 32 hidden units. The experimental setting and implementation details are the same as that in Section 5.1.1.

Results and Analysis. Due to the lacking of the sensitivity label of each word, it is difficult to measure the experiment results using general metrics like accuracy and precision. Therefore, we conduct a manual analysis on the results. By inspecting the "normal" (i.e., non-toxic) examples, we found there are no significant words and they seldom provide useful information for measuring our system. This is as expected considering that our goal here is to detect toxic examples. Therefore, to better demonstrate this application, we give five representative examples from the results which are correctly classified as "toxic" in Fig. 9, where the words with smaller

Example	**this**	is	a	**stupid**	**idea**	all	it	is	doing	is	adding	junk	to	an	already	good	page
Bound	0.0183	0.0188	0.0222	0.0178	0.0183	0.0232	0.0315	0.0320	0.0315	0.0334	0.0320	0.0334	0.0398	0.0427	0.0457	0.0496	0.0564
Example	you	are	an	**idiot**	nothing	suggests	that	she	needs	to	attend	a	hearing				
Bound	0.0178	0.0173	0.0188	0.0149	0.0188	0.0193	0.0212	0.0247	0.0247	0.0305	0.0305	0.0305	0.0217				
Example	hi	,	**idiot**	**,**	why	are	you	delate	my	talking	,	just	come	out			
Bound	0.0134	0.0110	0.0071	0.0085	0.0115	0.0139	0.0183	0.0154	0.0208	0.0159	0.0193	0.0232	0.0256	0.0291			
Example	oh	yeah	,	you	're	really	proof	of	the	**hypocrisy**	of	wikipedia	right	here			
Bound	0.0090	0.0095	0.0110	0.0120	0.0129	0.0129	0.0090	0.0110	0.0129	0.0085	0.0120	0.0153	0.0193	0.0242			
Example	you	must	be	a	real	**loser**	and	**mental**	infant	to	try	to	block	me			
Bound	0.0105	0.0095	0.0125	0.0144	0.0105	0.0081	0.0134	0.0081	0.0139	0.0183	0.0198	0.0212	0.0247	0.0237			

Figure 9: Five examples in the toxic comment detection task. The upper row gives the sample sentence where the most sensitive words (words with smallest bounds) are underlined. The lower row shows the CERT-RNN certified robustness bound (ℓ_∞-norm) of each individual word.

certified robustness bounds tend to be more important for the final prediction result, i.e., more sensitive. From Fig. 9, we can see that the most sensitive words identified by CERT-RNN are indeed more closely tied to the category of each sentence. For instance, in the first example, **stupid** is shortlisted in the top-3 most sensitive words, which is consistent with human cognition. Such observed consistency demonstrates CERT-RNN's potential in distinguishing the importance of different words consistently with their sentiment polarities, which is very helpful for explaining the prediction of RNNs. Thus, the robustness bound certified by CERT-RNN can be used as a meaningful quantitative metric for improving the interpretability of RNNs.

7 LIMITATION AND DISCUSSION

Improving Zonotope Approximations. In this paper, we formalize the zonotope approximation problem as finding the smallest coefficient of the new error term as defined in Section 4.1. In fact, the best zonotope approximation should be the one that generates the output zonotope with the smallest range. Hence, the approximations defined in Sections 4.2 and 4.3 may not be the tightest one under certain circumstances. It is interesting to explore alternative zonotope approximations which lead to tighter robustness bounds or discover a better algorithm for the best zonotope approximation.

Supporting Other Norm-Bounded Attacks. While abstract interpretation is immediately applicable to ℓ_∞, it can also be used to approximate other norms (e.g., ℓ_2). Intuitively, because ℓ_∞ allows the most flexible perturbations, the perturbations bounded by other norms can be considered as the subsets of those allowed by the ℓ_∞ bound. We consider supporting other norm-bounded attacks as our ongoing research. Therefore, if CERT-RNN can certify the non-existence of adversarial examples for an RNN within the ℓ_∞ norm bound, the RNN is also guaranteed to be safe for the ℓ_p-norm ($p = 1, 2, \ldots$) bound. If CERT-RNN identifies an adversarial region for the ℓ_∞ norm bound, we can iteratively check whether any such region lies within the ℓ_p-norm bound. If not, we can declare the model to contain no adversarial examples for the given ℓ_p-norm bound. We plan to explore this direction in the future.

Supporting More RNN Types. Following another track, we can investigate to extend CERT-RNN to support more RNN types, such as gated recurrent unit (GRU) networks and attention-based RNNs. Meanwhile, it is an open question of whether CERT-RNN can give non-trivial bounds for sequence-to-sequence tasks like machine translation [9]. We believe these extensions would further improve CERT-RNN's applicability.

Supporting Other Threat Models. In our experiments, we certify the robustness bounds of RNNs under the threat model in which attackers can directly perturb the word embeddings. This worst-case setting considers the strongest adversary. It is possible to consider other adversarial scenarios, e.g., the word substitution perturbation attack [2]. We believe this extension would further improve CERT-RNN's practicality.

8 CONCLUSION

In this paper, we present the design, implementation, and evaluation of CERT-RNN, a robustness certification framework for RNNs. At a high level, CERT-RNN abstracts the non-linear operations unique to RNNs within the framework of abstract interpretation and enables flexible trade-off between certification precision and execution scalability. Through extensive evaluation across different applications, we demonstrate that CERT-RNN is able to provide tight robustness bounds for RNNs and outperforms the state-of-the-art methods in this space by a large margin in terms of both precision and scalability. This work represents a solid step towards ensuring the robustness of RNNs and AI systems in general, leading to a few promising directions for further research.

9 ACKNOWLEDGMENTS

This work was partly supported by the National Key Research and Development Program of China under No. 2020AAA0140004 and 2020YFB2103802, NSFC under No. 61772466, U1936215, and U1836202, and the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars under No. LR19F020003. Ting Wang is partially supported by the National Science Foundation under Grant No. 1953893, 1953813, and 1951729.

REFERENCES

[1] Michael E Akintunde, Andreea Kevorchian, Alessio Lomuscio, and Edoardo Pirovano. 2019. Verification of rnn-based neural agent-environment systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6006–6013.

[2] Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating Natural Language Adversarial Examples. In *EMNLP*. 2890–2896.

[3] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*. 173–182.

[4] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *International Conference on Machine Learning*. 274–283.

[5] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *ICASSP*. IEEE, 4945–4949.

[6] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. 2018. Reconciling modern machine learning practice and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118* (2018).

[7] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[8] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. 39–57.

[9] Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. 2020. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *AAAI*.

[10] Patrick Cousot and Radhia Cousot. 1977. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*. ACM, 238–252.

[11] Tianyu Du, Shouling Ji, Jinfeng Li, Qinchen Gu, Ting Wang, and Raheem Beyah. 2020. SirenAttack: Generating Adversarial Audio for End-to-End Acoustic Systems. In *ASIACCS*.

[12] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In *ACL*. 31–36.

[13] Ruediger Ehlers. 2017. Formal verification of piece-wise linear feed-forward neural networks. In *ATVA*. Springer, 269–286.

[14] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. 2018. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.

[15] Khalil Ghorbal, Eric Goubault, and Sylvie Putot. 2009. The zonotope abstract domain taylor1+. In *International Conference on Computer Aided Verification*. Springer, 627–633.

[16] Yotam Gil, Yoav Chai, Or Gorodissky, and Jonathan Berant. 2019. White-to-Black: Efficient Distillation of Black-Box Adversarial Attacks. In *Proceedings of NAACL-HLT*. 1373–1379.

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

[18] Sven Gowal, Krishnamurthy Dj Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. 2019. Scalable Verified Training for Provably Robust Image Classification. In *Proceedings of the IEEE International Conference on Computer Vision*. 4842–4851.

[19] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. 2017. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. 2017. Safety verification of deep neural networks. In *ICCAV*. Springer, 3–29.

[22] Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *EMNLP*. 2021–2031.

[23] 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 (EMNLP-IJCNLP)*. 4120–4133.

[24] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*. Springer, 97–117.

[25] Ching-Yun Ko, Zhaoyang Lyu, Lily Weng, Luca Daniel, Ngai Wong, and Dahua Lin. 2019. POPQORN: Quantifying Robustness of Recurrent Neural Networks. In *International Conference on Machine Learning*. 3468–3477.

[26] Ji Young Lee and Franck Dernoncourt. 2016. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 515–520.

[27] Jinfeng Li, Tianyu Du, Shouling Ji, Rong Zhang, Quan Lu, Min Yang, and Ting Wang. 2020. Textshield: Robust text classification based on multimodal embedding and neural machine translation. In *USENIX Security 20*. 1381–1398.

[28] Jinfeng Li, Tianyu Du, Xiangyu Liu, Rong Zhang, Hui Xue, and Shouling Ji. 2021. Enhancing Model Robustness by Incorporating Adversarial Knowledge into Semantic Representation. In *ICASSP*. IEEE, 7708–7712.

[29] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating Adversarial Text Against Real-world Applications. *NDSS*.

[30] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. 2018. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1778–1787.

[31] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 2873–2879.

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*.

[33] Matthew Mirman, Timon Gehr, and Martin Vechev. 2018. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*. 3575–3583.

[34] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*. Association for Computational Linguistics, 115–124.

[35] Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2, 1–2 (2008), 1–135.

[36] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 582–597.

[37] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. How to construct deep recurrent neural networks: Proceedings of the Second International Conference on Learning Representations (ICLR 2014). In *2nd International Conference on Learning Representations, ICLR 2014*.

[38] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. 2019. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. In *International Conference on Machine Learning*. 5231–5240.

[39] Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *ACL*. 1085–1097.

[40] Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *ICML*. 1818–1826.

[41] Karsten Scheibler, Leonore Winterer, Ralf Wimmer, and Bernd Becker. 2015. Towards Verification of Artificial Neural Networks.. In *MBMV*. 30–40.

[42] Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. 2019. Robustness Verification for Transformers. In *ICLR*.

[43] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. 2018. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems*. 10802–10813.

[44] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. 2019. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages* 3, POPL (2019), 41.

[45] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)* (2014).

[46] Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. 2018. Challenges for Toxic Comment Classification: An In-Depth Error Analysis. In *ALW2*. 33–42.

[47] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. 2017. Residual attention network for image classification. In *CVPR*. 3156–3164.

[48] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. Cnn-rnn: A unified framework for multi-label image classification. In *CVPR*. 2285–2294.

[49] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. 2018. Towards Fast Computation of Certified Robustness for ReLU Networks. In *ICML*. 5273–5282.

[50] Eric Wong and J Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*. 5283–5292.

[51] Hiromu Yakura and Jun Sakuma. 2019. Robust audio adversarial example for a physical attack. In *IJCAI*. 5334–5341.

[52] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. 2018. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*. 4939–4948.

[53] Ye Zhang and Byron Wallace. 2017. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In *IJCNLP*, Vol. 1. 253–263.

[54] Peilin Zhao and Steven CH Hoi. 2013. Cost-sensitive online active learning with application to malicious URL detection. In *KDD*. 919–927.

Appendices

A PROOF OF THEOREM 4.1

Proof. We enumerate nine possible cases for the bounding plane.

Case 1: $l_x \le B \le u_x$ **and** $l_y \le A \le u_y$. By the definition of the bounding plane, the following conditions must be met:

$$\begin{aligned}&\min\{(xy - Ax - By - C_1)\}\\ =&\min\{(y-A)(x-B)\} - AB - C_1 \ge 0 \qquad (2)\\ &\max\{(xy - Ax - By - C_2)\}\\ =&\max\{(y-A)(x-B)\} - AB - C_2 \le 0 \qquad (3)\end{aligned}$$

The minimum of $(y-A)(x-B)$ is reached when $x = l_x, y = u_y$ or $x = u_x, y = l_y$, and the maximum of $(y-A)(x-B)$ is reached when $x = u_x, y = u_y$ or $x = l_x, y = l_y$. Then, we have:

$$\min\{(u_y - A)(l_x - B), (l_y - A)(u_x - B)\} - AB - C_1 \ge 0 \qquad (4)$$
$$\max\{(u_y - A)(u_x - B), (l_y - A)(l_x - B)\} - AB - C_2 \le 0 \qquad (5)$$

Suppose $(u_y - A)(l_x - B) \le (l_y - A)(u_x - B)$ and $(u_y - A)(u_x - B) \ge (l_y - A)(l_x - B)$. Then, the conditions (2), (3) can be simplified as:

$$\begin{aligned}(u_y - A)(l_x - B) - AB - C_1 \ge 0\\ (u_y - A)(u_x - B) - AB - C_2 \le 0\end{aligned}$$

Under the assumption of $(u_y - A)(l_x - B) \le (l_y - A)(u_x - B)$ and $(u_y - A)(u_x - B) \ge (l_y - A)(l_x - B)$, we have

$$-Bw_y + Aw_x + l_x u_y - u_x l_y \le 0 \qquad (6)$$
$$-Bw_y - Aw_x + u_x u_y - l_x l_y \ge 0 \qquad (7)$$

where $w_x = u_x - l_x, w_y = u_y - l_y$. Therefore,

$$\begin{aligned}(6)-(7) \Rightarrow & 2Aw_x + l_x u_y - u_x l_y - u_x u_y + l_x l_y \le 0\\ & 2Aw_x - w_x u_y - w_x l_y \le 0\\ & A \le \frac{u_y + l_y}{2}\\ \frac{l_x u_y - u_x l_y + Aw_x}{w_y} \le & B \le \frac{u_x u_y - l_x l_y - Aw_x}{w_y}\\ C_2 - C_1 \ge & u_y w_x - Aw_x \ge w_x \frac{u_y - l_y}{2} = \frac{w_x w_y}{2}\end{aligned}$$

Thus, the minimum of $C_2 - C_1$ can be reached when $A = \frac{u_y + l_y}{2}$. We substitute the A in the constraint of B by $\frac{u_y + l_y}{2}$.

$$\begin{aligned}l_x u_y - u_x l_y + \frac{u_y + l_y}{2} w_x \le & Bw_y \le u_x u_y - l_x l_y - \frac{u_y + l_y}{2} w_x\\ u_y \frac{u_x + l_x}{2} - l_y \frac{u_x + l_x}{2} \le & Bw_y \le u_y \frac{u_x + l_x}{2} - l_y \frac{u_x + l_x}{2}\\ \frac{u_x + l_x}{2} \le & B \le \frac{u_x + l_x}{2}\\ B = & \frac{u_x + l_x}{2}\end{aligned}$$

By substituting A and B in (4) and (5), we have:

$$\begin{aligned}C_1 \le & \min\left(-\frac{l_x \cdot l_y + u_x \cdot u_y}{2}, -\frac{u_x \cdot l_y + l_x \cdot u_y}{2}\right)\\ C_2 \ge & \max\left(-\frac{l_x \cdot l_y + u_x \cdot u_y}{2}, -\frac{u_x \cdot l_y + l_x \cdot u_y}{2}\right)\end{aligned}$$

By symmetry, regardless of the value of $(u_y - A)(l_x - B)$, $(l_y - A)(u_x - B)$, $(u_y - A)(u_x - B)$ and $(l_y - A)(l_x - B)$, the above property can be proved.

Case 2: $u_x \le B$ **and** $u_y \le A$. By definition,

$$\begin{aligned}&\min\{(xy - Ax - By - C_1)\}\\ &= (u_y - A)(u_x - B) - AB - C_1 \ge 0\\ &\max\{(xy - Ax - By - C_2)\}\\ &= (l_y - A)(l_x - B) - AB - C_2 \le 0\end{aligned}$$

$$\begin{cases} u_x u_y - Au_x - Bu_y - C_1 \ge 0\\ -l_x l_y + Al_x + Bl_y + C_2 \ge 0\\ A \ge u_y\\ B \ge u_x \end{cases} \qquad (8)$$

$$\begin{aligned}C_2 - C_1 &\ge l_x l_y - u_x u_y + Aw_x + Bw_y\\ &\ge l_x l_y - u_x u_y + u_y w_x + u_x w_y = w_x w_y \qquad (9)\end{aligned}$$

when $A = u_y$ and $B = u_x$. By symmetry, the above property (9) can be proved when $l_x \ge B$, $u_y \le A$ or $u_x \le B$, $l_y \ge A$ or $l_x \ge B$, $l_y \ge A$, a total of 4 cases including the current case. In this case, the minimum of $C_2 - C_1$ is larger than the one in case 1.

Case 3: $l_x \le B \le u_x$ **and** $u_y \le A$. By definition:

$$\begin{aligned}&\min\{(xy - Ax - By - C_1)\}\\ &= (l_y - A)(u_x - B) - AB - C_1 \ge 0\\ &\max\{(xy - Ax - By - C_2)\}\\ &= (l_y - A)(l_x - B) - AB - C_2 \le 0\end{aligned}$$

$$\begin{aligned}u_x l_y - Au_x - Bl_y - C_1 \ge 0\\ l_x l_y - Al_x - Bl_y - C_2 \le 0\end{aligned}$$

$$C_2 - C_1 \ge -l_y w_x + Aw_x \ge -l_y w_x + u_y w_x = w_x w_y \qquad (10)$$

when $A = u_y$. By symmetry, the above property (10) can be proved when $l_x \le B \le u_x$ and $l_y \ge A$ or $u_x \le B$ and $l_y \le A \le u_y$ or $l_x \ge B$ and $l_y \le A \le u_y$, a total of 4 cases including the current case. In this case, the minimum of $C_2 - C_1$ is larger than the one in case 1.

Then, the smallest $C_2 - C_1$ is reached, i.e., the near best zonotope approximation is reached when $l_x \le B \le u_x$ and $l_y \le A \le u_y$. □

B PROOF OF THEOREM 4.2

Tab. 8 shows nine cases for the abstract transformer design of the Sigmoid ⊙ Tanh function.

Table 8: Nine cases for the abstract transformer design of the Sigmoid ⊙ Tanh function.

Case	Conditions	Solutions	Proof
1	$l_x \geq 0$ and $l_y \geq 0$	$A = \frac{(\sigma(u_x)-\sigma(l_x))\tanh(u_y)+(\sigma(u_x)-\sigma(l_x))\tanh(l_y)}{2w_x}$, $B = \frac{(\sigma(u_x)+\sigma(l_x))\tanh(u_y)-(\sigma(u_x)+\sigma(l_x))\tanh(l_y)}{2w_y}$, $C_1 = f_{\sigma\cdot\tanh}(l_x,u_y) - Al_x - Bu_y$, $C_2 = f_{\sigma\cdot\tanh}(x^\star,y^\star) - Ax^\star - By^\star$, $(x^\star,y^\star)$ is the point of tangency of the concave surface $f_{\sigma\cdot\tanh}$ and tangent plane $Z^U = Ax + By + C_2$	Appendix B.1
2	$u_x \leq 0$ and $l_y \geq 0$	$A = \frac{f_{\sigma\cdot\tanh}(u_x,u_y)-f_{\sigma\cdot\tanh}(l_x,u_y))}{w_x}$, $B = \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x,u_y)$, $C_1 = f_{\sigma\cdot\tanh}(l_x,u_y) - Al_x - Bu_y$, $C_2 = f_{\sigma\cdot\tanh}(x^\star,y^\star) - Ax^\star - By^\star$ $(x^\star,y^\star) = \begin{cases} (u_x,l_y) & A \geq \frac{\tanh(l_y)}{4} \\ (x',l_y) & A < \frac{\tanh(l_y)}{4}, \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(u_x,l_y) \\ (u_x,l_y) & \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(u_x,l_y) \leq A < \frac{\tanh(l_y)}{4} \end{cases}$, where x' is the tangent point of the line $z = Ax + b$ to the curve $z = \tanh(l_y)\sigma(x)$	Appendix B.2.1
		$A = \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(u_x,l_y)$, $B = \begin{cases} \frac{f_{\sigma\cdot\tanh}(x',u_y)}{w_y} - \frac{f_{\sigma\cdot\tanh}(u_x,l_y)}{w_y} & A > \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(l_x,u_y) \\ \frac{f_{\sigma\cdot\tanh}(l_x,u_y)}{w_y} - \frac{f_{\sigma\cdot\tanh}(u_x,l_y)}{w_y} & A \leq \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(l_x,u_y) \end{cases}$, $C_1 = f_{\sigma\cdot\tanh}(u_x,l_y) - Au_x - Bl_y$, $C_2 = \begin{cases} \max\{f_{\sigma\cdot\tanh}(u_x,y') - Au_x - By', f_{\sigma\cdot\tanh}(l_x,y'') - Al_x - By''\} & \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,u_y) \leq B \\ \max\{f_{\sigma\cdot\tanh}(u_x,y') - Au_x - By', f_{\sigma\cdot\tanh}(l_x,u_y) - Al_x - Bu_y\} & \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,u_y) > B \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x,u_y), x' \text{ is the tangent point of } z = \sigma(x)\tanh(u_y) \\ \max\{f_{\sigma\cdot\tanh}(u_x,u_y) - Au_x - Bu_y, f_{\sigma\cdot\tanh}(l_x,u_y) - Al_x - Bu_y\} & \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,u_y), \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x,u_y) > B \end{cases}$ and $z = Ax + Bu_y + C_1$, y' is the tangent point of $z = \sigma(l_x)\tanh(y)$ and $z = Al_x + By + C_1$ and y'' is the tangent point of $z = \sigma(u_x)\tanh(y)$ and $z = Au_x + By + C_1$	Appendix B.2.2
		$A = \begin{cases} \frac{B(l_y-y')+f_{\sigma\cdot\tanh}(u_x,y')-f_{\sigma\cdot\tanh}(l_x,l_y)}{w_x} & B > \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x,l_y) \\ \frac{B(y''-y')+f_{\sigma\cdot\tanh}(u_x,y')-f_{\sigma\cdot\tanh}(l_x,y'')}{w_x} & B \leq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x,l_y) \end{cases}$, $B = \frac{f_{\sigma\cdot\tanh}(u_x,u_y)-f_{\sigma\cdot\tanh}(u_x,l_y)}{w_y}$, $C_1 = f_{\sigma\cdot\tanh}(u_x,y') - Au_x - By'$, $C_2 = f_{\sigma\cdot\tanh}(x^\star,u_y) - Ax^\star - Bu_y$, $x^\star = \begin{cases} l_x & A < \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(l_x,u_y) \\ x' & A \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(l_x,u_y) \end{cases}$,, where y' is the tangent point of $z = \sigma(u_x)\tanh(y)$ and $z = By + Au_x + C_2$, y'' is the tangent point of $z = \sigma(l_x)\tanh(y)$ and $z = By + Al_x + C_2$ if exists. x' is the tangent point of $z = \sigma(x)\tanh(u_y)$ and $z = Ax + b$	Appendix B.2.3
		$A = \begin{cases} \frac{B(l_y-y')+f_{\sigma\cdot\tanh}(u_x,y')-f_{\sigma\cdot\tanh}(l_x,l_y)}{w_x} & B > \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x,l_y) \\ \frac{B(y''-y')+f_{\sigma\cdot\tanh}(u_x,y')-f_{\sigma\cdot\tanh}(l_x,y'')}{w_x} & B \leq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x,l_y) \end{cases}$, $B = \frac{f_{\sigma\cdot\tanh}(u_x,u_y)-f_{\sigma\cdot\tanh}(u_x,l_y)+f_{\sigma\cdot\tanh}(l_x,u_y)-f_{\sigma\cdot\tanh}(l_x,l_y)}{2w_y}$, $C_1 = f_{\sigma\cdot\tanh}(u_x,y') - Au_x - By', C_2 = \begin{cases} \min\{F(l_x,u_y),F(u_x,l_y)\} & \frac{\partial F}{\partial x}(l_x,u_y) \geq 0 \geq \frac{\partial F}{\partial x}(u_x,l_y) \\ \min\{F(l_x,u_y),F(x',l_y)\} & \frac{\partial F}{\partial x}(l_x,u_y) \geq 0, \frac{\partial F}{\partial x}(u_x,l_y) > 0 \\ \min\{F(x'',u_y),F(u_x,l_y)\} & \frac{\partial F}{\partial x}(l_x,u_y) < 0, \frac{\partial F}{\partial x}(u_x,l_y) \leq 0 \\ \min\{F(x'',u_y),F(x',l_y)\} & \frac{\partial F}{\partial x}(l_x,u_y) < 0 < \frac{\partial F}{\partial x}(u_x,l_y) \end{cases}$, $F(x,y) = f_{\sigma\cdot\tanh}(x,y) - Ax - By$, x' is the tangent point of $z = \sigma(x)\tanh(l_y)$ and $z = Ax + Bl_y + C_1$, x'' is the tangent point of $z = \sigma(x)\tanh(u_y)$ and $z = Ax + Bu_y + C_1$, y' is the tangent point of $z = \sigma(u_x)\tanh(y)$ and $z = By + Au_x + b$ and y'' is the tangent point of $z = \sigma(l_x)\tanh(y)$ and $z = By + Al_x + b$	Appendix B.2.4
		$A = \frac{(\sigma(u_x)-\sigma(l_x))(\tanh(u_y)+\tanh(l_y))}{2w_x}$, $B = \frac{(\sigma(u_x)+\sigma(l_x))(\tanh(u_y)-\tanh(l_y))}{2w_y}$, $C_1 = \begin{cases} \min\{F(x',l_y),F(x'',u_y)\} & G_2(l_x,u_y) < A, G_2(u_x,l_y) > A \\ F(x'',l_y) & G_2(l_x,u_y) < A, G_2(u_x,l_y) \leq A \\ F(x',l_y) & G_2(l_x,u_y) \geq A, G_2(u_x,l_y) > A \\ F(u_x,l_y) & G_2(l_x,u_y) \geq A, G_2(u_x,l_y) \leq A \end{cases}$, $C_2 = \begin{cases} \max\{F(x',l_y),F(x'',u_y)\} & G_1(u_x,u_y) < B < G_1(l_x,l_y) \\ F(x'',u_y) & G_1(u_x,u_y) < B, G_1(l_x,l_y) \leq B \\ \max\{F(u_x,u_y),F(l_x,u_y)\} & G_1(u_x,u_y) \geq B, G_1(l_x,l_y) > B, B < 0 \\ \max\{F(u_x,l_y),F(l_x,y')\} & G_1(u_x,u_y) \geq B, G_1(l_x,l_y) > B, B \geq 0 \\ F(l_x,l_y) & G_1(u_x,u_y) \geq B, G_1(l_x,l_y) \leq B \end{cases}$, $G_1(x,y) = \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(x,y)$, $G_2(x,y) = \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(x,y)$, $F(x,y) = f_{\sigma\cdot\tanh}(x,y) - Ax - By$	Appendix B.2.5
3	$u_x \leq 0$ and $u_y \leq 0$	This case is symmetric to the case 2, and the proof can be deducted in the same way.	
4	$l_x \geq 0$ and $u_y \leq 0$	This case is symmetric to the case 1, and the proof can be deducted in the same way.	
5	$l_x \geq 0, u_y > 0$ and $l_y < 0$	$A = 0$, $B = \min\{\frac{f_{\sigma\cdot\tanh}(u_x,u_y) - f_{\sigma\cdot\tanh}(l_x,l_y)}{w_y}, \frac{f_{\sigma\cdot\tanh}(l_x,u_y) - f_{\sigma\cdot\tanh}(u_x,l_y)}{w_y}\}$, $C_1 = f_{\sigma\cdot\tanh}(u_x,y^{\star\star}) - Au_x - By^{\star\star}$, $C_2 = f_{\sigma\cdot\tanh}(u_x,y^\star) - Au_x - By^\star$ $y^\star = \begin{cases} y' & B \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,u_y) \\ u_y & B < \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,u_y) \end{cases}, y^{\star\star} = \begin{cases} y'' & B \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,l_y) \\ l_y & B < \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,l_y) \end{cases}$, where y' is the tangent point of $By + C_2$ and $\sigma(u_x)\tanh(y)$ when $B \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,u_y)$ and y'' is the tangent point of $By + C_1$ and $\sigma(u_x)\tanh(y)$ when $B \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x,l_y)$	Appendix B.3
6	$u_x \leq 0, u_y > 0$ and $l_y < 0$	This case is symmetric to the case 5, and the proof can be deducted in the same way.	
7	$l_y \geq 0, u_x > 0$ and $l_x < 0$	$A = \frac{f_{\sigma\cdot\tanh}(u_x,l_y)-f_{\sigma\cdot\tanh}(l_x,l_y)}{w_x}$, $B = \frac{f_{\sigma\cdot\tanh}(l_x,u_y)-f_{\sigma\cdot\tanh}(l_x,l_y)}{w_y}$, $C_1 = f_{\sigma\cdot\tanh}(x^{\star\star},y^{\star\star}) - Ax^{\star\star} - By^{\star\star}$, $C_2 = f_{\sigma\cdot\tanh}(x^\star,l_y) - Ax^\star - Bl_y$	Appendix B.4
8	$u_y \leq 0, u_x > 0$ and $l_x < 0$	This case is symmetric to the case 7, and the proof can be deducted in the same way.	
9	$u_y > 0, l_y < 0$, $u_x > 0$ and $l_x < 0$	In this case, we use the same method used in case 5 and case 6.	

B.1 Proof for Case 1

Proof. Suppose line 1 crosses $(l_x, l_y, f_{\sigma\cdot\tanh}(l_x, l_y))$ and $(u_x, u_y, f_{\sigma\cdot\tanh}(u_x, u_y))$, and line 2 crosses $(l_x, u_y, f_{\sigma\cdot\tanh}(l_x, u_y))$ and $(u_x, l_y, f_{\sigma\cdot\tanh}(u_x, l_y))$, the plane that is parallel to the two lines has two properties:

$$\begin{cases} f_{\sigma\cdot\tanh}(l_x, l_y) + Aw_x + Bw_y = f_{\sigma\cdot\tanh}(u_x, u_y) \\ f_{\sigma\cdot\tanh}(u_x, l_y) - Aw_x + Bw_y = f_{\sigma\cdot\tanh}(l_x, u_y) \end{cases}$$

Hence, we have the slope for x and y:

$$\begin{cases} A = \frac{(\sigma(u_x)-\sigma(l_x))(\tanh(u_y)+\tanh(l_y))}{2w_x} \\ B = \frac{(\sigma(u_x)+\sigma(l_x))(\tanh(u_y)-\tanh(l_y))}{2w_y} \end{cases}$$

Notice that the heights of the middle points of the two lines are $\frac{f_{\sigma\cdot\tanh}(u_x,u_y)+f_{\sigma\cdot\tanh}(l_x,l_y)}{2}$ and $\frac{f_{\sigma\cdot\tanh}(l_x,u_y)+f_{\sigma\cdot\tanh}(u_x,l_y)}{2}$, respectively. Therefore, we have:

$$\begin{aligned} &\frac{f_{\sigma\cdot\tanh}(u_x, u_y) + f_{\sigma\cdot\tanh}(l_x, l_y)}{2} \\ &- \frac{f_{\sigma\cdot\tanh}(l_x, u_y) + f_{\sigma\cdot\tanh}(u_x, l_y)}{2} \\ &= \frac{1}{2}\left(\sigma(u_x) - \sigma(l_x)\right)\left(\tanh(u_y) - \tanh(l_y)\right) \geq 0 \end{aligned}$$

The highest lower plane that parallel to both line 1 and line 2 crosses through line 2. Hence, $C_1 = f_{\sigma\cdot\tanh}(l_x, u_y) - Al_x - Bu_y = f_{\sigma\cdot\tanh}(u_x, l_y) - Au_x - Bl_y$. □

B.2 Proof for Case 2

B.2.1 Proof for Case 2 (1).

Proof. The A of the upper plane is defined by crossing points $\sigma(l_x)\tanh(u_y)$ and $\sigma(u_x)\tanh(u_y)$. Since the curve $z = \sigma(x)\tanh(u_y)$ is convex, any point on $z = \sigma(x)\tanh(u_y)$ is below the plane. Define B as the slope of y at point $\sigma(l_x)\tanh(u_y)$, for any (x, y):

$$\begin{aligned} &\frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(x, y) - \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x, u_y) \\ =&\sigma(x)\frac{d\tanh(y)}{dy} - \sigma(l_x)\frac{d\tanh(u_y)}{dy} \\ =&\sigma(x)\left(\frac{d\tanh(y)}{dy} - \frac{d\tanh(u_y)}{dy}\right) \\ &+ (\sigma(x) - \sigma(l_x))\frac{d\tanh(u_y)}{dy} \geq 0 \end{aligned}$$

Thus, for any (x, y), we have:

$$\begin{aligned} B &< \sigma(x)\frac{\tanh(u_y) - \tanh(y)}{u_y - y} \\ \sigma(x)\tanh(y) &< B(y - u_y) + \sigma(x)\tanh(u_y) \\ \sigma(x)\tanh(y) &< B(y - u_y) + Ax + Bu_y + C_2 = Z^U \end{aligned}$$

Hence, the upper bound is proved.

For $A \geq \frac{\tanh(l_y)}{4}$, it is obvious that $A \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(u_x, l_y)$. Since C_1 can be calculated by the point of (u_x, l_y), we have:

$$\begin{aligned} \sigma(x)\tanh(l_y) &> Ax + Bl_y + C_1 \\ B &< \sigma(x)\frac{\tanh(y) - \tanh(l_y)}{y - l_y} \\ \sigma(x)\tanh(y) &> B(y - l_y) + \sigma(x)\tanh(l_y) \\ \sigma(x)\tanh(y) &> B(y - l_y) + Ax + Bl_y + C_1 = Z^L \end{aligned}$$

Hence, the lower bound is proved. □

B.2.2 Proof for Case 2 (2).

Proof. When $A > \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x, u_y)$, since $\frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(x', u_y) = \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(u_x, l_y)$, we have $\sigma(x')(1-\sigma(x'))\tanh(u_y) = \sigma(u_x)(1-\sigma(u_x))\tanh(l_y)$ and $x' < u_x$. Then:

$$\begin{aligned} B &= \frac{f_{\sigma\cdot\tanh}(x', u_y) - f_{\sigma\cdot\tanh}(u_x, l_y))}{w_y} \\ &= \frac{1}{w_y}\left(\frac{\sigma(u_x)(1-\sigma(u_x))\tanh(l_y)}{1-\sigma(x')} - \sigma(u_x)\tanh(l_y)\right) \\ &= \frac{\sigma(u_x)\tanh(l_y)}{w_y}\left(\frac{\sigma(x') - \sigma(u_x)}{1-\sigma(x')}\right) < 0, \end{aligned}$$

$A = \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(u_x, l_y)$ and C_1 is calculated by (u_x, l_y). Then, we have $z = Ax + Bl_y + C_1$ lower than $f_{\sigma\cdot\tanh}(x, l_y)$ for any x.

$$\begin{aligned} \sigma(x)\tanh(y) > \sigma(x)\tanh(l_y) &> Ax + Bl_y + C_1 \\ &> Ax + By + C_1 \end{aligned}$$

When $A \leq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x, u_y)$, by definition, we have $z = Ax + Bu_y + C_1$ lower than $f_{\sigma\cdot\tanh}(x, u_y)$ for any x. Given x, $f_{\sigma\cdot\tanh}(x, y)$ is concave. Therefore, for any y, $f_{\sigma\cdot\tanh}(x, y) > Ax + By + C_1$ holds. Thus, the lower bound is proved. In addition, C_2 is defined by the higher value where $z = By + Au_x + C_2$ is higher than $z = f_{\sigma\cdot\tanh}(u_x, y)$ and $z = By + Al_x + C_2$ is higher than $z = f_{\sigma\cdot\tanh}(l_x, y)$. Therefore, with the convexity of $f_{\sigma\cdot\tanh}(x, y)$, given y, the plane $z = Ax + By + C_2$ is higher than $f_{\sigma\cdot\tanh}(x, y)$. □

B.2.3 Proof for Case 2 (3).

Proof. For $B \leq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x, l_y)$, since (u_x, y') is the tangent point of $By + Au_x + C_2$ and $\sigma(u_x)\tanh(y)$, therefore $By + Au_x + C_2$ is above $\sigma(u_x)\tanh(y)$. Since (l_x, y'') is the tangent point of $By + Al_x + C_2$ and $\sigma(l_x)\tanh(y)$, therefore, $By + Al_x + C_2$ is above $\sigma(l_x)\tanh(y)$. With the convexity of $\sigma(x)\tanh(y)$ given y, we have $\sigma(x)\tanh(y) < By + Ax + C_2$. The detailed proof is shown below:

$$\begin{aligned} &Au_x + By + C_2 - \sigma(u_x)\tanh(y) \\ =&Au_x + By + \sigma(u_x)\tanh(y') - Au_x - By' - \sigma(u_x)\tanh(y) \\ =&\frac{\sigma(u_x)\tanh(u_y) - \sigma(u_x)\tanh(l_y)}{w_y}(y - y') \\ &+ \sigma(u_x)(\tanh(y') - \tanh(y)) \\ =&\sigma(u_x)(y - y')\left(\frac{\tanh(u_y) - \tanh(l_y)}{u_y - l_y} - \frac{\tanh(y) - \tanh(y')}{y - y'}\right) \leq 0 \end{aligned}$$

Similarly, for $B > \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(l_x, l_y)$, $By+Ax+C_2$ is above $\sigma(x)\tanh(y)$. Hence, the upper bound is proved.

For $A < \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(l_x, u_y)$, we have:

$$\begin{aligned}
&A(x-x'') + \sigma(x'')\tanh(l_y) + Bw_y - A(x-l_x) - \sigma(l_x)\tanh(u_y)\\
=&A(l_x - x'') + \tanh(u_y)(\sigma(u_x) - \sigma(l_x)) + \tanh(l_y)(\sigma(x'') - \sigma(u_x))\\
>&\frac{\tanh(u_y)(\sigma(u_x)-\sigma(l_x))}{u_x - l_x}(u_x - x'') + \tanh(l_y)(\sigma(x'') - \sigma(u_x))\\
=&\left(\frac{\tanh(u_y)(\sigma(u_x)-\sigma(l_x))}{u_x - l_x} - \frac{\tanh(l_y)(\sigma(u_x)-\sigma(x''))}{u_x - x'}\right)(u_x - x'')\\
>&0
\end{aligned}$$

where x'' is the tangent point of $A(x-x'') + \sigma(x'')\tanh(l_y)$ and $\sigma(x)\tanh(l_y)$ if exists. Thus, the line $A(x-x'') + \sigma(x')\tanh(l_y) + Bw_y$ is higher than the line $A(x-l_x) + \sigma(l_x)\tanh(u_y)$. Therefore, we use the lower plane which crosses line $A(x-l_x)+\sigma(l_x)\tanh(u_y)$ and can be determined by point (l_x, u_y). When $A \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(lx, uy)$, it can be proved in same way that the lower plane crosses $A(x - x') + \sigma(x')\tanh(u_y)$. Hence, the lower bound is proved. □

B.2.4 Proof for Case 2 (4).

Proof. The upper bound can be proved in the same way as in Section B.2.3. The lower bound coefficient C_1 can be chosen in the same way as in Section B.2.3, which makes $Ax + Bu_y + C_1$ below $\sigma(x)\tanh(u_y)$ and $Ax + Bl_y + C_1$ below $\sigma(x)\tanh(l_y)$. Therefore, the lower bound can be proved. □

B.2.5 Proof for Case 2 (5).

Proof. Similar to the proof in previous sections, the upper bound is determined by making $Al_x + By + C_2$ above $\sigma(l_x)\tanh(y)$ and $Au_x + By + C_2$ above $\sigma(u_x)\tanh(y)$. With the convexity, $Ax + By + C_2$ is above $\sigma(x)\tanh(y)$. The lower bound is determined by making $Ax + Bl_y + C_1$ below $\sigma(x)\tanh(l_y)$ and $Ax + Bu_y + C_1$ below $\sigma(x)\tanh(u_y)$. With the concavity, $Ax + By + C_1$ is below $\sigma(x)\tanh(y)$. □

B.3 Proof for Case 5

Proof. If $B \geq \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x, u_y)$, then y' exists and $B = \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x, y')$. We also have $\frac{f_{\sigma\cdot\tanh}(u_x,u_y) - f_{\sigma\cdot\tanh}(u_x,y')}{u_y - y'} < B$. Then,

$$\begin{aligned}
&B(y' - l_y) - (\sigma(u_x)\tanh(y') - \sigma(l_x)\tanh(l_y))\\
=&B(y' - l_y) + f_{\sigma\cdot\tanh}(u_x, u_y) - f_{\sigma\cdot\tanh}(u_x, y')\\
&- \left(\sigma(u_x)\tanh(y') - \sigma(l_x)\tanh(l_y)\right)\\
&- \left(f_{\sigma\cdot\tanh}(u_x, u_y) - f_{\sigma\cdot\tanh}(u_x, y')\right)\\
<&B(y' - l_y) + B(u_y - y') - \left(\sigma(u_x)\tanh(u_y) - \sigma(l_x)\tanh(_l y)\right)\\
<&B(u_y - l_y) - \left(\sigma(u_x)\tanh(u_y) - \sigma(l_x)\tanh(l_y)\right) < 0
\end{aligned}$$

Thus, we have $B(l_y - y') + \sigma(u_x)\tanh(y') > \sigma(l_x)\tanh(l_y)$. With the concavity in $[0, u_x]$, $\frac{\sigma(u_x)\tanh(y')}{y'} > B$, we have $B(0 - y') + \sigma(u_x)\tanh(y') > \sigma(l_x)\tanh(0)$. With the convexity in $[l_x, 0]$, we have $B(y - y') + \sigma(u_x)\tanh(y') > \sigma(l_x)\tanh(y) > \sigma(x)\tanh(y)$. Hence, the upper bound is proved.

If $B < \frac{\partial f_{\sigma\cdot\tanh}}{\partial y}(u_x, u_y)$, we have $B(y - u_y) + \sigma(u_x)\tanh(u_y) > f_{\sigma\cdot\tanh}(u_x, y)$ when $y \in [0, u_y]$ and $B(0 - u_y) + \sigma(u_x)\tanh(u_y) > 0$. Therefore, for $y < u_y$, we have:

$$\frac{f_{\sigma\cdot\tanh}(u_x, u_y) - f_{\sigma\cdot\tanh}(l_x, l_y)}{u_y - l_y}(y - u_y) \leq B(y - u_y)$$

With the convexity of $f_{\sigma\cdot\tanh}(l_x, y)$ with $y \in [l_y, 0]$, we have $B(y - u_y) + \sigma(u_x)\tanh(u_y) > f_{\sigma\cdot\tanh}(l_x, y)$. Thus, the upper bound is proved. Similarly, the lower bound can be proved in the same manner. □

B.4 Proof for Case 7

Proof. To prove the lower bound, we only need to prove that $Ax + Bl_y + C_1$ is below $\sigma(x)\tanh(l_y)$ and $Ax + Bu_y + C_1$ is below $\sigma(x)\tanh(u_y)$. With the concavity of $\sigma(x)\tanh(y)$ given x, the lower bound can be proved.

As C_1 is determined by making the line $Ax + Bl_y + C_1$ lower than $\sigma(x)\tanh(l_y)$, we have $Ax + Bl_y + C_1 < \sigma(x)\tanh(l_y)$. Now, we only need to prove that line $Ax + Bu_y + C_1$ is lower than $\sigma(x)\tanh(u_y)$.

$$\begin{aligned}
B &= \frac{\sigma(l_x)\tanh(u_y) - \sigma(l_x)\tanh(l_y)}{w_y}\\
B &\leq \frac{\sigma(x)\tanh(u_y) - \sigma(x)\tanh(l_y)}{w_y}\\
\sigma(x)\tanh(u_y) &> \sigma(x)\tanh(l_y) + Bw_y\\
\sigma(x)\tanh(u_y) &> Ax + Bl_y + C_1 + Bw_y\\
\sigma(x)\tanh(u_y) &> Ax + Bu_y + C_1
\end{aligned}$$

Thus, the lower bound is proved.

Given $x \geq 0$, since $f_{\sigma\cdot\tanh}(x, y)$ is concave, the plane $Ax+By+C_2$ determined by the tangent point is above $f_{\sigma\cdot\tanh}(x, y)$. Therefore, we only need to prove that when $x < 0$, the plane $Ax + By + C_2$ is also above $f_{\sigma\cdot\tanh}(x, y)$. To prove this, we need to prove that $A \times 0 + By + C_2$ is above $f_{\sigma\cdot\tanh}(0, y)$ and $Al_x + By + C_2$ is above $f_{\sigma\cdot\tanh}(l_x, y)$ due to the convexity. Since $A \times 0 + By + C_2$ above $f_{\sigma\cdot\tanh}(0, y)$ has been proved in the $x \geq 0$ case, now we need to prove that $Al_x + By + C_2$ is above $f_{\sigma\cdot\tanh}(l_x, y)$. For $x' < u_x$, we have $\frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(x', y^{\star\star}) = A = \frac{\partial f_{\sigma\cdot\tanh}}{\partial x}(x'', l_y)$. For $x'' < x' < u_x$ and $x'' < x < u_x$, we have $\sigma(x) > \frac{\sigma(u_x)-\sigma(l_x)}{w_x}(x - l_x) + \sigma(l_x)$ which means $\frac{\sigma(x')-\sigma(l_x)}{x'-l_x} > \frac{\sigma(u_x)-\sigma(l_x)}{w_x}$. Thus,

$$\begin{aligned}
Ax' + By + C_2 >&\sigma(x')\tanh(y)\\
A(x' - l_x) >&\sigma(x')\tanh(y) - (Al_x + By + C_2)\\
A(x' - l_x) >&\tanh(y)(\sigma(x') - \sigma(l_x))\\
&- (Al_x + By + C_2) + \sigma(l_x)\tanh(y)\\
\sigma(l_x)\tanh(y) <&(x' - l_x)\left(\frac{\tanh(l_y)(\sigma(u_x) - \sigma(l_x))}{u_x - l_x}\right.\\
&\left. - \frac{\tanh(y)(\sigma(x') - \sigma(l_x))}{x' - l_x}\right) + (Al_x + By + C_2)\\
\sigma(l_x)\tanh(y) <&(x' - l_x)\tanh(l_y)\left(\frac{\sigma(u_x) - \sigma(l_x)}{u_x - l_x} - \frac{\sigma(x') - \sigma(l_x)}{x' - l_x}\right)\\
&+ (Al_x + By + C_2)\\
\sigma(l_x)\tanh(y) <&(Al_x + By + C_2)
\end{aligned}$$

Hence, the upper bound is proved. □

Table 9: Three cases for the abstract transformer design of the Sigmoid ⊙ Identity function.

Case	Conditions	Solutions	Proof
1	$l_x \geq 0$	$A = \frac{(\sigma(u_x)-\sigma(l_x))(\tanh(u_y)+\tanh(l_y))}{2w_x}$, $B = \frac{(\sigma(u_x)+\sigma(l_x))(\tanh(u_y)-\tanh(l_y))}{2w_y}$, $C_1 = f_{x\cdot\sigma}(x^{\star\star}, y^{\star\star}) - Ax^{\star\star} - By^{\star\star}$, $C_2 = f_{x\cdot\sigma}(x^{\star}, y^{\star}) - Ax^{\star} - By^{\star}$ $(x^{\star}, y^{\star})$ is determined to make $Au_x + By + C_2$ above $u_x\sigma(y)$ and $Al_x + By + C_2$ above $l_x\sigma(y)$. $(x^{\star\star}, y^{\star\star})$ is determined to make $Au_x + By + C_1$ below $u_x\sigma(y)$ and $Al_x + By + C_1$ below $l_x\sigma(y)$.	**Appendix C.1**
2	$u_x \leq 0$	In this case, we use the same method used in Case 1.	
3	$l_x < 0$ and $u_x > 0$	$A = \min\{\frac{f_{x\cdot\sigma}(u_x,u_y)-f_{x\cdot\sigma}(l_x,l_y)}{w_x}, \frac{f_{x\cdot\sigma}(u_x,l_y)-f_{x\cdot\sigma}(l_x,u_y)}{w_x}\}$, $B = 0$, $C_1 = f_{x\cdot\sigma}(l_x, u_y) - Al_x$, $C_2 = f_{x\cdot\sigma}(u_x, u_y) - Au_x$	**Appendix C.2**

C PROOF OF THEOREM 4.3

Tab. 9 shows the three cases for the abstract transformer design of the Sigmoid ⊙ Identity function.

C.1 Proof for Case 1

Proof. In this case, C_2 is determined by making $Au_x + By + C_2$ above $u_x\sigma(y)$ and making $Al_x + By + C_2$ above $l_x\sigma(y)$. Denote $x \in [l_x, u_x]$ as $x = \alpha l_x + (1-\alpha)u_x$, where $\alpha \in [0, 1]$. When both conditions are satisfied, we have

$$\begin{aligned} &Ax + By + C_2 = A\alpha l_x + A(1-\alpha)u_x + By + C_2 \\ =&\alpha(Al_x + By + C_2) + (1-\alpha)(Au_x + By + C_2) \\ \geq&\alpha l_x\sigma(y) + (1-\alpha)u_x\sigma(y) = \sigma(y)(\alpha l_x + (1-\alpha)u_x) = x\sigma(y) \end{aligned}$$

Thus, the upper bound is proved.

Similarly, C_1 is determined by making $Au_x + By + C_1$ below $u_x\sigma(y)$ and making $Al_x + By + C_1$ below $l_x\sigma(y)$. Thus, the lower bound can be proved in the same way. □

C.2 Proof for Case 3

Proof. By the definition of A, C_1 and C_2, we have

$$\begin{cases} Au_x + C_2 \geq f_{x\cdot\sigma}(u_x, y) \\ Al_x + C_2 \geq f_{x\cdot\sigma}(l_x, y) \end{cases} \qquad \begin{cases} Au_x + C_1 \leq f_{x\cdot\sigma}(u_x, y) \\ Al_x + C_1 \leq f_{x\cdot\sigma}(l_x, y) \end{cases}$$

Then

$$\begin{aligned} Ax + C_2 &= A(\alpha l_x + (1-\alpha)u_x) + C_2 \\ &= \alpha(Al_x + C_2) + (1-\alpha)(Au_x + C_2) \\ &\geq \alpha l_x\sigma(y) + (1-\alpha)u_x\sigma(y) = x\sigma(y) \\ Ax + C_1 &= A(\alpha l_x + (1-\alpha)u_x) + C_2 \\ &= \alpha(Al_x + C_1) + (1-\alpha)(Au_x + C_1) \\ &\leq \alpha l_x\sigma(y) + (1-\alpha)u_x\sigma(y) = x\sigma(y) \end{aligned}$$

Hence, the upper and lower bounds are proved. □