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

Tianyu Du Shouling Ji Lujia Shen Yao Zhang Jinfeng Li
Jie Shi Chengfang Fang Jianwei Yin Raheem Beyah Ting Wang

Background

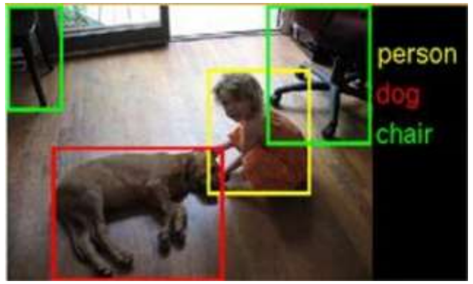
Speech Recognition

Sentiment Analysis

Face Recognition

Deep Learning for Multiple Tasks

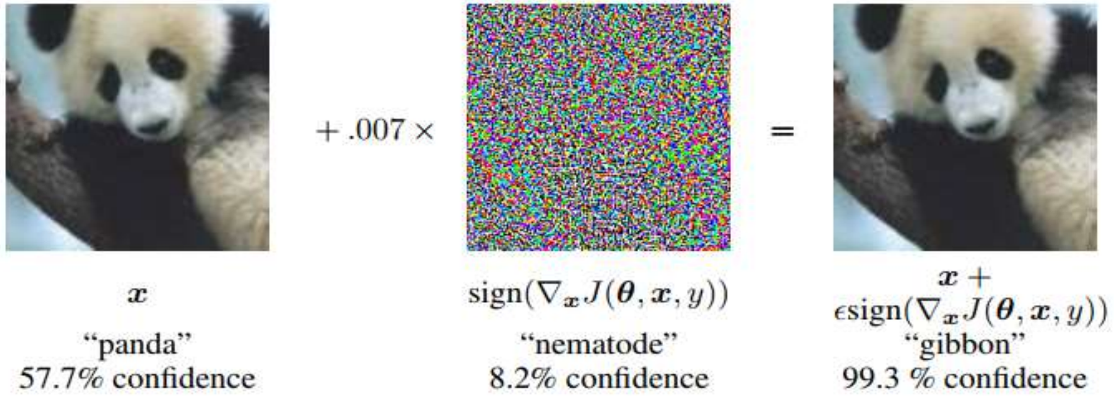
Object Detection

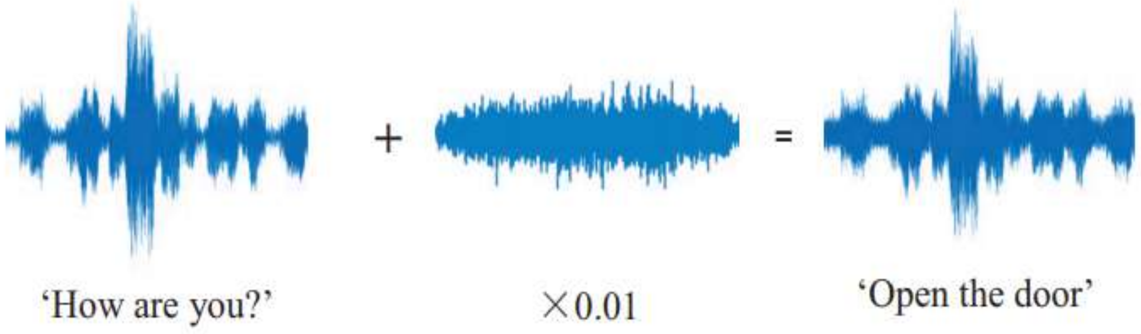


Political Content Detection

Breaking Thing Is Easy

- Adversarial examples exist **EVERYWHERE**!





Task: Toxic Content Detection. ***Classifier:*** LSTM. ***Original label:*** 96.7% Toxic. ***Adversarial label:*** 83.5% Non-toxic.

Text: hello how are you? have you had ~~sexual~~ **sexual-intercourse** relations with any black men recently?

Task: Toxic Content Detection. ***Classifier:*** Perspective. ***Original label:*** 92% Toxic. ***Adversarial label:*** 78% Non-toxic.

Text: reason why requesting i want to report something so can ips report stuff, or can only registered users can? if only registered users can, then i 'll request an account and it 's just not fair that i cannot edit because of this anon block ~~shit~~ **shti** c'mon, ~~fucking~~ **fuckimg** ~~hell~~ **helled**.

Defense against Adversarial Attacks

- ## Empirical Defense ☹
  - Example: Adversarial training [Madry et al. 2017]
  - Work empirically but no theoretical guarantee
  - Attack specific – leading to an arms race that attackers are winning

ICML 2018

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

Anish Athalye *[1] Nicholas Carlini *[2] David Wagner [2]

Key Questions

- Guaranteed accuracy: what is the minimum accuracy under any attack?
- Prediction robustness: given a prediction, can any attack change it?

- ## Certified Defense
  - Theoretical guarantees against all attacks within a certain threat model
  - Robustness certificate $RC(x, F, \epsilon)$: for all $x' \in B(x, \epsilon)$ we have that $F(x) = F(x')$

Robustness Certification

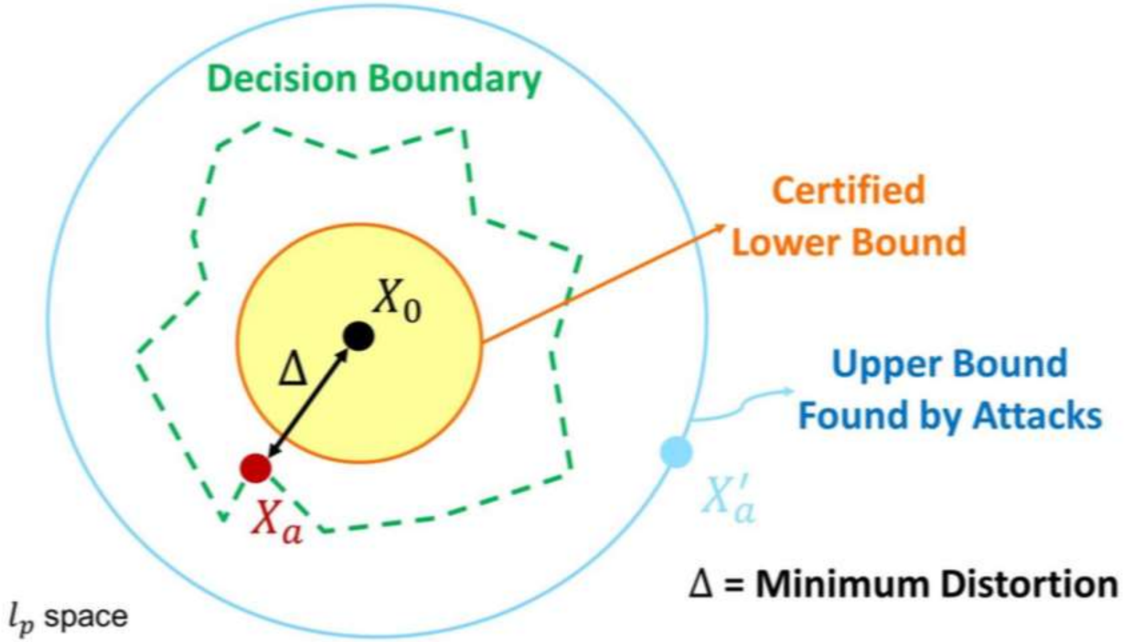
- The robustness property is determined by the **exact decision boundary**, which can be approximated by upper bound and lower bound.
- Adversarial attacks provide the asymptotic **upper bound**.
- The challenge is to compute the **lower bound** of the minimum adversarial distortions, i.e., certifying the robustness space around the input such that the model's prediction result is consistent within the space.

Problem Statement

Given

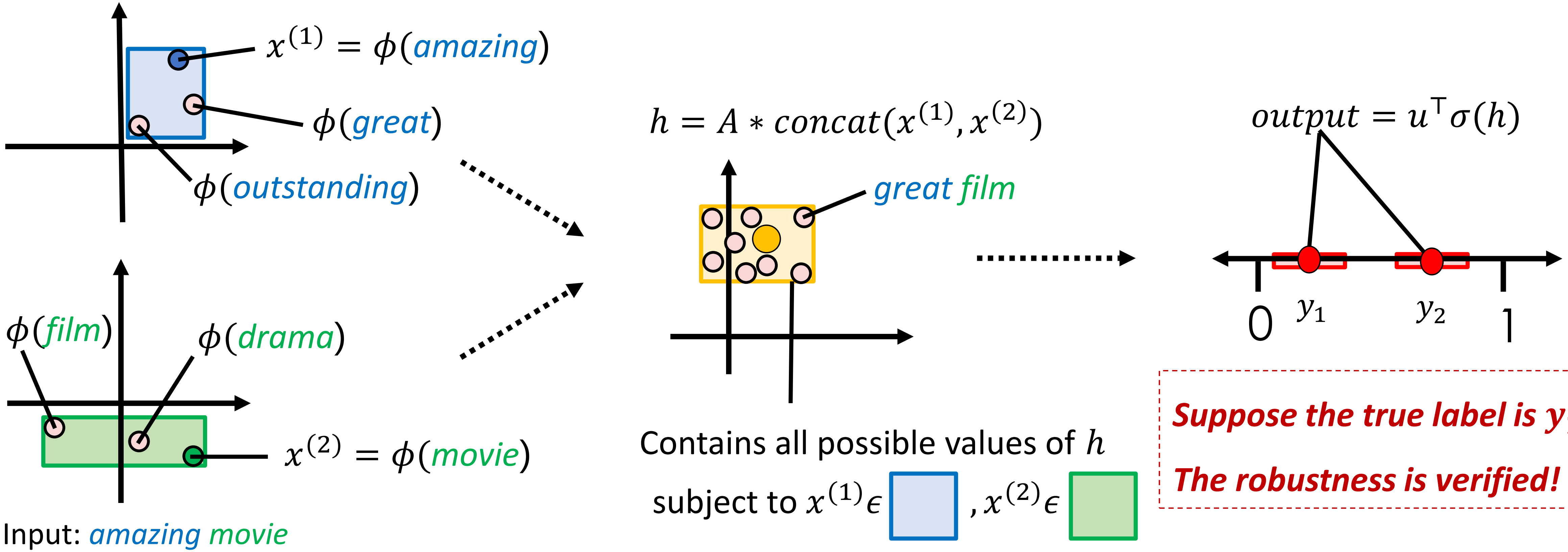
- a neural network N
- a property over inputs φ
- a property over outputs ψ

check whether $\forall i \in I . i \vDash \varphi \Longrightarrow N(i) \vDash \psi$ holds



Robustness Certification

- The basic idea to verify the robustness for a given ℓ_p-norm perturbation space:
  - Compute the lower and upper bounds of the output units for the given perturbation space.
  - If the lower bound of the true label output is larger than the upper bounds of all other labels, the robustness for the given perturbation space is verified.



Suppose the true label is y_2

The robustness is verified!

Robustness Certification Methods

Exact Certification

- Satisfiability Modulo Theories (SMT) [Ehlers *et al.* ATVA'17, Huang *et al.* CAV'17, Katz *et al.* CAV'17]
- Mixed-Integer Linear Programming (MILP) [Tjeng *et al.* ICLR '19]
- Accurate but usually **computationally expensive**, therefore cannot be scaled to large networks

Relaxed Certification

- Convex Polytope [Wong & Kolter ICML'18]
- Reachability Analysis [Weng *et al.* ICML'18, Zhang *et al.* NeurIPS'18]
- Abstract Interpretation [Mirman *et al.* ICML'18, Singh *et al.* POPL'19]
- Efficient but **cannot provide precise robustness bounds**

However, they are almost designed for FCNs and CNNs, seldom for RNNs!

Challenges for Certifying RNNs

$$\boldsymbol{c}^{(t)} = \sigma\left(\boldsymbol{f}^{(t)}\right) \odot \boldsymbol{c}^{(t-1)} + \sigma\left(\boldsymbol{i}^{(t)}\right) \odot \tanh\left(\tilde{\boldsymbol{c}}^{(t)}\right)$$

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{o}^{(t)}\right) \odot \tanh\left(\tilde{\boldsymbol{c}}^{(t)}\right)$$

Cross-nonlinearity

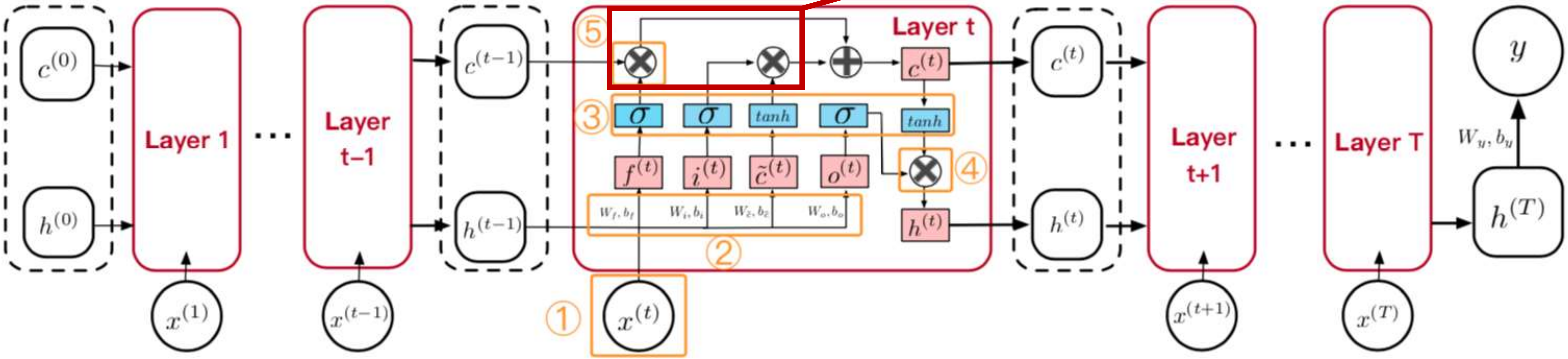


Figure 2: The architecture of an LSTM.

Robustness Certification for RNNs

➢Current Works (categorized by threat model)

- Symbol/Word Substitutions (limited attackers' ability)
  - Wang et al. NAACL'21
  - Dong et al. ICLR'21
  - Ye et al. ACL'20
  - Huang et al. EMNLP'19
  - Jia et al. EMNLP'19
- Embedding Perturbation (strong attackers' ability)
  - POPQORN [Ko *et al.* ICML'19]
    - imprecise – its linear relaxations do not retain high inter-variable correlations
    - inefficient – use gradient-based optimization to compute bounding planes
    - impractical – only evaluate one single word (one input frame) perturbation

Possible? Yes!

✓ Precise

✓ Efficient

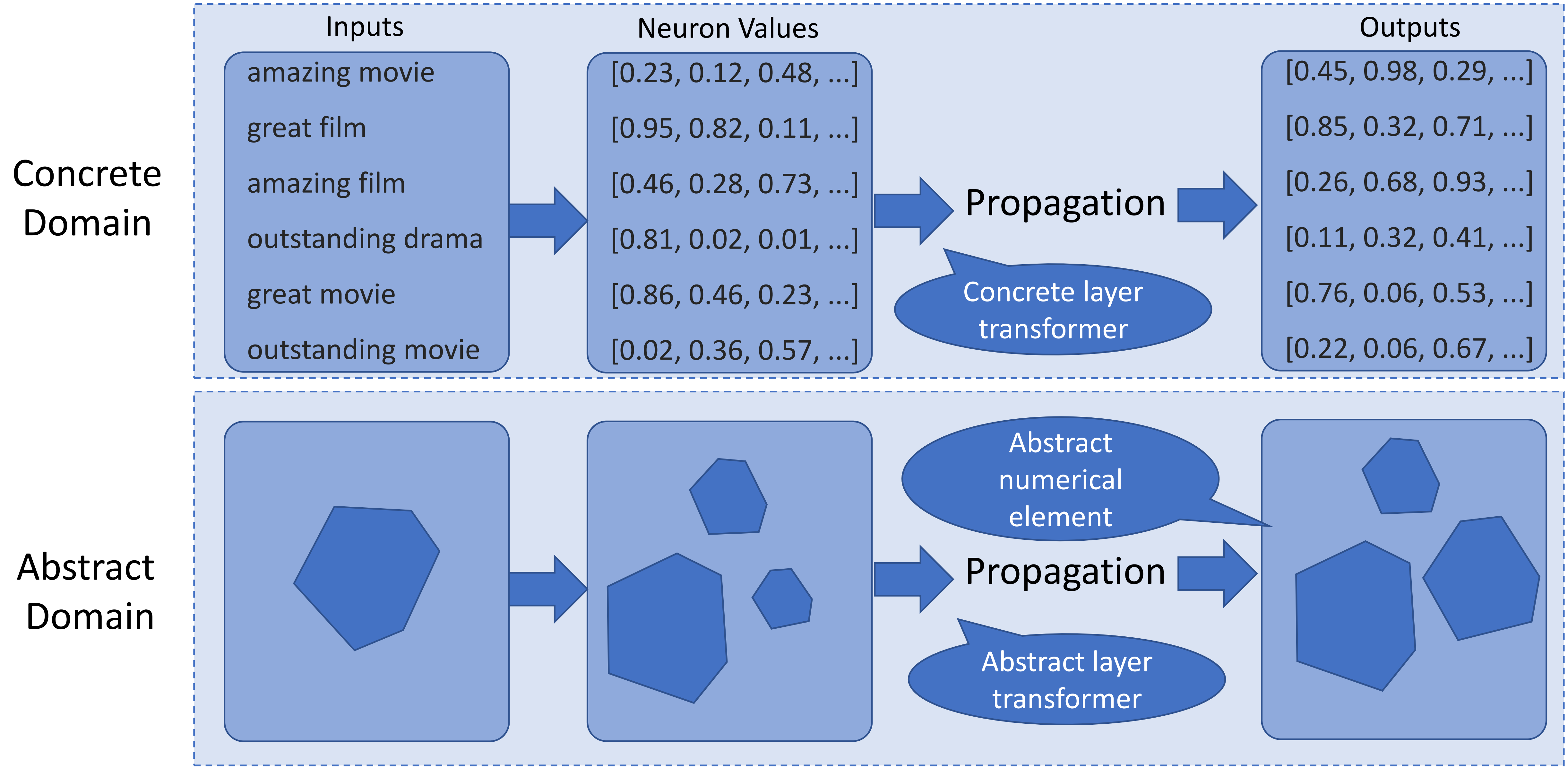
✓ Practical

Our Contribution

- 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**.
- We conduct extensive evaluation on **four security-sensitive applications** across **various network architectures** to empirically validate Cert-RNN's superiority.
- 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**.

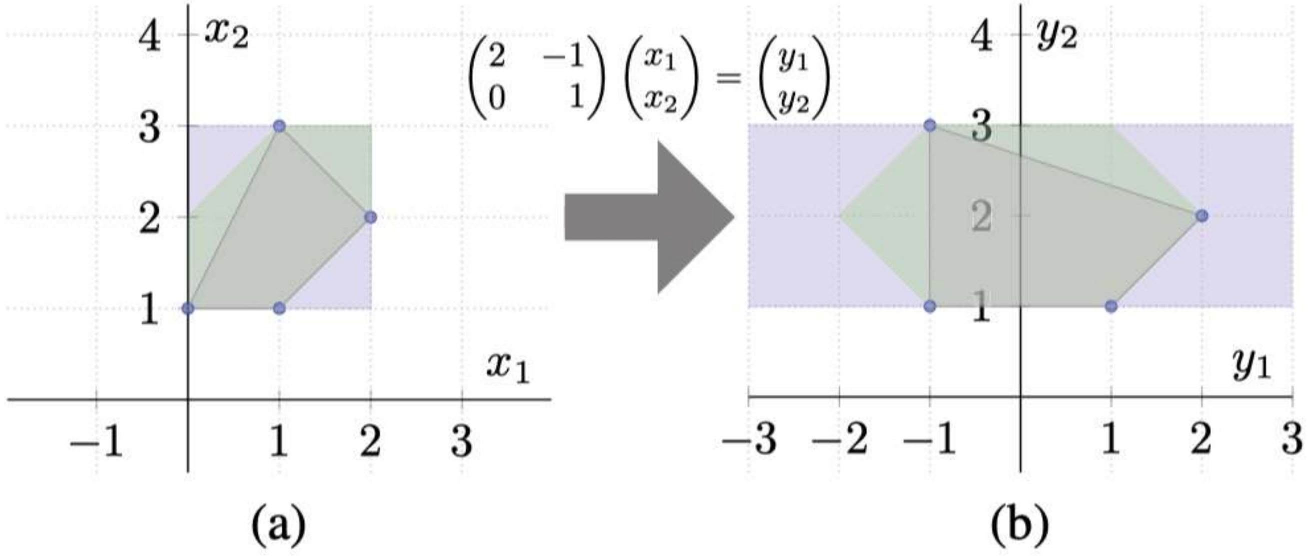
Method Design

Abstract Interpretation



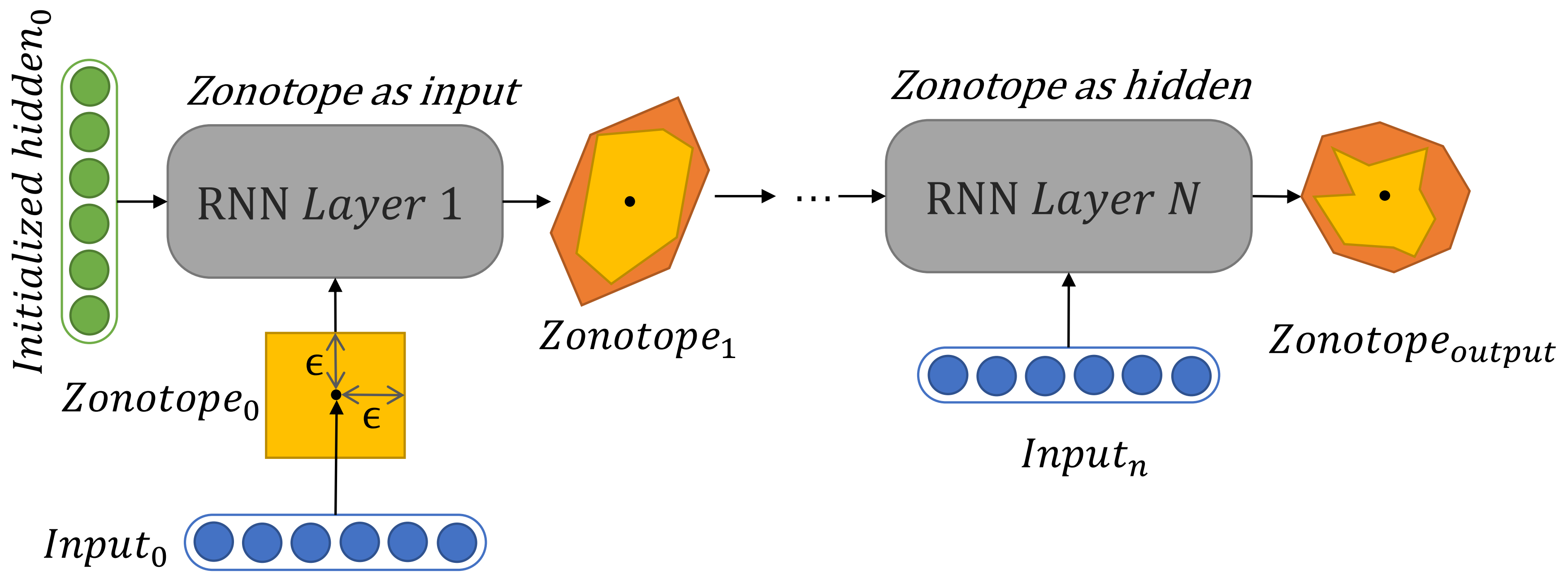
Abstract Interpretation

- Three popular numerical abstract domains
- We choose **zonotope** abstract domain for the following reasons:
  - Trades off precision and performance
  - Each variable (abstract neuron) captured in an affine form -> exact for linear operations
  - Allows relating variables through parameters



Box domain vs. Zonotope domain vs. Polyhedra domain
(From scalable to precise.)

Framework of Cert-RNN



Main Steps

1. A zonotope abstract domain is first defined to capture all potential adversarial inputs

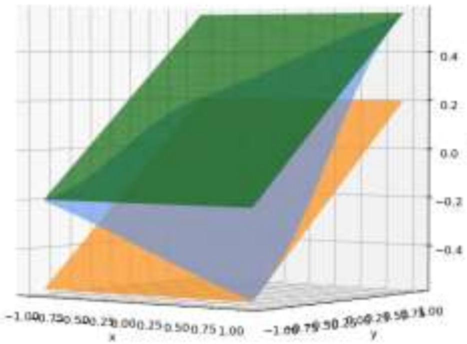
2. An abstract transformer is created for each non-linear operation of the RNN

3. Propagating the zonotope through all the layers of the target RNN

4. The output zonotope of the RNN's last layer is used to certify the robustness

Problem Definition

- **Definition 4.2** Given a continuously differentiable non-linear function $f(x_1, x_2, \dots)$ 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 $(x_1, x_2, \dots) \in \mathbf{z}$ as follows:



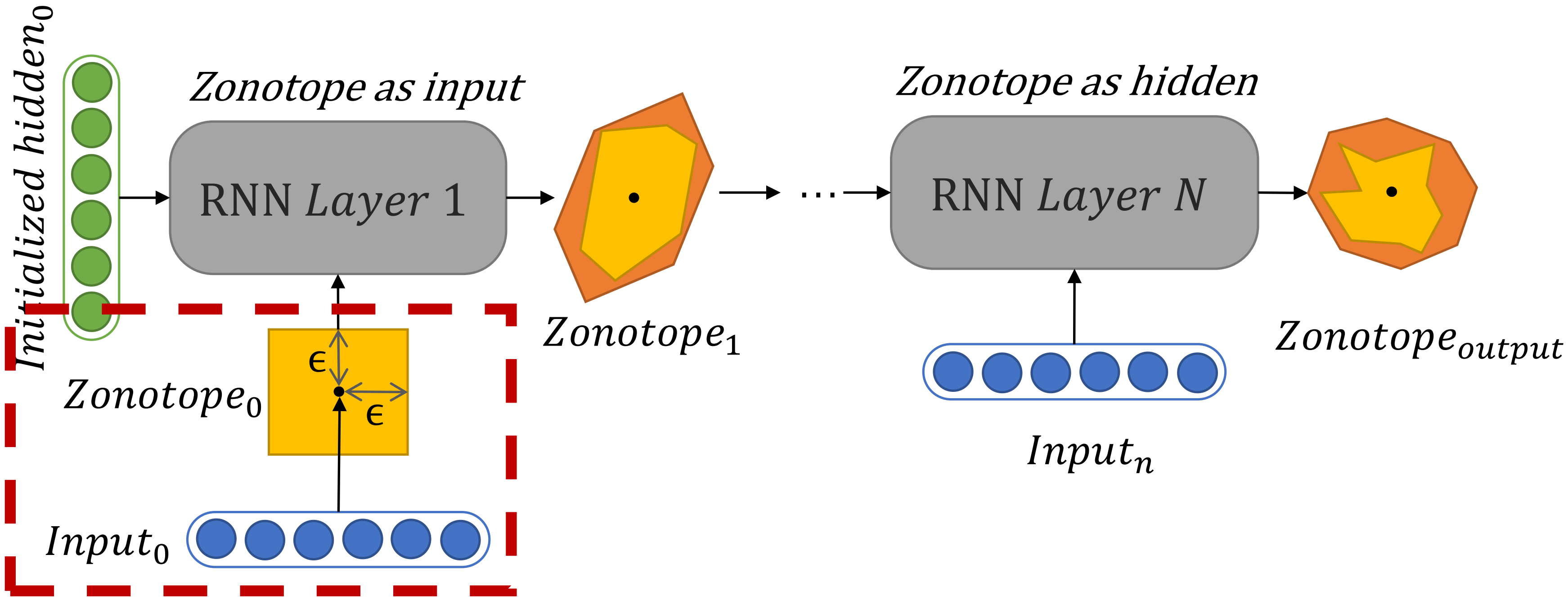
$$Z^L = C_1 + a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots ,$$
$$Z^U = C_2 + a_1 \cdot x_1 + a_2 \cdot x_2 + \cdots ,$$

  - $C_1, C_2, a_i \in \mathbb{R}$
  - when $a_i = 0 (i = 1,2, \dots)$, the zonotope approximation returns the interval range of f , i.e., $[C_1, C_2]$

- **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 $z_o = a_1 \cdot z_1 + a_2 \cdot z_2 + \cdots + \frac{C_2 - C_1}{2} \varepsilon_{new}$, 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:

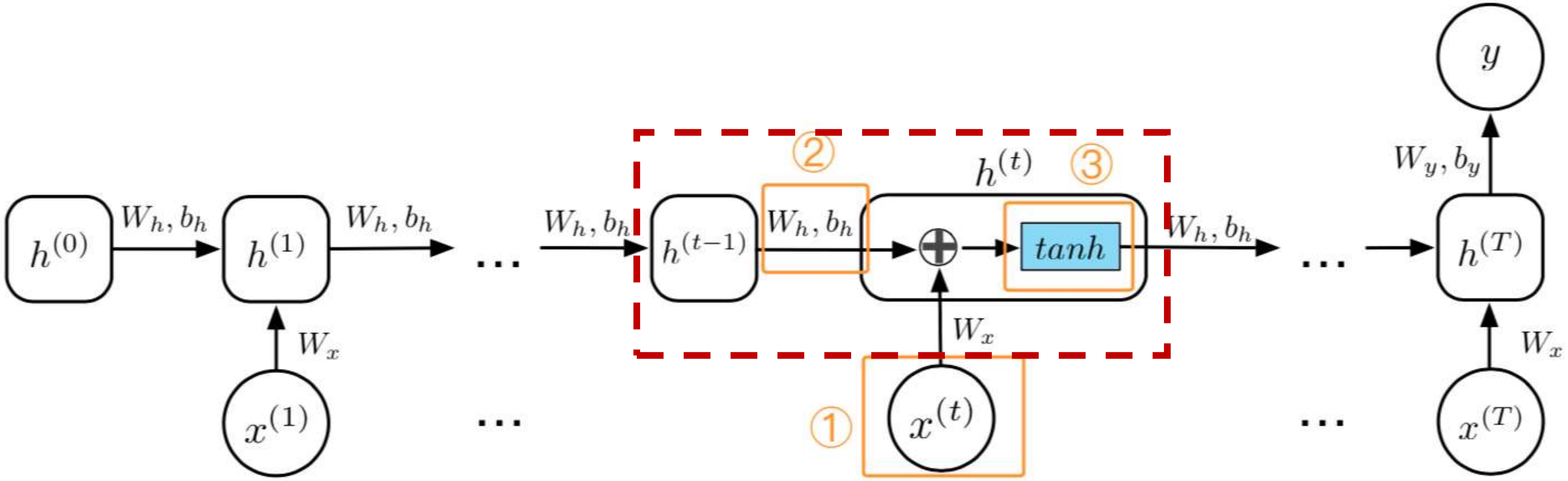
$$\min \quad \frac{C_2 - C_1}{2}.$$

Step 1: Input Region Abstraction

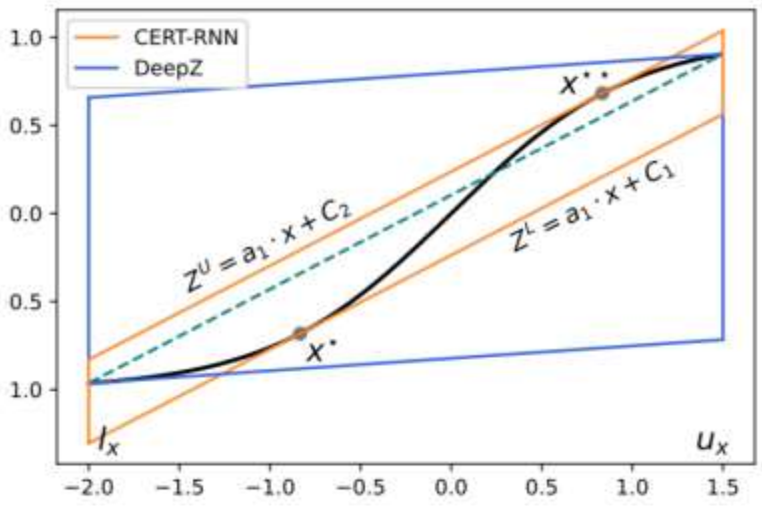


- Given an input sequence $\boldsymbol{X} = [\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(t)}, \cdots, \boldsymbol{x}^{(T)}]$, where $\boldsymbol{x}^{(t)} = [x_1^{(t)}, x_2^{(t)}, \cdots, x_K^{(t)}]$ represents the t-th input frame.
- Based on Definition 4.1, the input frame $\boldsymbol{x}^{(t)}$ is mapped to the center coefficient α_0 of a zonotope $\mathbf{z}$.
- For ℓ_∞-norm bounded attack, the adversarial perturbation of the j-th dimension of $\boldsymbol{x}^{(t)}$ is mapped to the coefficient α_{ij}.

Step 2: Intermediate Operation Abstraction



- Affine Transformation Abstraction
  - Can be exactly captured in our approximation
- Tanh Function Abstract Transformer
  - We propose a new abstract transformer for tanh
  - Tighter than DeepZ[Singh et al. NeurIPS'18]



Cert-RNN

➢ Intermediate Operation Abstraction

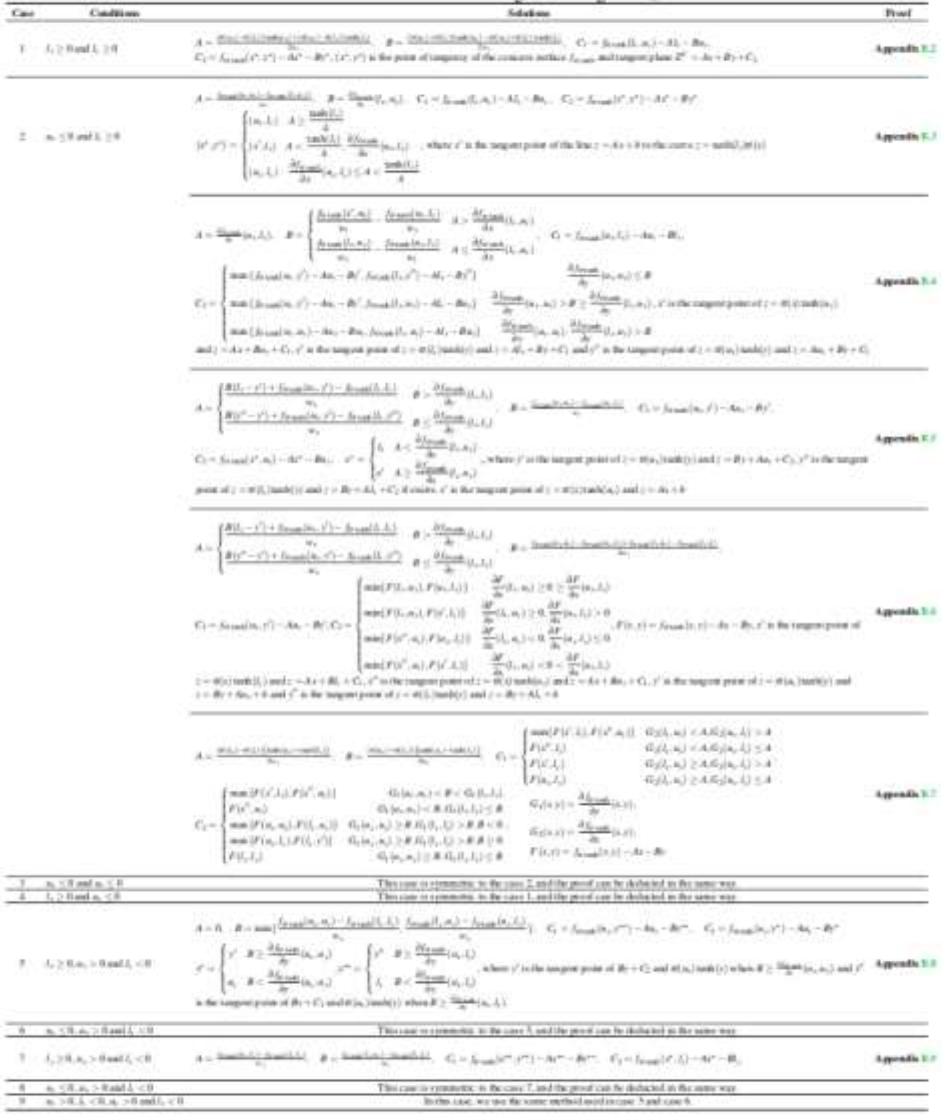
- Sigmoid ⊙ Tanh Abstract Transformer

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.

Table 5: Nine cases for the abstract transformer design of the Sigmoid ⊙ Tanh function.



Case	Conditions	Solutions	Proof

Cert-RNN

➢ Intermediate Operation Abstraction

- Sigmoid ⊙ Identity Abstract Transformer

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.

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**

Cert-RNN

➢ Certifying the Robustness Bound

- Specifically, finding the largest robustness bound ε_c for the input sequence with true label c can be formalized as the following optimization problem:

$$\begin{aligned} \max \quad & \varepsilon_c \\ s.t. \quad & \alpha_{0c} - \sum_{j=1}^{p} |\alpha_{jc} \cdot \varepsilon_j| \geq \alpha_{0i} + \sum_{j=1}^{p} |\alpha_{ji} \cdot \varepsilon_j|, \quad \forall i \neq c \end{aligned}$$

Algorithm 1: Computing the robustness bound.

Result: Certified robustness bound ε_c

Data: model $\mathcal{F}$, input sequence $\mathbf{X}_0$, true label c

for t *in* T **do**
  $\varepsilon^{(t)} = 0.5$
  for $l = 2$ *to* 13 **do**
    $\mathbf{z}_o = \textsc{Cert-RNN}(t, \mathcal{F}, \mathbf{X}_0, \varepsilon^{(t)})$;
    if $\alpha_{c0} - \sum_{j=1}^{p} |\alpha_{c_j} \cdot \varepsilon_j| \geq \alpha_{i0} + \sum_{j=1}^{p} |\alpha_{ij} \cdot \varepsilon_j|$ **then**
      $\varepsilon^{(t)} = \varepsilon^{(t)} + 0.5^l$;
    else
      $\varepsilon^{(t)} = \varepsilon^{(t)} - 0.5^l$;

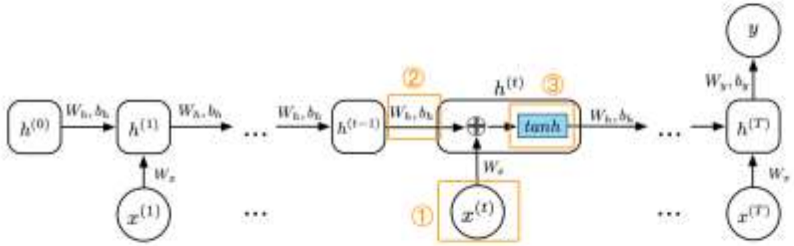
$\varepsilon_c = min(\varepsilon^{(1)}, \varepsilon^{(2)}, \cdots, \varepsilon^{(T)})$

A Toy Example

Steps

1. Adversarial Input Region Abstraction
2. Affine Transformation Abstraction
3. Tanh Function Abstract Transformer
4. Affine Transformation Abstraction
5. Robustness Verificatio for the given region

$\boldsymbol{x_0} = [x_1, x_2]^T = [1, -1]^T$



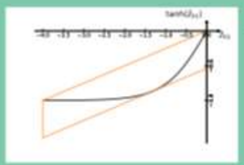
① $\ell_\infty - norm\ perturbation:\ \varepsilon = 1$

$\boldsymbol{z}_{input} = [z_1, z_2]^T = [x_1 + \varepsilon \cdot \varepsilon_1, x_2 + \varepsilon \cdot \varepsilon_2] = [1 + \varepsilon_1, -1 + \varepsilon_2]^T$

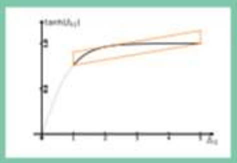
② $\tilde{\boldsymbol{z}}_{\boldsymbol{h}} = W_h \cdot h + W_x \cdot \boldsymbol{z}_{input} + b_h$ ← $W_h = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $W_x = \begin{bmatrix} 0 & 2 \\ 1 & -1 \end{bmatrix}$ $b_h = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $h = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\tilde{\boldsymbol{z}}_{\boldsymbol{h}} = [\tilde{z}_{h1}, \tilde{z}_{h2}]^T = [-2 + 2\varepsilon_2, 3 + \varepsilon_1 - \varepsilon_2]$

③ $\boldsymbol{z_h} = \tanh(\tilde{\boldsymbol{z}}_{\boldsymbol{h}})$ ←



z_{h1} *Approximation:* $\begin{cases} a_1 = 0.2498 \\ C_{11} = -0.5370 \\ C_{12} = 0 \end{cases}$



z_{h2} *Approximation:* $\begin{cases} a_2 = 0.0596 \\ C_{21} = 0.7020 \\ C_{22} = 0.8453 \end{cases}$

$$\boldsymbol{z_h} = [z_{h1}, z_{h2}]^T = \left[a_1\tilde{z}_{h1} + \frac{C_{12} - C_{11}}{2}\varepsilon_3, a_2\tilde{z}_{h2} + \frac{C_{22} - C_{21}}{2}\varepsilon_4\right] = [0.2498\tilde{z}_{h1} + 0.2685\varepsilon_3, 0.0596\tilde{z}_{h2} + 0.0717\varepsilon_4]^T$$

② $\boldsymbol{z_o} = W_y \cdot \boldsymbol{z_h} + b_y$ ← $W_y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $b_y = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$\boldsymbol{z_o} = [z_{o1}, z_{o2}]^T = [-0.4996 + 0.4996\varepsilon_2 + 0.2685\varepsilon_3, 1.1788 + 0.0596\varepsilon_1 - 0.0596\varepsilon_2 + 0.0717\varepsilon_4]^T$

$\xrightarrow{\varepsilon_1, \varepsilon_2, \varepsilon_3 \in [-1,1]}$ $z_{o1} \in [-1.2677, 0.2685], z_{o2} \in [0.9879, 1.3697]$

The true label's confidence value z_{o2} always larger than z_{o1} , thus the robustness is verified for $\varepsilon = 1$.

Evaluation

Experimental Setting

Dataset & Models

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

- 8 vanilla RNNs and 9 LSTMs with different hidden units and layers for MNIST Sequence
- an RNN and an LSTM with 32 hidden units for the other three datasets

Baseline Method

- POPQORN [Ko *et al.*, ICML’19]

Experimental Setting

Evaluation Metrics

- **Certified robustness bound**
  - the certified robustness bound of a particular sample x is the maximum ε for which we can certify that the model $f(x')$ will return the correct label, where x' is any adversarially perturbed version of x such that $\|x - x'\|_\infty \leq \varepsilon$
- **Verified accuracy**
  - the verified accuracy at ε of a dataset if the fraction of data items in the dataset with certified robustness bound of at least ε

Effectiveness & Efficiency

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 Mean	POPQORN Std	POPQORN Time (min)	CERT-RNN Mean	CERT-RNN Std	CERT-RNN 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

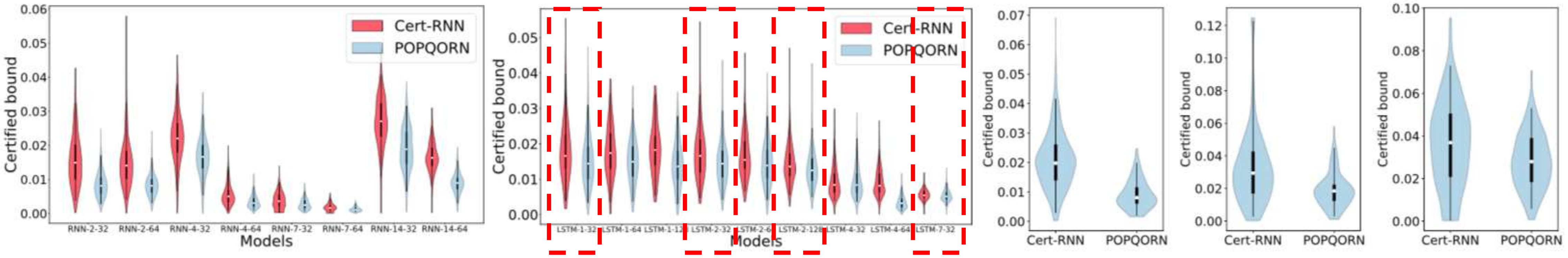
Table 3: Mann-Whitney U test results.

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}

➢ Remarks

- In all cases, Cert-RNN can obtain larger robustness bounds than that of POPQORN, i.e., the result of Cert-RNN is more accurate.
- Cert-RNN is much more efficient than POPQORN in general, especially for large and complex networks.
- For Mann-Whitney U test, the p-values of all models are small enough to reject the null hypothesis, which further demonstrates the superiority of Cert-RNN.

Effectiveness & Efficiency



(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.

➢ Remarks

- When the number of hidden units is the same, LSTMs with less layers would be more robust.
- When the number of layers is same, LSTMs with less hidden units would be more robust.
- 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.

Verified Accuracy

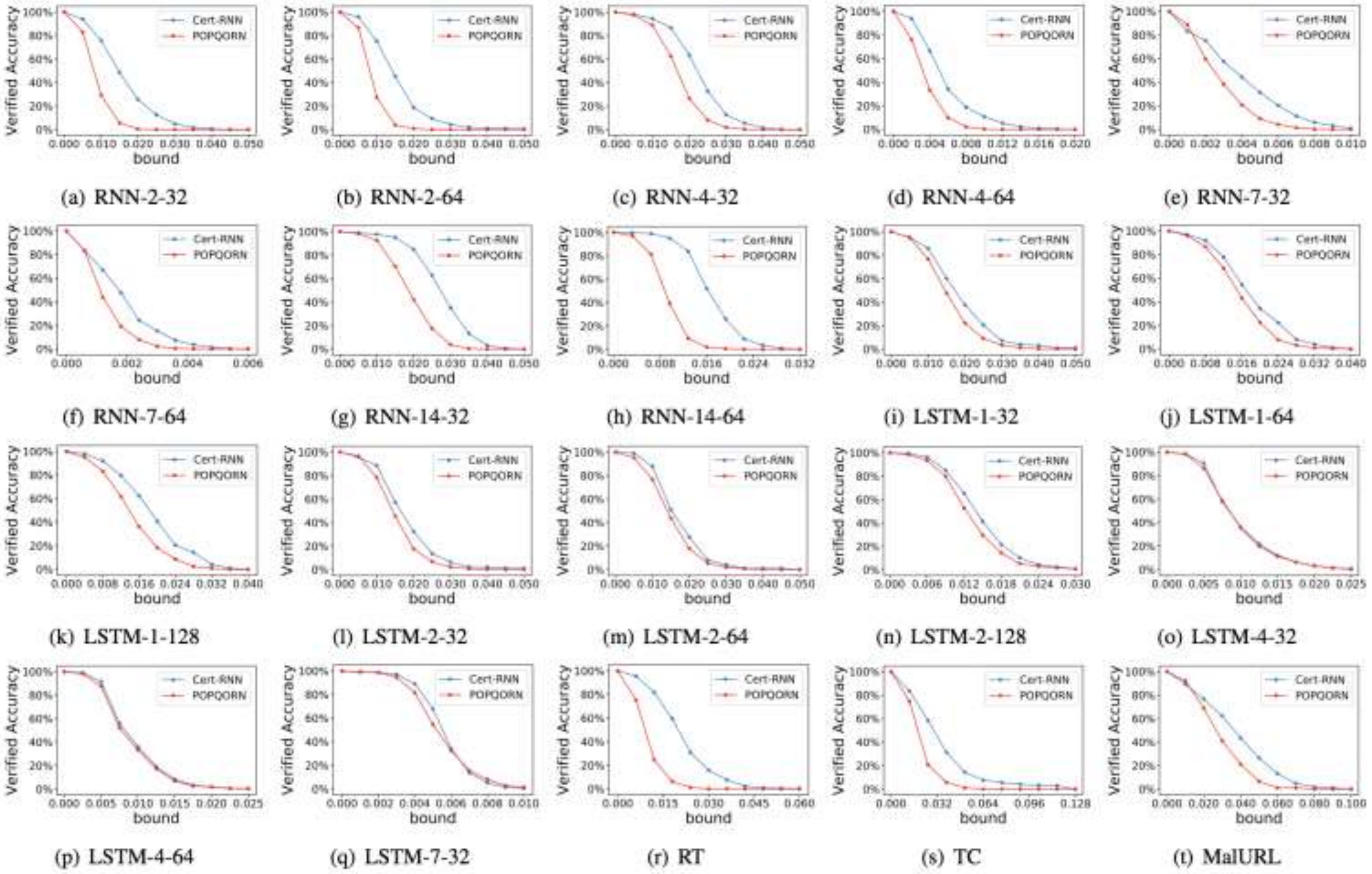


Fig. 7. Verified accuracy of the four datasets for each bound $\epsilon \in \Delta$(the x axis). The subfigures (a) to (q) are the results of the MNIST sequence dataset.

➢ Remarks

- The verified accuracy of Cert-RNN is much higher than that of POPQORN in most cases.

A More Threatening Scenario

➢ Perturbing All Frames

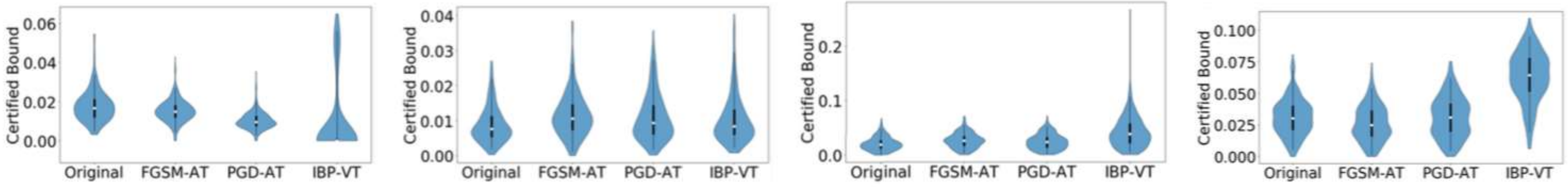
- Cert-RNN can handle this threat model while POPQORN cannot.
- Compared with perturbing one single frame, the robustness bounds for perturbing all frames decrease to some extent.

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

Application

Certifying Adversarial Defenses



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

➢ Defense Methods

- FGSM-AT (Fast Gradient Sign Method-based Adversarial Training) (Goodfellow et al. ICLR'15)
- PGD-AT (Projected Gradient Descent-based Adversarial Training) (Madry et al. ICLR'18)
- IBP-VT (Interval Bound Propagation-based Verified Training) (Gowal et al. ICCV'19)

➢ Remarks

- 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.

Improving RNN Robustness

➢ Implementation

- Our training follows [Gowal et al. CVPR'19, Mirman et al. ICML'18] – 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.

➢ Experimental Results

- The RNNs trained with Cert-RNN-VT achieve larger robustness bounds, outperforming the RNNs trained with IBP-VT on all three datasets. This is because the interval bounds obtained by our approximation of the tanh function is more accurate than that obtained by the IBP method.

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

Identifying Sensitive Words

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

➢ Remarks

- The words with smaller certified robustness bounds tend to be more important for the final prediction result, i.e., more sensitive.

Limitation & Discussion

➢Improving Zonotope Approximation

- Explore alternative zonotope approximations which lead to tighter robustness bounds

➢Supporting Other Norm-bounded Attacks

- The perturbations bounded by other norms can be considered as the subsets of ℓ_∞ in Cert-RNN

➢Supporting More Network Types

- Directly applicable to Gated Recurrent Unit (GRU) model
- New abstract transformers for attention module in Transformers
- The possibility for certifying sequence-to-sequence models

➢Supporting Other Threat Models

- Word substitution perturbation

Conclusion

Cert-RNN has three important advantages:

a) **Effectiveness** - it provides much tighter robustness bounds.

b) **Efficiency** – it scales to much more complex models.

c) **Practicality** - it enables a range of practical applications including evaluating the provable effectiveness for various defenses, improving the robustness of RNNs and identifying sensitive words.

Q&A

zjradty@zju.edu.cn