GaLileo: General Linear Relaxation Framework for Tightening Robustness Certification of Transformers

Yunruo Zhang[1], **Lujia Shen**[2], **Shanqing Guo**[1*], **Shouling Ji**[2]

[1] School of Cyber Science and Technology, Shandong University
[2] College of Computer Science and Technology, Zhejiang University
zhangyunruo@mail.sdu.edu.cn, shen.lujia@zju.edu.cn, guoshanqing@sdu.edu.cn, sji@zju.edu.cn

Abstract

Transformers based on attention mechanisms exhibit vulnerability to adversarial examples, posing a substantial threat to the security of their applications. Aiming to solve this problem, the concept of robustness certification is introduced to formally ascertain the presence of any adversarial example within a specified region surrounding a given sample. However, prior works have neglected the dependencies among inputs of softmax (the most complex function in attention mechanisms) during linear relaxations. This oversight has consequently led to imprecise certification results. In this work, we introduce **GaLileo**, a **G**ener**a**l **Li**near Re**l**axation Frame**w**o**r**k designed to certify the robustness of Transformers. GaLileo effectively surmounts the trade-off between precision and efficiency in robustness certification through our innovative n-dimensional relaxation approach. Notably, our relaxation technique represents a pioneering effort as the first linear relaxation for n-dimensional functions such as softmax. Our novel approach successfully transcends the challenges posed by the curse of dimensionality inherent in linear relaxations, thereby enhancing linear bounds by incorporating input dependencies. Our evaluations encompassed a thorough analysis utilizing the SST and Yelp datasets along with diverse Transformers of different depths and widths. The experimental results demonstrate that, as compared to the baseline method CROWN-BaF, GaLileo achieves up to 3.24 times larger certified radii while requiring similar running times. Additionally, GaLileo successfully attains certification for Transformers' robustness against multi-word ℓ_p perturbations, marking a notable accomplishment in this field.

Introduction

Transformers (Vaswani et al. 2017) have found applications across a wide range of fields, including natural language processing (Vaswani et al. 2017), image classification (Dosovitskiy et al. 2021), and automatic speech recognition (Dong, Xu, and Xu 2018). The introduction of large-scale pre-trained language models like BERT (Devlin et al. 2019), GPT-3 (Brown et al. 2020), and Chat-GPT (OpenAI 2022) has brought about a profound impact on modern society, reshaping the future by exceeding human performance levels. However, despite their remarkable achievements, recent research (Guo et al. 2021; Joshi, Jagatap, and Hegde 2021) has unveiled a significant vulnerability in Transformers——*adversarial attacks* (Szegedy et al. 2014; Papernot et al. 2016a). These attacks involve introducing subtle perturbations (e.g., ℓ_p perturbations) to clean input data, causing the target model to behave unexpectedly. These perturbed inputs are referred to as adversarial examples. The susceptibility of Transformers to these attacks highlights their lack of robustness, raising concerns about their deployment in safety-critical scenarios such as facial recognition (Tran, Vu, and Nguyen 2022) and autonomous driving (Prakash, Chitta, and Geiger 2021). In response to this issue, earlier defensive strategies, often founded on heuristics, were proposed to empirically bolster the robustness of the target model, including adversarial training (Goodfellow, Shlens, and Szegedy 2015) and model distillation (Papernot et al. 2016b). However, subsequent investigations (Carlini and Wagner 2017; Madry et al. 2018) have revealed that these empirical defenses lack solid theoretical guarantees and are often outwitted by more sophisticated attack techniques.

To end the everlasting competition between attackers and defenders, recent researchers have shifted their focus to *robustness certification* (Bunel et al. 2018; Weng et al. 2018; Raghunathan, Steinhardt, and Liang 2018b), which quantitatively measure the model's robustness by certifying whether its predictions change when input samples are perturbed. However, for large-scale and highly nonlinear models like ResNet (He et al. 2016) and Transformers, it becomes impractical to certify their robustness without error. To address this challenge, previous robustness certification methods have commonly employed linear relaxations to handle the non-linear functions in these models, such as ReLU and sigmoid. Unfortunately, relaxation-based methods trade precision for efficiency, leading to imprecise certification results. Thus, researchers have been dedicated to the pursuit of tighter relaxations (Salman et al. 2019; Zhang et al. 2022), which is a crucial area of research in robustness certification.

Limitation of prior works. Indeed, prior works (Shi et al. 2020; Bonaert et al. 2021) have made valuable contributions to the robustness certification of Transformers. However, we contend that their precision is compromised due to the loose relaxations applied to softmax functions. Specifically, they decompose softmax functions into several unary non-linear functions (e.g., exponential and reciprocal functions) and relax them individually, neglecting the dependen-

*Corresponding author.

cies among the inputs of softmax. As Transformers typically consist of multiple attention layers, the error introduced by these looser relaxations accumulates layer by layer, leading to trivial certification results. Addressing the issue of precision requires considering dependencies among inputs in linear relaxations, which inevitably involves calculations in an n-dimensional space. For instance, some researchers (Singh et al. 2019; Tjandraatmadja et al. 2020) attempt to tighten relaxations by jointly relaxing k ReLU neurons in a k-dimensional space. However, their methods come with a significant increase in computational time and are limited to simple neural networks comprising only affine transformations and specific activation functions (e.g., ReLU). Thus, they cannot be directly extended to larger and more complex models (e.g., Transformers) that involve functions such as softmax. Thus, tighter the relaxation of softmax is urgently needed in certifying the robustness of Transformers.

Key Challenge. The key challenge in designing linear relaxations for n-dimensional functions is the *curse of dimensionality*. Specifically, most linear relaxations are designed to be able to choose proper bounds according to the numerical bounds of its input because the nonlinearity of a certain segment of a non-linear function varies with respect to its input range. For instance, previous relaxations for unary functions usually consider 3 cases of bounds (Zhang et al. 2018) and those for binary functions usually consider 3^2 cases at most (Du et al. 2021). Thus, naively extending previous relaxations into an n-dimensional space leads to 3^n cases in the worst-case scenario, which is infeasible in practice.

This work. In this work, we address the above challenge by decoupling the nonlinearity in each dimension, which permits us to compute accurate bounds independently in each dimension without necessitating the consideration of bounds in other dimensions. We substantiate its soundness and efficiency via rigorous theoretical proofs. Leveraging this approach, we propose a general linear relaxation framework called GaLileo for certifying the robustness of Transformers. To assess GaLileo's performance, we conduct extensive evaluations on the SST and Yelp datasets, employing various Transformers for comparison with the baseline method, CROWN-BaF (Shi et al. 2020). The results show that GaLileo achieves *up to* 3.24 *times larger certified radii* than CROWN-BaF while maintaining *similar running times*, which indicates that our relaxation is indeed tighter and more precise. Moreover, since prior works in the field have primarily considered only 1 or 2-word perturbations, we further explore Transformers' robustness by certifying it under multi-word (≥ 3) ℓ_p perturbations. This extension broadens the scope of our research and enhances the understanding of Transformers' robustness in more complex scenarios.

Contributions. Our main contributions are:

- We identify the main limitation of prior works on robustness certification for Transformers, i.e., they neglect dependencies among the inputs in the relaxation of softmax functions, leading to imprecise certification results.
- To the best of our knowledge, we design the first n-dimensional linear relaxation for non-linear functions such as softmax, which utilize dependencies between the inputs to tighten bounds. Moreover, we theoretically prove the soundness and efficiency of our relaxation.
- We propose GaLileo, a general linear relaxation framework designed to certify the robustness of Transformers. GaLileo effectively addresses the traditional trade-off between precision and efficiency, setting a new standard in robustness certification of Transformers.
- We conduct comprehensive evaluations to demonstrate that GaLileo achieves up to 3.24 times larger certified radii than CROWN-BaF while consuming similar times. Furthermore, this is the first work to certify Transformers' robustness under multi-word (≥ 3) ℓ_p perturbations.

Related Works. A line of work relevant to ours is *robustness certification* for deep neural networks, which leverage mathematical techniques to certify whether the prediction of a DNN remains consistent when a given sample is subjected to perturbations within a specified region surrounding it. Earlier works on robustness certification usually model the robustness certification problem as satisfiability modulo theories (SMT) problems (Katz et al. 2017; Bunel et al. 2018) or mixed integer linear programming (MILP) problems (Cheng, Nührenberg, and Ruess 2017). Though precise, those methods are limited to small DNNs due to the lack of polynomial-time solutions for SMT and MILP problems. To scale up to larger DNNs, later robustness certification methods (Raghunathan, Steinhardt, and Liang 2018b,a) usually apply relaxations to trade precision for efficiency, among which linear relaxation-based methods (Weng et al. 2018; Salman et al. 2019; Tjandraatmadja et al. 2020) are shown to be more promising. One of the most widely adopted linear relaxation-based methods are *CROWN-like methods* (Zhang et al. 2018; Shi et al. 2020), which propagate linear bounds of neurons through the model and achieve a good balance between precision and efficiency.

As for certifying the robustness of Transformers, it's important to note that the straightforward application of existing methods designed for other models can lead to significantly prolonged running times. CROWN-BaF (Shi et al. 2020) addresses the above issue by using the forward mode to handle non-linear functions in attention layers. Unlike CROWN-BaF, DeepT (Bonaert et al. 2021), which is based on abstract interpretation and Zonotopes, improves efficiency via noise symbol reduction. Prior work shows that DeepT is more precise than CROWN-BaF but consumes more time. However, they share a common limitation, i.e., neglecting dependencies among the inputs of softmax in linear relaxations, which leads to imprecise results of robustness certification. In contrast, GaLileo considers the above dependencies and overcomes the trade-off between precision and efficiency in robustness certification, which, in our humble opinion, is a more meaningful direction.

Background

Transformer Architecture

The Transformer architecture in this work is shown in Fig. 1. The model processes an input sequence consisting of N tokens through itself and associates the input with a label.

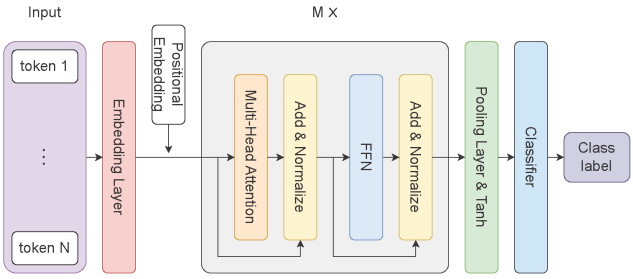

Figure 1: The architecture of Transformers used in this work.

The key component in a Transformer is its *self-attention mechanism*. Let $x^1, \ldots, x^N \in \mathbb{R}^E$ be the inputs of the self-attention function. For simplicity, we transpose and stack them into the matrix $X \in \mathbb{R}^{N \times E}$. The self-attention function first multiplies X with three separate matrices to obtain the queries Q, the keys K, and the values V. Then, the output $Z \in \mathbb{R}^{N \times E}$ of the self-attention function is obtained by

$$Z = \sigma\left(\frac{QK^\mathsf{T}}{\sqrt{d_k}}\right)V = \sigma\left(\frac{XW_Q W_K^\mathsf{T} X^\mathsf{T}}{\sqrt{d_k}}\right)XW_V, \quad (1)$$

where $W_Q, W_K \in \mathbb{R}^{E \times d_k}$ and $W_V \in \mathbb{R}^{E \times d_v}$. Next, the softmax function $\sigma : \mathbb{R}^N \mapsto \mathbb{R}^N$ is applied to every row:

$$\sigma_i(v_1, \ldots, v_N) = \frac{e^{v_i}}{\sum_{j=1}^N e^{v_j}} = \frac{1}{\sum_{j=1}^N e^{v_j - v_i}}, \quad (2)$$

where σ_i is the i-th component of σ. Finally, the rows of Z are returned as output embeddings of the self-attention layer.

Practical models usually adopt *multi-head self-attention*, where multiple self-attentions, which are referred to as attention heads, are combined in one layer for better performance. Specifically, the input X is first fed to each attention head and then their outputs are stacked as a matrix. Next, the matrix is multiplied with $W_o \in \mathbb{R}^{(n_a d_v) \times E}$, where n_a is the number of attention heads, resulting in $Z \in \mathbb{R}^{N \times E}$. Similar to self-attention, the rows of Z are returned.

We briefly introduce the rest structures. The normalization layer is different from the original. Following the previous works (Shi et al. 2020; Bonaert et al. 2021), we remove the division by the standard deviation for improving certification rates without significantly affecting the performance. The Feed-Forward Network (FFN) maps $\mathbb{R}^E$ to $\mathbb{R}^E$ for each embedding, which consists of one hidden ReLU layer of size H. The pooling layer picks the first output embedding and disregards the others, followed by a linear classifier.

CROWN-like Certification

Definition 1 (Robustness Certification Problem)**.** Given a model f and a neighborhood $\mathcal{B}_{p,\epsilon}(x_0)$ around a clean input x_0, the robustness certification problem is to verify whether the below condition holds.

$$\arg\max_i f(x)_i = \arg\max_i f(x_0)_i, \forall x \in \mathcal{B}_{p,\epsilon}(x_0), \quad (3)$$

where $\mathcal{B}_{p,\epsilon}(x_0) = \{x : \|x - x_0\|_p \leq \epsilon\}$.

The maximum ϵ that makes Eq. 3 hold is called the *certified radius*, which can be calculated by binary search.

Here we provide a concise overview of how CROWN-like methods solve the above problem. These methods propagate two linear bounds (a lower bound and an upper bound) through f layer by layer to derive the numerical bounds of each neuron when x is perturbed within $\mathcal{B}_{p,\epsilon}(x_0)$. Let $z_{i,j}$ be the j-th neuron (n_i in total) in the i-th layer. They calculate the lower bound $z_{i,j}^L$ and the upper bound $z_{i,j}^U$ of each neuron $z_{i,j}$ by propagating linear bounds from the first layer to the last layer. Specifically, for each neuron in the i-th layer, they compute the bounds as linear functions of previous neurons (as shown in Eq. 4). Thus, they can calculate linear bounds between any two layers ($k < i$) by propagating the bounds.

$$A_{j,:}^{k,i,L} z_k + B_j^{k,i,L} \leq z_{i,j} \leq A_{j,:}^{k,i,U} z_k + B_j^{k,i,U}, \quad (4)$$

where $A^{k,i,L/U} \in \mathbb{R}^{n_i \times n_k}$ and $B^{k,i,L/U} \in \mathbb{R}^{n_i}$ are parameters of lower and upper bounds.

CROWN-like methods propagate the bounds in either the *backward mode* or the *forward mode*. The backward mode propagates the bounds to previous layers by substituting $z_{i,j}$ with linear functions of its previous neurons. This process can be recursively conducted until the input layer. The forward mode propagates the bounds to the next layers by calculating the bounds of the $(i+1)$-th layer as linear functions of neurons in the previous layer according to the operations in the $(i+1)$-th layer and linear bounds of the i-th layer. Prior work (Shi et al. 2020) have shown that the forward mode is faster and looser than the backward mode. After deriving the linear bounds between the i-th layer and the input layer, the method summarizes the above bounds to calculate the numerical bounds of neurons in the i-th layer when tokens in $\mathcal{P}$ are perturbed within $\mathcal{B}_{p,\epsilon}(x_0^{p_k})$. Note that $1/p + 1/q = 1$ with $p, q \geq 1$.

$$z_{i,j}^L = -\epsilon \|A_{j,:}^{0,i,L}\|_q + A_{j,:}^{0,i,L} x_0 + B_j^{0,i,L}, \quad (5)$$

$$z_{i,j}^U = \epsilon \|A_{j,:}^{0,i,U}\|_q + A_{j,:}^{0,i,U} x_0 + B_j^{0,i,U}. \quad (6)$$

To accelerate robustness certifications of Transformers, CROWN-BaF (Shi et al. 2020) combines the backward mode with the forward mode, which is shown to be able to achieve a better balance between tightness and efficiency than using the fully backward mode or the fully forward mode. Specifically, Shi et al. use the backward mode to handle affine transformations and unary nonlinear functions while using the forward mode to handle the challenging operations in self-attention, including softmax functions.

Softmax Relaxation. Shi et al. conduct the relaxation by treating softmax functions as compositions of non-linear functions (as shown in Eq. 2) and extending existing relaxations to each non-linear function. Assume that the linear bounds have been propagated to the layer before softmax. The aim is to calculate linear bounds after softmax functions using the forward mode. First, linear relaxations for exponential functions are computed (as shown in Eq. 7) and linear bounds of e^{v_i} and $\sum_{j=1}^N e^{v_j}$ are derived.

$$y^L = a^L x + b^L \leq e^x \leq y^U = a^U x + b^U. \quad (7)$$

Then, relaxations for reciprocal functions are computed (similar to the above) and linear bounds for the reciprocal of $\sum_{j=1}^{N} e^{v_j}$ are calculated. Finally, relaxations for multiplications are computed (as shown in the following inequality) and linear bounds of softmax functions are derived.

$$y^L = a_1^L x_1 + a_2^L x_2 + b^L \leq x_1 x_2 \leq y^U = a_1^U x_1 + a_2^U x_2 + b^U.$$

Methodology

Overview. In this work, we adopt a procedure akin to that of CROWN-like methods. Specifically, we propagate two linear bounds through the model layer by layer and derive numerical bounds of the outputs according to Eq. 5. Our key observation is that the linear relaxation of softmax that was used in previous works (Shi et al. 2020) neglects dependencies among the inputs of softmax, which causes imprecise certification results of Transformers' robustness. To harness the dependencies among inputs and enhance the precision of robustness certification, it becomes necessary to address the linear relaxation of softmax within an n-dimensional space. This involves solving the following problem.

Problem Formulation. This study is focused on resolving the n-dimensional (n-d, for brevity) linear relaxation problem, which is presented as follows:

Definition 2 (n-dimensional Linear Relaxation Problem)**.** Given an n-d function $y = f(x_1, x_2, \ldots, x_n)$ where $x_i \in [l_i, u_i] (i = 1, 2, \ldots, n)$, its linear relaxation problem is to calculate a pair of linear functions, i.e., a lower bound y^L and an upper bound y^U, that satisfy the following condition.

$$y^L = a_1^L x_1 + \cdots + a_n^L x_n + b^L \leq f(x_1, x_2, \ldots, x_n) \leq y^U = a_1^U x_1 + \cdots + a_n^U x_n + b^U,\ \forall x_i \in [l_i, u_i].$$

Key Challenge. The key challenge in addressing the above problem lies in contending with the curse of dimensionality. For a unary function such as sigmoid or tanh (let l and u be the lower and upper bound of its input), since it is convex within $(-\infty, 0)$ and concave within $(0, \infty)$, its relaxation (Zhang et al. 2018) requires considerations of 3 cases of bounds: $l \leq u \leq 0$ (convex), $0 \leq l \leq u$ (concave), and $l < 0 < u$ (neither convex nor concave). For complex functions such as $y = \text{sigmoid}(x_1) \odot \tanh(x_2)$ in LSTM models, their relaxations (Du et al. 2021) need to consider numerical bounds of both inputs, which can result in 3^2 cases at most (3 cases of $x_1 \times 3$ cases of x_2). Thus, for softmax that is convex within half of its domain and concave within the other half on each v_j (Eq. 2), a naive extension of above relaxation need to consider 3^n cases at most. As n grows larger, computing such relaxation becomes increasingly unattainable. Here n is the number of words in an input sentence.

n-dimensional Linear Relaxation

Preliminaries. We introduce generalized monotonic functions, which play an important role in this work.

Definition 3 (Generalized Monotonic Function)**.** We call an n-dimensional function f a generalized monotonic function if it increases monotonically with each of its input variables increasing or decreasing respectively. The following condition corresponds to the increasing variables. The condition for the decreasing variables is similar.

$$f(x_1, \ldots, x_i, \ldots, x_n) \leq f(x_1, \ldots, x_i', \ldots, x_n)$$
$$\text{if } x_i \leq x_i', \forall i \in [n], \forall\, x_j \in [l_j, u_j], j \in [n], j \neq i.$$

According to the above definition, the softmax function (Eq. 2) is generalized monotonic function. $[n] = \{1, \ldots, n\}$.

Without loss of generality, here we assume that the function f increases monotonically with each of its input variables (i.e., x_is) increasing. For variables with which decreasing f increases monotonically, we can flip their signs without changing the nonlinearity. For example, we can flip the signs of a softmax function's certain inputs and obtain a symmetrical function $y_i' = \sigma_i(-x_1, \ldots, x_i, \ldots, -x_n)$ that is non-decreasing on each of its inputs. We will provide details of how to compute its linear relaxation in the following. The linear relaxation of softmax follows a similar approach.

Numerical Bounds. Calculating the linear bounds of a function f requires its minimum f^L and maximum f^U, which can be computed according to the following theorem.

Theorem 1. *An n-dimensional bounded generalized monotonic function f reaches its minimum and maximum at $\vec{l}$ and $\vec{u}$ respectively.*

$$f^L = f(\vec{l}) \leq f(\vec{x}) \leq f(\vec{u}) = f^U,\ \forall x_i \in [l_i, u_i],\ \forall i \in [n],$$
$$\vec{x} = (x_1, \ldots, x_n), \vec{l} = (l_1, \ldots, l_n), \vec{u} = (u_1, \ldots, u_n).$$

Similarly, for y_i', we have the below inequality, where $y_i'(\vec{l}), y_i'(\vec{u}) \in \mathbb{R}$ denote its minimum and maximum.

$$y_i'(\vec{l}) \leq y_i'(x_1, \ldots, x_n) \leq y_i'(\vec{u}), \forall x_j \in [l_j, u_j], \forall j \in [n].$$

Linear Bounds. The linear relaxation of f is calculated according to the following theorem. Let $\hat{x}_j$ denote the input dimensions except x_j and $f(x_j; \hat{x}_j) = f(\vec{x})$.

Theorem 2. *Given an n-dimensional bounded generalized monotonic function f, its linear bounds y^L and y^U can be computed according to the following equations.*

$$y^L(\vec{x}) \leq f(\vec{x}) \leq y^U(\vec{x}),\ \forall x_j \in [l_j, u_j], \forall j \in [n],\ \textit{where}$$
$$y^L(\vec{x}) = \sum_{j=1}^{n} a_j^L x_j + b^L,\ y^U(\vec{x}) = \sum_{j=1}^{n} a_j^U x_j + b^U, \quad (8)$$
$$a_j^L = \min \left\{ \frac{\mathrm{d} f(x_j; \hat{l}_j)}{\mathrm{d} x_j} : \forall x_j \in [l_j, u_j] \right\}, \quad (9)$$
$$a_j^U = \min \left\{ \frac{\mathrm{d} f(x_j; \hat{u}_j)}{\mathrm{d} x_j} : \forall x_j \in [l_j, u_j] \right\}, \quad (10)$$
$$b^L = f(\vec{l}) - \sum_{j=1}^{n} a_j^L l_j,\ b^U = f(\vec{u}) - \sum_{j=1}^{n} a_j^U u_j. \quad (11)$$

Theorem 2 gives linear bounds for generalized monotonic functions. Furthermore, we find tighter bounds for the specific function y_i' and calculate its a_j^L and a_j^U according to

the following equations. The remainder of y_i''s linear bounds follow the same formulation as described above.

$$a_j^L = \min\left\{ \left.\frac{\partial y_i'}{\partial x_j}\right|_{\vec{x}=\vec{l}}, \frac{y_i'(u_j;\hat{l}_j) - y_i'(l_j;\hat{l}_j)}{u_j - l_j} \right\}, \quad (12)$$

$$a_j^U = \min\left\{ \left.\frac{\partial y_i'}{\partial x_j}\right|_{\vec{x}=\vec{u}}, \frac{y_i'(u_j;\hat{u}_j) - y_i'(l_j;\hat{u}_j)}{u_j - l_j} \right\}. \quad (13)$$

As shown in the above equations, our relaxation for the function y_i' (and softmax) decouples its nonlinearity in each dimension, which allows us to calculate a_j^Ls or a_j^Us separately and avoids the exponential number of cases.

Solution to the curse of dimensionality. By decoupling the nonlinearity of softmax in each dimension of its input, we reduce the number of cases to only $2n$. This is significantly fewer than the impractical 3^n cases encountered in the naive extension of classic relaxations that consider input dependencies. As a result, we have successfully overcome the curse of dimensionality.

Tightness. CROWN-BaF (Shi et al. 2020) decomposes the softmax into simple functions and relaxes them sequentially. Errors generated in earlier relaxations adversely affect subsequent stages (resulting in looser 'pre-relaxation' bounds). To circumvent these negative impacts, we adopt one-stage relaxations that treat softmax as an n-d function without decomposition, which allows us to achieve tighter bounds.

Soundness. Soundness is crucial to linear relaxations in robustness certification because unsound relaxations (e.g., f exceeds its lower or upper bounds at certain points) lead to incorrect results. We confirm the soundness of our relaxation by providing the proof of that the inequality in Theorem 2 holds true. Due to the limited space, only the proof of the lower bound (i.e., $f(\vec{x}) - y^L(\vec{x}) \geq 0$) is presented. The proof for the upper bound follows a similar procedure. Note that we set $y^L(\vec{l}) = f(\vec{l})$ and $y^U(\vec{u}) = f(\vec{u})$.

We start with the following lemma, which converts the n-dimensional problem into several 1-dimensional problems by decoupling its input dimensions.

Lemma 1. *$f(\vec{x}) - y^L(\vec{x}) \geq 0$ holds true for any $x_j \in [l_j, u_j]$ if $\int_{l_j}^{x_j} \left(\frac{\partial f}{\partial v_j} - a_j^L\right) \mathrm{d}v_j \geq 0$ holds true for any $j \in [n]$.*

Proof. According to the fundamental theorem of line integrals, we have the following equations, where the integral is independent of the path between the endpoints.

$$\begin{aligned} & f(\vec{x}) - y^L(\vec{x}) \\ &= f(\vec{l}) + \int_{\vec{l}}^{\vec{x}} \sum_{j=1}^{n} \frac{\partial f}{\partial v_j} \mathrm{d}v_j - y^L(\vec{l}) - \int_{\vec{l}}^{\vec{x}} \sum_{j=1}^{n} \frac{\partial y^L}{\partial v_j} \mathrm{d}v_j \\ &= \sum_{j=1}^{n} \int_{l_j}^{x_j} \left(\frac{\partial f}{\partial v_j} - a_j^L\right) \mathrm{d}v_j \end{aligned}$$ □

Proving Lemma 1 involves the proof of the lemma below.

Lemma 2. *An integral $\int_{l_j}^{x_j} \left(\frac{\partial f}{\partial v_j} - a_j^L\right) \mathrm{d}v_j$ is non-negative if the integral $\int_{l_j}^{x_j} \left(\frac{\mathrm{d}f(v_j;\hat{l}_j)}{\mathrm{d}v_j} - a_j^L\right) \mathrm{d}v_j$ is non-negative.*

Proof. Since f is a generalized monotonic function, we have $f(x_j;\hat{x}_j) \geq f(x_j;\hat{l}_j)$. Then, according to the fundamental theorem of integrals, we have the following equations.

$$\begin{aligned} & \int_{l_j}^{x_j} \left(\frac{\partial f}{\partial v_j} - a_j^L\right) \mathrm{d}v_j \\ &= f(\vec{l}) + \int_{l_j}^{x_j} \frac{\partial f}{\partial v_j} \mathrm{d}v_j - y^L(\vec{l}) - \int_{l_j}^{x_j} a_j^L \mathrm{d}v_j \\ &= f(x_j;\hat{x}_j) - y^L(\vec{l}) - \int_{l_j}^{x_j} a_j^L \mathrm{d}v_j \\ &\geq f(x_j;\hat{l}_j) - y^L(\vec{l}) - \int_{l_j}^{x_j} a_j^L \mathrm{d}v_j \\ &= \int_{l_j}^{x_j} \left(\frac{\mathrm{d}f(v_j;\hat{l}_j)}{\mathrm{d}v_j} - a_j^L\right) \mathrm{d}v_j \end{aligned}$$ □

Finally, according to Eq. 9, we have $\mathrm{d}f(v_j;\hat{l}_j)/\mathrm{d}v_j - a_j^L \geq 0$ for any $v_j \in [l_j, u_j]$. Hence, by Lemma 1 and Lemma 2, we have $f(\vec{x}) - y^L(\vec{x}) \geq 0$. Q.E.D. The soundness proof of y_i''s linear bounds is provided in the Appendix.

Experimental Evaluation

Overview. In this section, we evaluate the effectiveness of GaLileo by comparing it to CROWN-BaF. We conducted comprehensive experiments on various Transformers with different depths and widths trained on the SST and Yelp datasets. We compared the certified radius calculated by the methods and their time cost. All experiments were conducted on a Linux server with two Intel Xeon Silver 4210R CPUs running at 2.40 GHz, 128 GB memory, 4TB HDD, and a GeForce RTX 2080 Ti GPU card.

Benchmarks and Metrics. We trained various models with different depths ($M = 3$, 6, and 12 layers) and widths (hidden size $E = 128$, 256, 384, and 512) on two widely used datasets, i.e., the SST dataset (Socher et al. 2013) and the Yelp dataset (Zhang, Zhao, and LeCun 2015). All models in this evaluation use 4 attention heads and FFNs with 128 hidden neurons. We also consider the cases with different types of norms including ℓ_1, ℓ_2, and ℓ_∞.

We compare GaLileo and CROWN-BaF in terms of the following quantities. *Certified Radius.* We randomly choose 10 correctly classified test examples with sentence lengths less than 32 and compute the maximum robustness radius around the embedding of each word in them with binary search. *Running Time.* We record the average running time of computing the maximum robustness radius for each word.

Experiment I. In the first experiment, we compare the precision and efficiency of GaLileo and CROWN-BaF on Transformers with different depths and widths. We present their results in Table 1. The scale of the Transformers ranges from 3 layers to 12 layers and hidden sizes are 128 to 512.

First, GaLileo exhibits a higher level of precision compared to CROWN-BaF. As shown in Table 1, GaLileo achieves up to 3.24 times larger certified radii than CROWN-BaF. The distinct advantage of GaLileo stems from the utilization of our n-d relaxation, which effectively incorporates

Dataset	Model		Average Certified Radius						Ratio			Time (s)					
			CROWN-BaF			GaLileo						CROWN-BaF			GaLileo		
	Depth	Width	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞
SST	3	128	1.686	0.328	0.033	**1.749**	**0.337**	**0.034**	1.04	1.03	1.03	5.8	5.9	5.6	5.8	5.8	5.8
		256	1.309	0.272	0.028	**1.353**	**0.279**	**0.028**	1.03	1.03	1.00	5.7	5.5	5.5	5.7	5.6	5.7
		384	0.994	0.210	0.022	**1.072**	**0.225**	**0.023**	1.08	1.07	1.05	5.7	5.9	6.0	6.1	5.9	5.9
		512	1.048	0.219	0.022	**1.123**	**0.233**	**0.024**	1.07	1.06	1.09	6.4	6.5	6.5	6.5	6.5	6.4
	6	128	0.470	0.083	8.0e-3	**0.620**	**0.111**	**0.011**	1.32	1.34	1.38	16.9	15.9	16.4	16.3	16.0	15.8
		256	0.360	0.070	6.8e-3	**0.438**	**0.086**	**8.4e-3**	1.22	1.23	1.24	16.0	16.3	15.4	16.1	16.1	15.7
		384	0.227	0.046	4.5e-3	**0.280**	**0.057**	**5.6e-3**	1.23	1.24	1.24	17.4	17.8	17.7	17.6	17.4	17.9
		512	0.310	0.060	5.9e-3	**0.361**	**0.070**	**7.0e-3**	1.16	1.17	1.19	19.7	19.2	20.0	19.2	19.6	19.6
	12	128	0.018	3.7e-3	3.5e-4	**0.023**	**4.7e-3**	**4.6e-4**	1.28	1.27	1.31	53.9	53.8	54.0	54.3	52.5	51.8
		256	0.020	4.2e-3	4.1e-4	**0.024**	**4.9e-3**	**4.8e-4**	1.20	1.17	1.17	55.2	53.3	52.2	53.5	56.7	55.3
		384	0.022	4.0e-3	3.9e-4	**0.026**	**4.8e-3**	**4.7e-4**	1.18	1.20	1.21	62.7	62.4	64.5	61.5	61.7	60.2
		512	0.043	9.0e-3	9.3e-4	**0.050**	**0.011**	**1.1e-3**	1.16	1.22	1.18	71.1	71.4	74.1	70.5	72.3	71.8
Yelp	3	128	0.573	0.137	0.015	**0.755**	**0.181**	**0.020**	1.32	1.32	1.33	5.5	5.6	5.6	5.4	5.5	5.5
		256	0.468	0.125	0.014	**0.546**	**0.148**	**0.017**	1.17	1.18	1.21	5.4	5.4	6.0	5.5	5.6	5.9
		384	0.528	0.137	0.015	**0.622**	**0.160**	**0.018**	1.18	1.17	1.20	6.4	6.8	6.6	5.8	6.0	6.2
		512	0.462	0.118	0.013	**0.522**	**0.133**	**0.015**	1.13	1.13	1.15	6.3	6.3	7.5	6.2	6.7	6.6
	6	128	0.022	5.6e-3	5.8e-4	**0.034**	**8.1e-3**	**8.5e-4**	1.54	1.45	1.47	28.8	26.8	23.0	20.8	22.2	21.8
		256	0.023	5.4e-3	5.8e-4	**0.051**	**0.012**	**1.3e-3**	2.22	2.22	2.24	22.6	23.7	21.7	20.5	20.4	20.5
		384	0.016	4.0e-3	4.3e-4	**0.027**	**6.6e-3**	**7.2e-4**	1.69	1.65	1.67	29.6	27.3	22.8	26.6	26.4	24.2
		512	0.023	7.4e-3	8.2e-4	**0.067**	**0.014**	**1.3e-3**	2.91	1.89	1.59	33.0	31.3	26.2	31.1	28.6	27.9

Table 1: Average certified radius and running time by CROWN-BaF and GaLileo.

input dependencies and results in a tighter relaxation compared to those employed in CROWN-BaF. Second, GaLileo maintains a similar level of efficiency to that of CROWN-BaF. As shown in Table 1, the execution time of GaLileo closely approximates that of CROWN-BaF. To summarize, GaLileo successfully overcomes the trade-off between precision and efficiency in the realm of robustness certification. Thus, we firmly contend that GaLileo represents a superior option for robustness certification of Transformers, particularly when dealing with larger models.

Experiment II. In the second experiment, we proceed to evaluate the robustness of Transformers against multi-word ℓ_p perturbations. Prior works focus on 1 or 2-word perturbations, which are challenged by more powerful attackers capable of perturbing multiple words within the target sentence. We have observed that in most multi-word attacks, attackers often target specific vulnerable words for perturbation, rather than opting for random word substitutions. In light of this, we undertake the certification of Transformers' robustness by simulating scenarios where words with the smallest certified radii within a sentence are perturbed simultaneously. The results are presented in Table 3, where the models have 3 layers and hidden size $E = 128$.

The results reveal a consistent trend: as the number of perturbed words increases, the certified radii exhibit a noticeable decrease. The results indicate that models are more susceptible to alteration when more words are perturbed simultaneously.

Experiment III. In the third experiment, we further evaluate the impact of the n-d relaxation by comparing the precision and efficiency of GaLileo and CROWN-BaF on input sequences with different lengths, where n is the sequence length. The results are shown in Table 2, where the models are trained on the Yelp dataset and hidden size $E = 128$. We set 3 ranges of sequence lengths. For each range, we randomly choose 10 correctly classified test examples with lengths within the range to calculate their certified radii.

As indicated by the results, irrespective of the lengths of input sequences, GaLileo consistently demonstrates a higher level of precision compared to CROWN-BaF and exhibits comparable efficiency to that of CROWN-BaF. To summarize, the results affirm that GaLileo's advantage remains consistent across varying sequence lengths.

Experiment IV. In this experiment, we integrate the n-dimensional linear relaxation with Zonotopes and conduct a comparative analysis against DeepT (Bonaert et al. 2021). The models are trained on the Yelp dataset with $E = 128$.

The results, as illustrated in Table 4, reveal that GaLileo (Zonotope variant) attains larger certified radii in many cases, although not universally. Notably, Zonotope-based methods (e.g., DeepT) necessitate parallel bounds in linear relaxations, distinguishing them from CROWN-like methods (e.g., CROWN-BaF). Consequently, the straightforward extension of the n-d relaxation to Zonotope-based methods does not guarantee the attainment of tighter bounds. We are planning to find tighter bounds for Zonotopes in the future.

Depth	length	Average Certified Radius						Ratio			Time (s)					
		CROWN-BaF			GaLileo						CROWN-BaF			GaLileo		
		ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞
3	$(8, 16]$	0.561	0.134	0.015	**0.745**	**0.178**	**0.019**	1.33	1.33	1.27	5.6	5.4	6.0	5.7	5.6	5.9
	$(16, 24]$	0.544	0.128	0.014	**0.702**	**0.167**	**0.018**	1.29	1.30	1.29	5.3	5.3	5.4	5.6	5.6	5.6
	$(24, 32]$	0.665	0.157	0.017	**0.920**	**0.218**	**0.024**	1.38	1.39	1.41	5.7	5.7	5.5	5.7	5.8	6.0
6	$(8, 16]$	0.018	4.7e-3	4.9e-4	**0.029**	**6.9e-3**	**7.2e-4**	1.61	1.47	1.47	28.2	24.8	22.0	22.8	22.5	22.7
	$(16, 24]$	0.032	7.9e-3	8.3e-4	**0.036**	**9.7e-3**	**1.0e-3**	1.13	1.23	1.20	16.9	17.4	16.5	15.4	15.6	15.3
	$(24, 32]$	0.043	0.010	1.1e-3	**0.059**	**0.014**	**1.5e-3**	1.37	1.40	1.36	15.8	15.9	15.8	17.2	17.3	18.1

Table 2: Evaluation results of input sequences with different lengths.

Perturbed Words	ℓ_p	SST		Yelp	
		C.R.	Time (s)	C.R.	Time (s)
2	ℓ_1	0.796	5.7	0.294	5.8
	ℓ_2	0.149	6.1	0.070	5.7
	ℓ_∞	0.015	6.0	7.6e-3	5.8
3	ℓ_1	0.290	6.7	0.120	6.6
	ℓ_2	0.073	6.8	0.038	6.3
	ℓ_∞	0.010	6.4	5.8e-3	6.3
4	ℓ_1	0.216	7.0	0.099	7.2
	ℓ_2	0.054	7.5	0.030	6.6
	ℓ_∞	0.008	6.3	4.6e-3	6.6
5	ℓ_1	0.186	7.3	0.061	7.6
	ℓ_2	0.047	7.4	0.022	7.1
	ℓ_∞	0.006	6.0	4.0e-3	6.8

Table 3: Average certified radius (C.R.) and running time against multi-word perturbations.

Experiment V. In the final experiment, we extend GaLileo to assess the robustness of Vision Transformers (ViTs). We trained three ViTs on the MNIST dataset, where the hidden size E is set to 64 and the FFNs have 64 hidden neurons each. Subsequently, we randomly selected 100 correctly classified test images and computed the average certified radii using the ℓ_p threat model in computer vision. The results, including both the average certified radii and the corresponding running times, are presented in Table 5.

Conclusion

In this paper, we introduce GaLileo, a pioneering robust certification method that leverages our n-dimensional linear relaxation to establish security assurances for Transformers against adversarial attacks. Through the application of the n-dimensional linear relaxation approach, GaLileo successfully reconciles the age-old trade-off between precision and efficiency in the field of robustness certification. This approach capitalizes on the inherent dependencies among inputs of softmax functions to tighten their linear bounds, all while circumventing the curse of dimensionality by decoupling input dimensions. Furthermore, we substantiate the soundness and efficiency of the n-dimensional linear relaxation approach through rigorous theoretical validation. Our comprehensive evaluation assesses GaLileo's performance across diverse models and two widely used datasets, benchmarking it against the well-established CROWN-BaF method. Experimental results unequivocally demonstrate that GaLileo achieves higher precision than CROWN-BaF in certifying the robustness of Transformers, while maintaining a similar level of computational efficiency. These findings provide solid validation for our assertions. Moreover, GaLileo pioneers the certification of Transformers' robustness against multi-word (no less than three words) ℓ_p perturbations for the first time, extending the scope of our contributions and enhancing the understanding of Transformers' robustness.

Depth	ℓ_p	DeepT		GaLileo+Zono		Ratio
		C.R.	Time	C.R.	Time	
3	ℓ_1	0.795	24.1	**0.863**	12.1	1.09
	ℓ_2	0.194	24.1	**0.216**	12.0	1.11
	ℓ_∞	0.021	24.2	**0.023**	11.6	1.10
6	ℓ_1	0.243	48.7	**0.249**	24.5	1.02
	ℓ_2	**0.088**	48.4	0.078	26.7	0.89
	ℓ_∞	**0.011**	49.9	0.009	26.4	0.82
12	ℓ_1	0.088	92.3	**0.205**	52.5	2.33
	ℓ_2	0.025	112.4	**0.068**	50.9	2.72
	ℓ_∞	2.8e-3	111.5	**8.7e-3**	52.3	3.11

Table 4: Evaluation results on the Yelp dataset.

Depth	C.R.			Time(s)		
	ℓ_1	ℓ_2	ℓ_∞	ℓ_1	ℓ_2	ℓ_∞
2	9.6e-3	3.7e-3	6.8e-4	6.1	6.9	7.8
3	4.8e-4	1.7e-4	3.2e-5	13.5	14.3	16.4
4	8.8e-5	2.9e-5	9.8e-6	21.8	23.4	24.6

Table 5: Evaluation results of Vision Transformers.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (under Grant No.62372268), Shandong Provincial Natural Science Foundation (under Grant No.ZR2021LZH007, No.ZR2020MF055, and No.ZR2022LZH013), and Jinan City "20 New Universities" Funding Project (under Grant No.2021GXRC084).

References

Bonaert, G.; Dimitrov, D. I.; Baader, M.; and Vechev, M. T. 2021. Fast and precise certification of transformers. In *ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 466–481.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Conference on Neural Information Processing Systems*.

Bunel, R.; Turkaslan, I.; Torr, P. H. S.; Kohli, P.; and Mudigonda, P. K. 2018. A Unified View of Piecewise Linear Neural Network Verification. In *Conference on Neural Information Processing Systems*, 4795–4804.

Carlini, N.; and Wagner, D. A. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy*, 39–57.

Cheng, C.; Nührenberg, G.; and Ruess, H. 2017. Maximum Resilience of Artificial Neural Networks. In *International Symposium on Automated Technology for Verification and Analysis*, volume 10482, 251–268.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 4171–4186.

Dong, L.; Xu, S.; and Xu, B. 2018. Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 5884–5888.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Du, T.; Ji, S.; Shen, L.; Zhang, Y.; Li, J.; Shi, J.; Fang, C.; Yin, J.; Beyah, R.; and Wang, T. 2021. Cert-RNN: Towards Certifying the Robustness of Recurrent Neural Networks. In *ACM Conference on Computer and Communications Security*, 516–534.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

Guo, C.; Sablayrolles, A.; Jégou, H.; and Kiela, D. 2021. Gradient-based Adversarial Attacks against Text Transformers. In *Conference on Empirical Methods in Natural Language Processing*, 5747–5757.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–778.

Joshi, A.; Jagatap, G.; and Hegde, C. 2021. Adversarial Token Attacks on Vision Transformers. arXiv:2110.04337.

Katz, G.; Barrett, C. W.; Dill, D. L.; Julian, K.; and Kochenderfer, M. J. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *International Conference on Computer Aided Verification*, volume 10426, 97–117.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

OpenAI. 2022. Chat-GPT. https://chat.openai.com. Accessed: 2023-03-14.

Papernot, N.; McDaniel, P. D.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016a. The Limitations of Deep Learning in Adversarial Settings. In *IEEE European Symposium on Security and Privacy*, 372–387.

Papernot, N.; McDaniel, P. D.; Wu, X.; Jha, S.; and Swami, A. 2016b. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. In *IEEE Symposium on Security and Privacy*, 582–597.

Prakash, A.; Chitta, K.; and Geiger, A. 2021. Multi-Modal Fusion Transformer for End-to-End Autonomous Driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7077–7087.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018a. Certified Defenses against Adversarial Examples. In *International Conference on Learning Representations*.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018b. Semidefinite relaxations for certifying robustness to adversarial examples. In *Conference on Neural Information Processing Systems*, 10900–10910.

Salman, H.; Yang, G.; Zhang, H.; Hsieh, C.; and Zhang, P. 2019. A Convex Relaxation Barrier to Tight Robustness Verification of Neural Networks. In *Conference on Neural Information Processing Systems*, 9832–9842.

Shi, Z.; Zhang, H.; Chang, K.; Huang, M.; and Hsieh, C. 2020. Robustness Verification for Transformers. In *International Conference on Learning Representations*.

Singh, G.; Ganvir, R.; Püschel, M.; and Vechev, M. T. 2019. Beyond the Single Neuron Convex Barrier for Neural Network Certification. In *Conference on Neural Information Processing Systems*, 15072–15083.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Conference on Empirical Methods in Natural Language Processing*, 1631–1642.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Tjandraatmadja, C.; Anderson, R.; Huchette, J.; Ma, W.; Patel, K.; and Vielma, J. P. 2020. The Convex Relaxation Barrier, Revisited: Tightened Single-Neuron Relaxations for Neural Network Verification. In *Conference on Neural Information Processing Systems*.

Tran, C.; Vu, A. N.; and Nguyen, V. 2022. Baby Learning with Vision Transformer for Face Recognition. In *International Conference on Multimedia Analysis and Pattern Recognition*, 1–6.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Conference on Neural Information Processing Systems*, 5998–6008.

Weng, T.; Zhang, H.; Chen, H.; Song, Z.; Hsieh, C.; Daniel, L.; Boning, D. S.; and Dhillon, I. S. 2018. Towards Fast Computation of Certified Robustness for ReLU Networks. In *International Conference on Machine Learning*, volume 80, 5273–5282.

Zhang, H.; Weng, T.; Chen, P.; Hsieh, C.; and Daniel, L. 2018. Efficient Neural Network Robustness Certification with General Activation Functions. In *Conference on Neural Information Processing Systems*, 4944–4953.

Zhang, X.; Zhao, J. J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In *Conference on Neural Information Processing Systems*, 649–657.

Zhang, Z.; Wu, Y.; Liu, S.; Liu, J.; and Zhang, M. 2022. Provably Tightest Linear Approximation for Robustness Verification of Sigmoid-like Neural Networks. In *IEEE/ACM International Conference on Automated Software Engineering*, 80:1–80:13.