Robustness Evaluation for NNCS with different distribution functions

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Abstract—The robustness of Neural Networks (NNs) is important in security. A considerable works employ the lower bound on minimum distortion on the robustness of classification NNs. Compared to classification NNs, Neural Network Controlled Systems (NNCS) are more complex and are more pertinent to security. However, a gap exists in establishing a framework for the lower bound on minimum distortion in NNCS. To fill this gap and offer support for future works, we provide one of the properties of NNCS, the lower bound on its minimum distortion, for all states of interest with soundness. This paper demonstrates that NNCS adheres to Lipschitz continuity under general conditions. Subsequently, we utilize this continuity to delineate a lower bound for the minimum distortion of NNCS. More importantly, this paper proves that the distribution of the Lipschitz constant varies with the running time of NNCS and uses probability approaches to compute the lower bound. This framework is termed the Robustness Evaluation for NNCS framework (REN). Experiments validate that our framework is a promising work which can offer a lower bound on minimum distortion within both theoretical and applied domains.

Index Terms—Neural Network Controlled Systems, Lipschitz continuity, Probability distribution function

I. Introduction

Neural networks (NNs) are extensively applied in the Internet of Things (IoT) vehicles [1], [2], Large Language Models (LLMs) [3], [4], software maintaining [5], [6], and so on. These applications are intrinsically linked to the security of individuals. The assertion that the inherent uncertainty of NNs renders them susceptible to perturbations, thereby posing security risks, has been demonstrated in several studies [7], [8]. For instance, manipulated traffic signs could mislead specific autonomous driving systems, resulting in erroneous predictions [9]. The minimum perturbation that induces an unsafe situation is termed minimum distortion [10], which reflects the robustness of NN. Additionally, the lower bound on minimum distortion of NNs serves as a role for training more robust classification NNs [11]–[14], improving attack or defence methods [15], [16], evaluations [10], and so on.

In the fields of control theory, such as IoT vehicles [1], [17] and drones [18], [19], there is a growing trend to substitute traditional controllers with NNs [20]–[22], resulting in systems termed Neural Network Controlled Systems (NNCS). In practical applications, NNCS demonstrates enhanced efficiency, improved security performance, and greater adaptability to environmental changes [22], [23]. NNCS are closely tied to the security of human life and property, necessitate guarantees on robustness.

Recently, the verification of NNCS has attracted significant interest, prompting several verification initiatives [24]–[26]. Verification efforts by [27]–[29] inner-approximate the reach-avoid set (RA) to ensure the security of NNCS. The calculation of precise RA is NP-hard [27]. RA is approximated through the construction of barrier functions, which delineate safe conditions of the NNCS.

Meanwhile, several works further the barrier function of RA in control theory. Their barrier function inner-approximates the region-of-attraction (ROA) [30]–[32]. ROA is widely used in control fields such as stability analysis [33], [34], controller design [35], [36], robustness verification [31], [32] and so on. The calculation of the precise ROA is also NP-hard [30]. ROA is proposed according to the Lyapunov stability [37] in control theory offering a robust framework to verify the closed-loop stability of dynamical systems [32].

The lower bound on minimum distortion provided by barrier functions is limited within the domain of states under consideration and overestimated due to inner-approximation, as detailed in Exp V-C. Consequently, a gap persists in determining the lower bound for minimum distortion for all states of interest within NNCS.

Due to the fact that the calcualtion of the RA or ROA is NP-hard, it is challenging to obtain the precise minimum distortion of NNCS. Presently, attack methods [10], [16] can estimate the upper bound on the minimum distortion for NNCS [10]. However, a gap exists in establishing the sound lower bound on the minimum distortion for NNCS which is more applicable than upper bound from attacks [14]. To fill this gap, our method focuses on providing the lower bound on minimum distortion in NNCS within either RA or ROA's barrier functions for all state of interest. The distribution that governs this lower

This work is partially supported by TODO: Question for JiangTao Li.

bound is then proven to mitigate the risks due to sampling uncertainty. Experiments validate that our framework is a promising work which can offer a lower bound on minimum distortion within both theoretical and applied domains.

The key contributions of this work include:

- We establish that NNCS exhibits Lipschitz continuity under general conditions in two different scenarios (state/output feedback control) in Theorem 1. Subsequently, we derive a lower bound for the minimum distortion of NNCS in Theorem 2.
- We demonstrate that the Lipschitz constant invoked in the lower bound for minimum distortion of NNCS obeys One-point distribution or a non-degenerate cumulative distribution according to the duration(or time steps) of the NNCS in Theorem 3. Then we use probability approaches to compute this lower bound which is the robustness evaluation in our REN framework.
- We realize our proposed framework REN and evaluate it on several standard tasks for NNCS. The experiments validate the distribution properties established by our study and distinguish the REN framework from the barrier functions. Comparative results reveal that the REN framework provides a practical and sound lower bound.

a) Organaization.: Sec II provides the preliminaries. Our methodology and proofs are detailed in Sec III and Sec IV. Experiments are estimated in Sec V. Sec VI concludes this paper.

II. Preliminaries

This section provides the notions about NNCS, the definitions of RA and ROA.

A. Neural Network Controlled Systems (NNCS)

We discuss a nonlinear discrete-time system generally applied in tasks such as robot control and Programmable Logic Controller (PLC) operations [20], [38] with the following plant:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \tag{1a}$$

$$\boldsymbol{y}_t = \boldsymbol{h}(x_t). \tag{1b}$$

where $\boldsymbol{x}_t \in \mathbb{R}^{n_{\boldsymbol{x}}}$ denotes the state of the NNCS at time step $t, t \in \mathbb{N}$, $\boldsymbol{u}_t \in \mathbb{R}^{n_{\boldsymbol{u}}}$ represents the control input or action produced by the controller, which is a component of the NNCS, and the $\boldsymbol{y}_t \in \mathbb{R}^{n_{\boldsymbol{y}}}$ constitutes the output of the system. In the context of discrete-time NNCS research, the continuity of systems in reality needs to be considered. As a result, current state-of-the-art studies uniformly assume that the dynamic functions $f(\boldsymbol{x}_t, \boldsymbol{u}_t)$ and $h(\boldsymbol{x}_t)$ are continuous with respect to both $\boldsymbol{x}_t$ and $\boldsymbol{u}_t$ [27], [39], [40]. This work adheres to this assumption. We discuss two kinds of discrete-time NNCS:

1) State feedback control: In this scenario, the system only has a plant and a controller π parameterized by neural networks $\phi_\pi : \mathbb{R}^{n_{\boldsymbol{x}}} \to \mathbb{R}^{n_{\boldsymbol{u}}}$. This configuration is depicted in the left of Fig 1. The state feedback controller is defined as follows:

$$\boldsymbol{u}_t = \pi(\boldsymbol{x}_t) = \phi_\pi(\boldsymbol{x}_t) - \phi_\pi(\boldsymbol{x}^\star) + \boldsymbol{u}^\star. \tag{2}$$

Here, $\boldsymbol{x}^\star$ denotes the equilibrium state, generally a zero-filled vector, $\boldsymbol{0}_{n_{\boldsymbol{x}}} \in \mathbb{R}^{n_{\boldsymbol{x}}}$. Additionally, $\boldsymbol{u}^\star \in \mathbb{R}^{n_{\boldsymbol{u}}}$ represents the action at equilibrium.

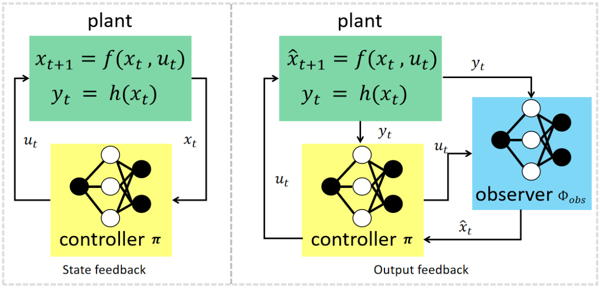


Fig. 1: In state feedback secnario, the NNCS has a plant and a controller implemented by a neural network π. In output feedback secnario, the NNCS has a plant, a controller neural network π, and an observer neural network ϕ_{obs}.

2) Output feedback control: In output feedback secnario, an additional model named observer is considered in NNCS, shown on the right of Fig 1. The observer is instrumental in mitigating errors both within and beyond the model [39], [41], [42]. It utilizes system input and output data to estimate unmeasurable internal states:

$$\hat{\boldsymbol{x}}_{t+1} = f(\hat{\boldsymbol{x}}_t, \boldsymbol{u}_t) + \phi_{obs}(\hat{\boldsymbol{x}}_t, \boldsymbol{y} - \boldsymbol{h}(\hat{\boldsymbol{x}}_t)) - \phi_{obs}(\hat{\boldsymbol{x}}_t, \boldsymbol{0}_{n_{\boldsymbol{y}}}), \tag{3a}$$

$$\boldsymbol{u}_t = \pi(\hat{\boldsymbol{x}}_t, \boldsymbol{y}_t) = \phi_\pi(\hat{\boldsymbol{x}}_t, \boldsymbol{y}_t) - \phi_\pi(\boldsymbol{x}^\star, \boldsymbol{h}(\boldsymbol{x}^\star)) + \boldsymbol{u}^\star. \tag{3b}$$

where $\hat{\boldsymbol{x}} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ represents the estimated state derived from the observer. According to the Luenberger observer [39] invoked in Yang et.al [32], $\hat{\boldsymbol{x}}_0 = \boldsymbol{H}\boldsymbol{x}_0$, $\boldsymbol{H} \in \mathbb{R}^{n_{\boldsymbol{x}}} \times \mathbb{R}^{n_{\boldsymbol{x}}}$ is a set matrix. The function $\phi_{obs} : \mathbb{R}^{n_{\boldsymbol{x}}} \times \mathbb{R}^{n_{\boldsymbol{y}}} \to \mathbb{R}^{n_{\boldsymbol{x}}}$ constitutes a neural network that supplants the conventional observer process, as detailed in [39].

3) Union formulation: In the following, we introduce the union formulation [32] for the state and output feedback scenarios, which is based on an internal state $\boldsymbol{\xi}_t \in \mathbb{R}^{n_{\boldsymbol{\xi}}}$ and the dynamic function $\boldsymbol{\xi}_{t+1} = f_d(\boldsymbol{\xi}_t) = f_d^{(t+1)}(\boldsymbol{\xi}_0)$, to facilitate the analysis.

For state feedback secnario, there is simply $\boldsymbol{\xi}_t = \boldsymbol{x}_t$ and:

$$f_d(\boldsymbol{\xi}_t) = f(\boldsymbol{\xi}_t, \pi(\boldsymbol{\phi}_t)). \tag{4}$$

For output feedback secnario, the internal state $\boldsymbol{\xi}_t = [\boldsymbol{x}_t, \boldsymbol{e}_t]^T$, where $\boldsymbol{e}_t = \hat{\boldsymbol{x}}_t - \boldsymbol{x}_t$ is the prediction error. The

$\boldsymbol{\xi}_t$ is a verctor function of state $\boldsymbol{x}_t$. The dynamic function is defined as:

$$f_d(\boldsymbol{\xi}_t) = \begin{bmatrix} f(\boldsymbol{x}_t, \pi(\boldsymbol{x}_t, \boldsymbol{h}(\boldsymbol{x}_t))), \\ f(\boldsymbol{x}_t, \pi(\hat{\boldsymbol{x}}_t, \boldsymbol{h}(\boldsymbol{x}_t))) + \boldsymbol{g}(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t) - \boldsymbol{x}_t \end{bmatrix}. \quad (5)$$

where $\boldsymbol{g}(\boldsymbol{x}_t, \hat{\boldsymbol{x}}_t) = \phi_{obs}(\hat{\boldsymbol{x}}_t, \boldsymbol{h}(\boldsymbol{x}_t) - \boldsymbol{h}(\hat{\boldsymbol{x}}_t)) - \phi_{obs}(\hat{\boldsymbol{x}}_t, \boldsymbol{0}_{n_y})$.

B. Reach-Avoid set

Under the discrete-time systems with dynamics (4) or (5), the Reach-Avoid set (RA) [27], [28] enables any state starting from RA to reach a target set TR in finite time while staying within a compact set $\mathcal{B} \in \boldsymbol{X}$ until the target is hitten, where $\boldsymbol{X} = \{\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{x}} | h_b(\boldsymbol{x} < 1)\}$ is the region of all states in the secnario.

Definition 1: (Reach-Avoid Set (RA) [27]). The reach-avoid set RA is the set of all states which achieve the target set TR at finite time $t \in \mathbb{N}$ while maintaining in the set $\mathcal{B}$ over the time horizon $[0, t] \cap \mathbb{N}$, i.e.,

$$RA = \{\boldsymbol{x}_0 \in \mathcal{B} | \exists t \in \mathbb{N}, \boldsymbol{x}_t \in TR \wedge \bigwedge_{j=1}^{t} \boldsymbol{x}_j \in \mathcal{B}\} \quad (6)$$

where TR and $\mathcal{B}$ are defined by polynomial inequations

$$TR = \{\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{x}} | g(\boldsymbol{x}) < 1\} \quad (7a)$$

$$\mathcal{B} = \{\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{x}} | h_0(\boldsymbol{x}) \leq 0\} \quad (7b)$$

The $g(\boldsymbol{x})$, $h_0(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ and $TR \subset \mathcal{B}$, where $\mathbb{R}[\boldsymbol{x}]$ means the polynomial ring with respect to $\boldsymbol{x}$ which is smooth to $\boldsymbol{x}$. The RA set is inner-approximated in Xue et al. as $\mathcal{S} = \{\boldsymbol{x} \in \mathcal{B} | V(\boldsymbol{x}) < 1\}$, $\mathcal{S} \subset$ RA where

$$V(\boldsymbol{x}) := lim_{t \to \infty} inf \ \frac{\sum_{i=0}^{t-1} g(\boldsymbol{x}_i)}{t} \quad (8)$$

The barrier function $1 - V(\boldsymbol{x})$ in Xue et al [27] inner-approximates the RA through following optimization:

$$max \ \boldsymbol{c} \cdot \boldsymbol{w} \quad (9a)$$

$$s.t. V(\boldsymbol{x}) - V(f(\boldsymbol{x})) + s_0(\boldsymbol{x})h_0(\boldsymbol{x}) + s_1(\boldsymbol{x})(1 - g(\boldsymbol{x})) \in \sum[\boldsymbol{x}] \quad (9b)$$

$$V(\boldsymbol{x}) - g(\boldsymbol{x}) - w(f(\boldsymbol{x})) + w(\boldsymbol{x}) + s_2(\boldsymbol{x})h_0(\boldsymbol{x}) + s_3(\boldsymbol{x})(1 - g(\boldsymbol{x})) \in \sum[\boldsymbol{x}] \quad (9c)$$

$$V(\boldsymbol{x}) - 1 + s_4(\boldsymbol{x})h_b(\boldsymbol{x}) - s_5(\boldsymbol{x})h_0(\boldsymbol{x}) \in \sum[\boldsymbol{x}] \quad (9d)$$

where $\boldsymbol{c} \cdot \boldsymbol{w} = \int_{\mathcal{B}} V(\boldsymbol{x})dx$, $\boldsymbol{w}$ is computed by integrating the monomials in $V(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$, $\boldsymbol{c}$ is the optimizable vector of coffecients in $V(\boldsymbol{x})$, function $s_i(\boldsymbol{x})$, $w(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$.

C. Region of attraction

The barrier functions based on RA can be furthered in the field of control with the Region of Attraction (ROA). ROA is a fundamental concept in control theory. It denotes the subset of the state space in which every initial possible state leads system converge to the equilibrium states eventually. In other words, if the state of a system resides within its ROA, the system is guaranteed to transition into the equilibrium state with limited time.

Definition 2: (Region of attraction (ROA) [32]). The region of attraction for an equilibrium state $\boldsymbol{x}^\star$ is the largest invariant set $\mathcal{A} \in \mathcal{B}$, $\exists \ t \in \mathbb{N}$ s.t. $lim_{t \to \infty} \boldsymbol{x}_t = \boldsymbol{x}^\star, \forall \boldsymbol{x}_0 \in \mathcal{A}$ holds under the dynamics (4) or (5).

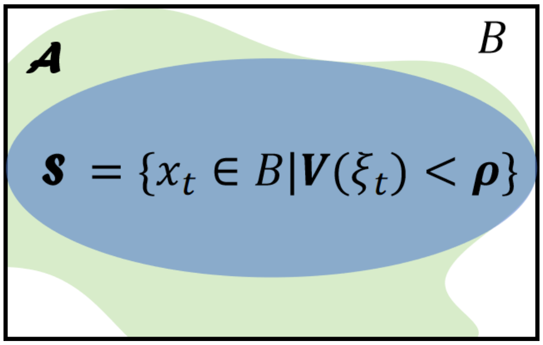


Fig. 2: The $\mathcal{B}$ is the region of interest, $\mathcal{A}$ is a non-covex region of attraction which is hard to be detected, and $\mathcal{S} = \{\boldsymbol{x}_t \in \mathcal{B} | V(\boldsymbol{\xi}_t) < \rho\}$ is the convex inner-approximation of $\mathcal{A}$, $V(\boldsymbol{\xi}) : \mathbb{R}^{n_{\boldsymbol{\xi}}} \to \mathbb{R}$ is a Lyapunov function and $\boldsymbol{\xi}_t = [\boldsymbol{x}_t, \boldsymbol{e}_t]^T$.

When the set of equilibrium states $\boldsymbol{x}^\star$ in ROA secnario constitutes the target set TR as defined in RA method, the ROA corresponds to the reach-avoid set as mentioned in RA method. Fig.2 presents the relationships among the region of interest $\mathcal{B}$, the region of attraction (ROA) $\mathcal{A}$, and one of the potential inner approximations of $\mathcal{A}$, denoted as $\mathcal{S}$. If $\exists \boldsymbol{x}_t \notin \mathcal{B}$, $t \in \mathbb{N}$, then the initial state $\boldsymbol{x}_0$ is deemed unsafe for the scenario. Since identifying the precise region of $\mathcal{A}$ is NP-hard. Verification studies [30]–[32] inner-approximate $\mathcal{A}$ through an optimization based on Lyapunov conditions [37]:

$$max_V Vol(\mathcal{S}) \quad (10a)$$

$$s.t. V(\boldsymbol{\xi}_t) > 0 \ \forall \boldsymbol{\xi}_t \neq \boldsymbol{\xi}^\star \in \mathcal{S} \quad (10b)$$

$$V(\boldsymbol{\xi}_{t+1}) - V(\boldsymbol{\xi}_t) \leq -\kappa V(\boldsymbol{\xi}_t) \ \forall \boldsymbol{\xi}_t \in \mathcal{S} \quad (10c)$$

$$V(\boldsymbol{\xi}^\star) = 0. \quad (10d)$$

The $\kappa > 0$ is a set parameter for exponential stability convergence rate. Constrains (10b), (10c), and (10d) are the Lyapunov conditions, guarantee the states from $\mathcal{S}$ lead system converge to the equilibrium state $\boldsymbol{x}^\star$ and $\boldsymbol{\xi}^\star = [\boldsymbol{x}^\star, \boldsymbol{0}_{n_{\boldsymbol{x}}}]^T$. The parameter ρ is learned during the training of NNCS. Additionally, the Lyapunov function $V(\boldsymbol{\xi}) : \mathbb{R}^{n_{\boldsymbol{\xi}}} \to \mathbb{R}$ has following learnable structure:

$$V(\boldsymbol{\xi}_t) = (\boldsymbol{\xi}_t - \boldsymbol{\xi}^\star)^T (\epsilon \boldsymbol{I} + \boldsymbol{R}^T \boldsymbol{R})(\boldsymbol{\xi}_t - \boldsymbol{\xi}^\star), \quad (11)$$

The structure (11) of $V(\boldsymbol{\xi})$ naturally satisfies the Lyapunov conditions (10b) and (10d).

III. Robustness evaluation for NNCS

In this section, we begin by providing the definition pertaining to minimum distortion and some lemmas. Subsequently, we conduct the lower bound on minimum distortion of NNCS by proving its Lipschitz continuity with respect to the initial state $\boldsymbol{x}_0$, as delineated in Theorem 1 and 2, no matter in the RA or ROA method. The conducted lower bound on the minimum distortion is remarked as robustness evaluation.

Definition 3: (Minimum distortion) Let $\boldsymbol{x_0} \in \mathbb{R}^{n_{\boldsymbol{x}}}$ is the initial vector of NNCS : $\mathbb{R}^{n_{\boldsymbol{x}}} \rightarrow \mathbb{R}^{n_{\boldsymbol{y}}}$, and $\boldsymbol{x}_0 \in \mathcal{A}$. The p-norm minimum distortion $\Delta_{p,min}$ for $\boldsymbol{x}_0$ is denoted as the smallest perturbation $\Delta_p = \|\delta\|_p$ s.t. $\boldsymbol{x}_0 + \delta \notin \mathcal{A}$.

However, identifying the precise region of $\mathcal{A}$ for the precise $\Delta_{p,min}$, poses a significant challenge as it is a NP-hard problem. Our robustness evaluation computed as the lower bound on $\Delta_{p,min}$, denoted as $\Delta^L_{p,min} = \|\delta^L_{min}\|_p$, such that the perturbed state $\boldsymbol{z}_t = f_d^{(t)}(\boldsymbol{x}_0 + \delta^L_{min}) \notin \mathcal{S}$ within a finite time t and $\boldsymbol{x}_0 \in \mathcal{A}$. Given that $\mathcal{S} \subseteq \mathcal{A}$, it follows that $\Delta^L_{p,min} \leq \Delta_{p,min}$, the $\Delta^L_{p,min}$ is the robustness evaluation in REN. Before presenting the robustness evaluation, it is necessary to introduce several assumptions and lemmas.

Assume 1: Because both the dynamics $f(\boldsymbol{x}, \boldsymbol{y})$ and $h(\boldsymbol{x})$ simulate the continuous dynamics of systems, it is a common assumption in various tasks such as stability analysis [33], [34], controller design [35], [36], and robustness verification [31], [32] and so on. to consider the dynamics $f(\boldsymbol{x}, \boldsymbol{y})$ to be continuously differentiable with respect to $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, as is $h(\boldsymbol{x})$ with respect to $\boldsymbol{x}$.

Assume 2: For the convenience of the following proofs, assume that the neural networks ϕ_π and ϕ_{obs} have a pair of input-output layers, and only one hidden connected layer with common activation functions (like ReLU, LeakyReLU, Sigmoid, Tanh and so on).

Lemma 1: According to the appendix C and D in Weng et al. [10], it is stated that a neural network $\phi(\boldsymbol{x})$ is continuously differentiable almost everywhere with respect to its input $\boldsymbol{x}$, despite the activation functions used in the network are not absolutely Lipschitz continuity (like ReLU). In other words, $\phi(\boldsymbol{x})$ is Lipschitz continuous almost everywhere with respect to $\boldsymbol{x}$ i.e., let $\mathcal{B}$ be a convex bounded closed input region, $\phi(x) : \mathcal{B} \rightarrow \mathbb{R}$ be the neural network, $\exists\ L_q > 0$, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{B}$, s.t.

$$||\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})|| \leq L_q \cdot \|\boldsymbol{x} - \boldsymbol{y}\|_p \tag{12}$$

where $\frac{1}{p} + \frac{1}{q} = 1$, $1 \leq p$ and $q \leq \infty$. $L_q = max\{||\nabla\phi(\boldsymbol{x})||_q : \boldsymbol{x} \in \mathbb{B}\}$, $\nabla\phi(\boldsymbol{x})$ is the gradient of $\phi(\boldsymbol{x})$.

Theorem 1: (Lipschitz continuity of barrier function $g(\boldsymbol{\xi})$) Given a NNCS considering the plant (1a), (1b) within dynamics (4) or (5), and the learned inner-approximation, $\mathcal{S} = \{\boldsymbol{x}_t \in \mathcal{B} | V(\boldsymbol{\xi}_t) < \rho\}$. The barrier function $g(\boldsymbol{\xi}_t) = \rho - V(\boldsymbol{\xi}_t)$ is Lipschitz continuous almost everywhere with respect to the initial input $\boldsymbol{x}_0 \in \mathcal{B}$ within limited time steps t.

Because of the differentiable continuous function $V(\boldsymbol{x})$ in both ROA and RA, the Theorem 1 discusses the Lipschitz continuity of $\boldsymbol{\xi}_t$ with respect to $\boldsymbol{x}_0$ in fact, the Lipschitz constant of $V(\boldsymbol{x})$ denoted as L_v. For convenience, we set $q = 1$ and $p = \infty$ in the proof. As the output feedback secnario contains the state feedback one, the subsequent proofs will be directed towards the output feedback scenario.

Proof 1: We employ the mathematical induction to establish the proof:

Base case: When $t = 0$, $\boldsymbol{\xi}_1 = f_d(\boldsymbol{\xi}_0) = [\boldsymbol{x}_1, \hat{\boldsymbol{x}}_1 - \boldsymbol{x}_1]^T$, let $\boldsymbol{z}_0 \in \mathcal{B}$, $\boldsymbol{z}_0 \neq \boldsymbol{x}_0$, where $\tilde{\boldsymbol{z}}_0 = [\boldsymbol{z}_0, \hat{\boldsymbol{z}}_0]$, and $\hat{\boldsymbol{z}}_0 = \boldsymbol{H}\boldsymbol{z}_0$, $\hat{\boldsymbol{z}}_0 \in \mathbb{R}^{n_{\boldsymbol{x}}}$, then:

$$\begin{aligned} ||f_d(\tilde{\boldsymbol{z}}_1) - f_d(\boldsymbol{\xi}_1)|| &= || \begin{bmatrix} \boldsymbol{z}_1 - \boldsymbol{x}_1, \\ \hat{\boldsymbol{z}}_1 - \hat{\boldsymbol{x}}_1 - (\boldsymbol{z}_1 - \boldsymbol{x}_1) \end{bmatrix} || \\ &\leq ||\boldsymbol{z}_1 - \boldsymbol{x}_1|| + ||\hat{\boldsymbol{z}}_1 - \hat{\boldsymbol{x}}_1 - (\boldsymbol{z}_1 - \boldsymbol{x}_1)|| \end{aligned} \tag{13}$$

$$||\boldsymbol{z}_1 - \boldsymbol{x}_1|| = ||f(\boldsymbol{z}_0, \pi(\hat{\boldsymbol{z}}_0, \boldsymbol{h}(\boldsymbol{z}_0))) - f(\boldsymbol{x}_0, \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0)))||, \tag{14}$$

$$\begin{aligned} ||\hat{\boldsymbol{z}}_1 - \hat{\boldsymbol{x}}_1 - (\boldsymbol{z}_1 - \boldsymbol{x}_1)|| &\leq ||\boldsymbol{z}_0 - \boldsymbol{x}_0|| + ||\boldsymbol{z}_1 - \boldsymbol{x}_1|| \\ &+ ||f(\hat{\boldsymbol{z}}_0, \pi(\hat{\boldsymbol{z}}_0, \boldsymbol{h}(\boldsymbol{z}_0))) - f(\hat{\boldsymbol{x}}_0, \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0)))|| \\ &+ ||\phi_{obs}(\hat{\boldsymbol{z}}_0, \boldsymbol{h}(\boldsymbol{z}_0) - \boldsymbol{h}(\hat{\boldsymbol{z}}_0)) - \phi_{obs}(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0) - \boldsymbol{h}(\hat{\boldsymbol{x}}_0))|| \\ &+ ||\phi_{obs}(\hat{\boldsymbol{z}}_0, \boldsymbol{0}_{n_{\boldsymbol{y}}}) - \phi_{obs}(\hat{\boldsymbol{x}}_0, \boldsymbol{0}_{n_{\boldsymbol{y}}})|| \end{aligned} \tag{15}$$

Add $0 = -f(\boldsymbol{x}_0, \pi(\hat{\boldsymbol{z}}_0, \boldsymbol{h}(\boldsymbol{z}_0))) + f(\boldsymbol{x}_0, \pi(\hat{\boldsymbol{z}}_0, \boldsymbol{h}(\boldsymbol{z}_0)))$ to $||f(\boldsymbol{z}_0, \pi(\hat{\boldsymbol{z}}_0, \boldsymbol{h}(\boldsymbol{z}_0))) - f(\boldsymbol{x}_0, \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0)))||$ on the right side of the inequality (15). Because of Assume 1, let the Lipschitz constants of $f(\boldsymbol{x}, \boldsymbol{y})$ with respect to $\boldsymbol{x}$, $\boldsymbol{y}$ are L_{fx} and L_{fy} respectively, then:

$$\begin{aligned} ||\boldsymbol{z}_1 - \boldsymbol{x}_1|| \leq & L_{fx}||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty + L_{fy}||\pi(\hat{\boldsymbol{z}}_0, \boldsymbol{h}(\boldsymbol{z}_0)) - \\ & \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0))||_\infty \end{aligned} \tag{16}$$

Because $\pi(\boldsymbol{x}, \boldsymbol{y})$ and $\phi_{obs}(\boldsymbol{x}, \boldsymbol{y})$ are in the form of $\boldsymbol{W}_2 \cdot \sigma(\boldsymbol{W}_1[:, :\ n_{\boldsymbol{x}}] \cdot \boldsymbol{x} + \boldsymbol{W}_1[:, n_{\boldsymbol{x}}\ :] \cdot \boldsymbol{y} + \boldsymbol{b}_1)$ (Assume 2), where $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ denote the weights and biases in the i-th layer of neural networks, $\sigma(\cdot)$ implies the activation functions. According to Lemma 1, the Lipschitz constants of $\pi(\boldsymbol{x}, \boldsymbol{y})$ with respect to $\boldsymbol{x}$ and $\boldsymbol{y}$ are denoted as $L_{\pi x}$ and $L_{\pi y}$, likewise, $L_{\phi x}$ and $L_{\phi y}$ for $\phi_{obs}(\boldsymbol{x}, \boldsymbol{y})$. Additionally, according to Assume 1 about $\boldsymbol{h}(\boldsymbol{x})$, the Lipschitz constant of $\boldsymbol{h}(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is denoted as L_h. As a result:

$$||\boldsymbol{z}_1 - \boldsymbol{x}_1|| \leq L^{(0)}||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{17}$$

where $L^{(0)} = (L_{fx} + L_{fy} \cdot L_{\pi x} + L_{fy} \cdot L_{\pi y} \cdot L_h)$.

Emulate the process of (13) to (17) through Assume 1, 2, and Lemma 1, we can obtain that:

$$||\hat{\boldsymbol{z}}_1 - \hat{\boldsymbol{x}}_1|| \leq \hat{L}^{(0)}||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{18}$$

where $\hat{L}^{(0)} = (L_{fx} + L_{fy} \cdot L_{\pi x} + L_{fy} \cdot L_{\pi y} \cdot L_h + 2L_{\phi y} \cdot L_h + L_{\phi x} + L^{(0)})$.

Based on Eq.(13), (16), and (17), we obtain that:

$$||f_d(\tilde{\boldsymbol{z}}_1) - f_d(\boldsymbol{\xi}_1)|| \leq (L^{(0)} + \hat{L}^{(0)})||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{19}$$

It is naturally get:

$$||g(\tilde{\boldsymbol{z}}_1) - g(\boldsymbol{\xi}_1)|| \leq L_v(L^{(0)} + \hat{L}^{(0)})||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{20}$$

Induction hypothesis: When $t = \tau > 0$, assume that $\exists L^{(\tau)}\ , \hat{L}^{(\tau)} > 0$ s.t.

$$||\boldsymbol{z}_\tau - \boldsymbol{x}_\tau|| \leq L^{(\tau)}||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{21a}$$

$$||\hat{\boldsymbol{z}}_\tau - \hat{\boldsymbol{x}}_\tau|| \leq \hat{L}^{(\tau)}||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{21b}$$

$$||f_d(\tilde{\boldsymbol{z}}_\tau) - f_d(\boldsymbol{\xi}_\tau)|| \leq (L^{(\tau)} + \hat{L}^{(\tau)})||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{21c}$$

Inductive step: When $t = \tau + 1$,

$$\begin{aligned} &||f_d(\tilde{\boldsymbol{z}}_{\tau+1}) - f_d(\boldsymbol{\xi}_{\tau+1})|| = \\ &|| \begin{bmatrix} \boldsymbol{z}_{\tau+1} - \boldsymbol{x}_{\tau+1}, \\ \hat{\boldsymbol{z}}_{\tau+1} - \hat{\boldsymbol{x}}_{\tau+1} - (\boldsymbol{z}_{\tau+1} - \boldsymbol{x}_{\tau+1}) \end{bmatrix} ||, \\ \le &||\boldsymbol{z}_{\tau+1} - \boldsymbol{x}_{\tau+1}|| + ||\hat{\boldsymbol{z}}_{\tau+1} - \hat{\boldsymbol{x}}_{\tau+1} - (\boldsymbol{z}_{\tau+1} - \boldsymbol{x}_{\tau+1})|| \end{aligned} \tag{22}$$

As the same process as the Eq.(16) to (19), we have:

$$||\boldsymbol{z}_{\tau+1} - \boldsymbol{x}_{\tau+1}|| \le L||\boldsymbol{z}_\tau - \boldsymbol{x}_\tau||_\infty, \tag{23a}$$

$$||\hat{\boldsymbol{z}}_{\tau+1} - \hat{\boldsymbol{x}}_{\tau+1}|| \le \hat{L}||\boldsymbol{z}_\tau - \boldsymbol{x}_\tau||_\infty \tag{23b}$$

where $L = (L_{fx} + L_{fy} \cdot L_{\pi x} + L_{fy} \cdot L_{\pi y} \cdot L_h)$, $\hat{L} = (L_{fx} + L_{fy} \cdot L_{\pi x} + L_{fy} \cdot L_{\pi y} \cdot L_h + 2L_{\phi y} \cdot L_h + L_{\phi x} + L)$, because of induction (21a) and (21b), $\exists L^{(\tau+1)},\ \hat{L}^{(\tau+1)} > 0$, s.t.

$$||g(\tilde{\boldsymbol{z}}_{\tau+1}) - g(\boldsymbol{\xi}_{\tau+1})|| \le L_v(L^{(\tau+1)} + \hat{L}^{(\tau+1)})||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty \tag{24}$$

where $L^{(\tau+1)} = L \cdot L^{(\tau)}$, $\hat{L}^{(\tau+1)} = \hat{L} \cdot \hat{L}^{(\tau)}$.

In conclusion, $\exists L^{(t)},\ \hat{L}^{(t)}, L_v > 0$, $\forall t \in \mathbb{N}$, $\boldsymbol{x}_0,\ \boldsymbol{z}_0 \in \mathcal{B}$, s.t. $||g(\tilde{\boldsymbol{z}}_t) - g(\boldsymbol{\xi}_t)|| \le L_v(L^{(t)} + \hat{L}^{(t)})||\boldsymbol{z}_0 - \boldsymbol{x}_0||_\infty$.

Theorem 2: (Robustness evaluation for NNCS) Given a p-norm ball $\mathbb{B}_p(\boldsymbol{x}_0, \delta) = \{\boldsymbol{x} \in \mathcal{A} | ||\boldsymbol{x} - \boldsymbol{x}_0||_p \le \delta\}$, the NNCS in Theorem 1, barrier function $g(\boldsymbol{\xi}_t) = \rho - V(\boldsymbol{\xi}_t)$ within limited time steps t, L_q is the Lipschitz constant of $g(\boldsymbol{\xi}_0)$ with respect to $\boldsymbol{x}_0 \in \mathcal{A}$. If perturbation $\Delta_p = ||\delta||_p$ satisfies:

$$||\delta||_p \le \frac{\rho - V(\boldsymbol{\xi}_t)}{L_q}, \tag{25}$$

then perturbed state $\boldsymbol{z}_t = f_d^{(t)}(\boldsymbol{z}_0) \in \mathcal{S}$ holds ture, where $\mathcal{S} \subseteq \mathcal{A}$, $\boldsymbol{z}_0 \in \mathbb{B}_p(\boldsymbol{x}_0, \delta)$. The value of $\Delta^L_{p,min} \le min_{\boldsymbol{z}_0 \ne \boldsymbol{x}_0} \frac{\rho - V(\boldsymbol{\xi}_t)}{L_q}$ is the robustness evaluation in REN.

Proof 2: Because of Theorem 1, $\exists L_q > 0$, s.t.

$$|g(\tilde{\boldsymbol{z}}_t) - g(\boldsymbol{\xi}_t)| = |V(\tilde{\boldsymbol{z}}_t) - V(\boldsymbol{\xi}_t)| \le L_q \cdot ||\boldsymbol{z}_0 - \boldsymbol{x}_0||_p \tag{26}$$

After decomposing the absolute value in (26), we have:

$$-L_q||\boldsymbol{z}_0 - \boldsymbol{x}_0||_p \le V(\tilde{\boldsymbol{z}}_t) - V(\boldsymbol{\xi}_t) \le L_q||\boldsymbol{z}_0 - \boldsymbol{x}_0||_p \tag{27}$$

For $\boldsymbol{z}_t \in \mathcal{S}$ to be true, there must be

$$V(\tilde{\boldsymbol{z}}_t) \le L_q||\boldsymbol{z}_0 - \boldsymbol{x}_0||_p + V(\boldsymbol{\xi}_t) \le \rho \tag{28}$$

(28) implies the (25) holds. Because the distortion from (25) ensures $\boldsymbol{z}_t \in \mathcal{S}$ and $\mathcal{S} \in \mathcal{A}$, the distortion from 25 is the $\Delta^L_{p,min}$.

IV. REN evaluation via different distributions

Recall Section III, we demonstrated that the lower bound on minimum distortion, given by $min_{\boldsymbol{z}_0 \ne \boldsymbol{x}_0} \frac{\rho - V(\boldsymbol{\xi}_t)}{L_q}$, is related to the value function $V(\boldsymbol{\xi}_t)$ and its maximum cross-Lipschitz constant L_q, as expressed in (25). The maximum L_q can be determined using the expression $max_{\boldsymbol{z}_0 \in \mathbb{B}_p(\boldsymbol{x}_0, \delta)} || \frac{V(\tilde{\boldsymbol{z}}_t) - V(\boldsymbol{\xi}_t)}{\boldsymbol{z}_0 - \boldsymbol{x}_0} ||_q$, which is detailed in (24). Obtaining the maximum value of L_q within a norm ball for the larger-scale neural networks of interest poses a challenge. This is because it is hard for L_q to be readily determined through a single backpropagation or exhaustive search [10].

To compute L_q in REN, we initially sample a pure state $\boldsymbol{x}_0 \in \mathcal{A}$ and satisfies $\rho - V(\boldsymbol{\xi}_t) > 0$ as required by Equation (24), we proceed to sample a collection of n perturbed states $\boldsymbol{z}_0^{(i)} \in \mathbb{B}_p(\boldsymbol{x}_0, \delta)$, where $i \in [|n|]$ and δ represents the prescribed perturbation radius. Subsequently, we compute the value of L_q from the pure and perturbed samples. However, obtaining a reliable estimate of $max_{\boldsymbol{z}_0 \ne \boldsymbol{x}_0} L_q$ necessitates a substantial number of samples, and it remains challenging to ascertain whether the estimated L_q corresponds to the true maximum value. Methods proposed by Wood&Zhang [43] and Weng et al. [10] suggest employing Extreme Value Theory to estimate the distribution of L_q, followed by maximum likelihood estimation to determine its value. Nonetheless, a prerequisite for the Extreme Value Theory is that the distribution of samples must be non-degenerate, a condition that is not always satisfied in the NNCS of interest, as demonstrated in Sec V-B.

In this section, we prove and establish that the Lipschitz constant proven in Theorem 1 adhere to two distinct distributions contingent upon the time step t in Sec IV-A. Furthermore, we delineate an algorithm for REN framework in Sec IV-B.

A. Estimate L_q via two different distributions

In Theorem 3, we establish that the L_q distribution adheres to two distinct forms: a non-degenerate cumulative distribution and a One-point distribution. Subsequently, we present the Fisher-Tippett-Gnedenko Theorem in Lemma 4 to facilitate the estimation of $max_{\boldsymbol{z}_0 \ne \boldsymbol{x}_0} L_q$ under the non-degenerate cumulative distribution.

Lemma 2: Given the NNCS, and barrier function $g(\boldsymbol{\xi}_t) = \rho - V(\boldsymbol{\xi}_t)$ in Theorem 1. $g(\boldsymbol{\xi}_t)$ is differentiable almost everywhere for $\boldsymbol{x}_0 \in \mathcal{B}$, $\forall t \in \mathbb{N}$.

Proof 3: Because of Assume 1, it is only necessary to prove that $\boldsymbol{\xi}_{t+1} = f_d(\boldsymbol{\xi}_t)$ is differentiable almost everywhere for $\boldsymbol{x}_0$ i.e., the Jacobi matrix $\nabla_{\boldsymbol{x}_0} \boldsymbol{\xi}_{t+1}$ exists almost everywhere. It further implies that $\nabla_{\boldsymbol{x}_0} \boldsymbol{x}_{t+1}$ and $\nabla_{\boldsymbol{x}_0} \hat{\boldsymbol{x}}_{t+1}$ exists almost everywhere.

Base case: When $t = 1$,

$$\nabla_{\boldsymbol{x}_0} \boldsymbol{x}_1 = \nabla_{\boldsymbol{x}_0} f_d + \nabla_\pi f_d \cdot \nabla_{\boldsymbol{x}_0} \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0)), \tag{29a}$$

$$\begin{aligned} \nabla_{\boldsymbol{x}_0} \hat{\boldsymbol{x}}_1 = &\nabla_{\hat{\boldsymbol{x}}_0} f_d \cdot \nabla_{\boldsymbol{x}_0} \hat{\boldsymbol{x}}_0 + \nabla_\pi f_d \cdot \nabla_{\boldsymbol{x}_0} \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0)) \\ &+ \nabla_{\boldsymbol{x}_0} \phi_{obs}(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0) - \boldsymbol{h}(\hat{\boldsymbol{x}}_0)) - \nabla_{\boldsymbol{x}_0} \phi_{obs}(\hat{\boldsymbol{x}}_0, \boldsymbol{0}_{n_{\boldsymbol{y}}}). \end{aligned} \tag{29b}$$

The derivative of neural networks $\nabla_{\boldsymbol{x}_0} \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0))$ has the form based on Assume 2:

$$\begin{aligned} \nabla_{\boldsymbol{x}_0} \pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0)) = &\boldsymbol{W}_{\pi 2} \cdot II(\boldsymbol{W}_{\pi 1}[:, : n_{\boldsymbol{x}}]\hat{\boldsymbol{x}}_0 + \boldsymbol{b}_{\pi 1} \\ &+ \boldsymbol{W}_{\pi 1}[:, n_{\boldsymbol{x}} :]\boldsymbol{h}(\boldsymbol{x}_0)) \cdot [\boldsymbol{W}^T_{\pi 1}[:, : n_{\boldsymbol{x}}] \cdot \nabla_{\boldsymbol{x}_0} \hat{\boldsymbol{x}}_0 \\ &+ \boldsymbol{W}^T_{\pi 1}[:, n_{\boldsymbol{x}} :] \cdot \nabla_{\boldsymbol{x}_0} h(\boldsymbol{x}_0)] \end{aligned} \tag{30}$$

where $II(\boldsymbol{x})$ is an univariate indicator function:

$$II(\boldsymbol{x}) = \begin{cases} 1, \text{ if } \boldsymbol{x} > 0, \\ 0, \text{ if } \boldsymbol{x} \leq 0. \end{cases} \tag{31}$$

Based on the Appendix D in [10], because the count of ReLU activation is limited, $\nabla_{\boldsymbol{x}_0}\pi(\hat{\boldsymbol{x}}_0, \boldsymbol{h}(\boldsymbol{x}_0))$ exists almost everywhere in $\boldsymbol{x}_0 \in \mathcal{B}$. The derivative of neural networks $\nabla_{\boldsymbol{x}_0}\phi_{obs}$ has the similar form as (30). Consequently, according to Lemma 1, Assume 1, and $\hat{\boldsymbol{x}}_0 = \boldsymbol{H}\boldsymbol{x}_0$ in [37], the (29a) and (29b) exist almost everywhere in $\boldsymbol{x}_0 \in \mathcal{B}$ when time step $t = 1$.

Induction hypothesis: When $t = \tau$, $\tau > 0$, assume that $\nabla_{\boldsymbol{x}_0}\boldsymbol{x}_\tau$ and $\nabla_{\boldsymbol{x}_0}\hat{\boldsymbol{x}}_\tau$ exists almost everywhere in $\boldsymbol{x}_0 \in \mathcal{B}$.

Inductive step: When $t = \tau + 1$,

$$\nabla_{\boldsymbol{x}_0}\boldsymbol{x}_{\tau+1} = (\nabla_{\boldsymbol{x}_\tau} f_d + \nabla_\pi f_d \cdot \nabla_{\boldsymbol{x}_\tau}\pi(\hat{\boldsymbol{x}}_\tau, \boldsymbol{h}(\boldsymbol{x}_\tau))) \cdot \nabla_{\boldsymbol{x}_0}\boldsymbol{x}_\tau, \tag{32a}$$

$$\begin{aligned} \nabla_{\boldsymbol{x}_0}\hat{\boldsymbol{x}}_{\tau+1} = (&\nabla_{\hat{\boldsymbol{x}}_\tau} f_d \cdot \nabla_{\boldsymbol{x}_\tau}\hat{\boldsymbol{x}}_\tau + \nabla_\pi f_d \cdot \nabla_{\boldsymbol{x}_\tau}\pi(\hat{\boldsymbol{x}}_\tau, \boldsymbol{h}(\boldsymbol{x}_\tau)) \\ &+ \nabla_{\boldsymbol{x}_\tau}\phi_{obs}(\hat{\boldsymbol{x}}_\tau, \boldsymbol{h}(\boldsymbol{x}_t) - \boldsymbol{h}(\hat{\boldsymbol{x}}_\tau)) \\ &- \nabla_{\boldsymbol{x}_\tau}\phi_{obs}(\hat{\boldsymbol{x}}_\tau, \boldsymbol{0}_{n_y})) \cdot \nabla_{\boldsymbol{x}_0}\hat{\boldsymbol{x}}_\tau. \end{aligned} \tag{32b}$$

According to Lemma 1, Assume 1, [37], and induction hypothesis, the (32a) and (32b) exist almost everywhere in $\boldsymbol{x}_0 \in \mathcal{B}$ when time step $t = \tau + 1$.

In conclusion, $g(\boldsymbol{\xi}_t)$ is differentiable almost everywhere for $\boldsymbol{x}_0 \in \mathcal{B}$, $\forall t \in \mathbb{N}$.

In the following, we mark the $\nabla_{\boldsymbol{x}_0} g(\boldsymbol{\xi}_t)$ as $\nabla g(\boldsymbol{\xi}_t)$ for convenience.

Lemma 3: ([44]) If $f(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is differentiable and continuous with respect to x, then its first-order derivative $f'(x)$ is continuous almost everywhere with respect to x. Lemma 3 is proved based on Baire category theorem [45] and Baire Class 1 functions [46] in real analysis.

Suppose n sampled $\{\nabla g(\tilde{\boldsymbol{z}}_t^{(i)})\} = \{||\frac{g(\tilde{\boldsymbol{z}}_t^{(i)}) - g(\boldsymbol{\xi}_t)}{\boldsymbol{z}_0 - \boldsymbol{x}_0}||_q\} = \{||\frac{V(\tilde{\boldsymbol{z}}_t^{(i)}) - V(\boldsymbol{\xi}_t)}{\boldsymbol{z}_0 - \boldsymbol{x}_0}||_q\}$. Denote them as a sequence of independent and identically distributed (iid) random variables $Y_t^{(i)}$, $i \in [|n|]$. $F_{Y_t^{(i)}}(y) = P(Y_t^{(i)} < y)$ is the CDF of $Y_t^{(i)}$. The Y_t is used when the property is satisfied by any $Y_t^{(i)}$. Notice that the sampled $\{\nabla g(\tilde{\boldsymbol{z}}_t^{(i)})\}$ can be interpreted as $||\nabla g(\boldsymbol{\xi}_t)||_p$, which represents the p-norm of the gradient of the function $g(\boldsymbol{\xi}_t) = \rho - V(\boldsymbol{\xi}_t)$ in following proofs.

Theorem 3: Considering the NNCS and the barrier function $g(\boldsymbol{\xi}_t) = \rho - V(\boldsymbol{\xi}_t)$ as stated in Theorem 1, the behavior of the $||\nabla g(\boldsymbol{\xi}_t)||_q$, $\boldsymbol{x}_0 \in \mathcal{A}$, varies depending on the convergence of $\boldsymbol{\xi}_t$ as t is finite. If the time step t lead to $\neg(\boldsymbol{\xi}_t \rightarrow \boldsymbol{\xi}^\star)$, $\forall \boldsymbol{x}_0 \in \mathcal{A}$, $||\nabla g(\boldsymbol{\xi}_t)||_q$ follows a non-degenerate distribution function, and it can be estimated using Maximum Likelihood Estimate (MLE). Conversely, if convergence occurs, $||\nabla g(\boldsymbol{\xi}_t)||_q$ adheres to a degenerate distribution function, precluding the use of MLE for estimation.

Proof 4: At first, we prove that $||\nabla g(\boldsymbol{\xi}_t)||_q$ follows a non-degenerate distribution function which is a cumulative distribution function (CDF) [47] while $\neg(\boldsymbol{\xi}_t \rightarrow \boldsymbol{\xi}^\star)$. It implies that the distribution function $F_{Y_t}(y)$ should satisfy limit property, boundedness, right-continuity, and monotonicity.

- Limit property: The $V(\xi)$ aheres to Lyapunov conditions (10b) and (10d) or belongs to a polynomial ring with respect to $\boldsymbol{x}$ in a closed-loop domain. All of the $\boldsymbol{z}_0 - \boldsymbol{x}_0$ in $\frac{V(\tilde{\boldsymbol{z}}_t) - V(\boldsymbol{\xi}_t)}{\boldsymbol{z}_0 - \boldsymbol{x}_0}$ are knowable. As a result, the limit property i.e., $lim_{y \rightarrow \infty} F_{Y_t}(y) = 1$ and $lim_{y \rightarrow -\infty} F_{Y_t}(y) = 0$ hold with both ROA and RA barrier functions.
- Right-continuity: Because $\boldsymbol{\xi}_t$ is differentiable and (Lipschitz) continuous almost everywhere to $\boldsymbol{x}_0$, proved in Theorem 1 and Lemma 2, $\nabla \boldsymbol{\xi}_t$ is continuous almost everywhere (i.e., right-continuous) to $\boldsymbol{x}_0$ according to Lemma 3. Consequently, the right-continuity $F_{Y_t}(y) = lim_{h \rightarrow 0} F_{Y_t}(y + h)$ holds under $\boldsymbol{x}_0 \in \mathcal{B}$, $y \in \mathbb{R}$ in limited time step t.
- Monotonicity: Because we have proven that $\nabla g(\boldsymbol{\xi}_t)$ is continuous almost everywhere to $\boldsymbol{x}_0 \in \mathcal{B}$, we can divide the state space $\mathcal{B}$ into limited n pieces, marked as $\mathcal{B}_i$, $i \in [|n|]$. In each of $\mathcal{B}_i$, $\nabla g(\boldsymbol{\xi}_t)$ is continuous to $\boldsymbol{x}_0$. The $P_{\mathcal{B}_i}(Y_t < y_1)$ can be denoted as the value of $Vol(\{\boldsymbol{x}_0 | Y_t < y_1,\ \boldsymbol{x}_0 \in \mathcal{B}_i\})/Vol(\mathcal{B})$, $Vol(S)$ is the volume of the set S. Due to the continuity in $\mathcal{B}_i$, there has:
$$\begin{aligned} &Vol(\{\boldsymbol{x}_0 | Y_t < y_1,\ \boldsymbol{x}_0 \in \mathcal{B}_i\}) \\ &\leq Vol(\{\boldsymbol{x}_0 | Y_t < y_2,\ \boldsymbol{x}_0 \in \mathcal{B}_i\}),\ \forall y_1 < y_2 \end{aligned} \tag{33}$$
which indicates $P_{\mathcal{B}_i}(Y_t < y_1) \leq P_{\mathcal{B}_i}(Y_t < y_2)$, $\forall y_1 < y_2$. Because $P(Y_t < y_1) = \sum_{i=0}^{i=n} P_{\mathcal{B}_i}(Y_t < y_1)$, then the monotonicity $P(Y_t < y_1) \leq P(Y_t < y_2)$, $\forall y_1 < y_2$ holds.
- Boundedness: Since the Limit property, Right-continuity and Monotonicity hold, the Boundedness can be derived by other three properties.

When $\neg(\boldsymbol{\xi}_t \rightarrow \boldsymbol{\xi}^\star)$, $F_{Y_t}(y) = P(Y_t < y)$ satisfies limit property, boundedness, right-continuity, and monotonicity, then $||\nabla g(\boldsymbol{\xi}_t)||_q$ (or samples Y_t) follows a non-degenerate cumulative distribution function. The value samples Y_t can be evaluated by MLE according to the Fisher-Tippett-Gnedenko Theorem [48].

When the time step t is large enough to guarantee $\boldsymbol{\xi}_t \rightarrow \boldsymbol{\xi}^\star$, the value of sampled $Y_t = ||\frac{V(\tilde{\boldsymbol{z}}_t) - V(\boldsymbol{\xi}_t)}{\boldsymbol{z}_0 - \boldsymbol{x}_0}||_q$. Because the time step t ensures $V(\tilde{\boldsymbol{z}}_t) - V(\boldsymbol{\xi}_t)$ converge to 0, $\boldsymbol{z}_0 \in \mathbb{B}_p(\boldsymbol{x}_0, \delta)$. However, the sampled finite value $\|\boldsymbol{z}_0 - \boldsymbol{x}_0\|$ does not converge to 0, the value of Y_t converge to 0, which adheres to an One-point distribution i.e., sampled Y_t converges to a constant $c \in \mathbb{R}$.

We demonstrate that the L_q distribution conforms to a non-degenerate cumulative distribution when the time step t is insufficiently large. Consequently, $max_{\boldsymbol{z}_0 \neq \boldsymbol{x}_0} L_q$ can be achieved through the application of extreme value theory, commonly referred to as the Fisher-Tippett-Gnedenko Theorem.

Lemma 4: (Fisher-Tippett-Gnedenko Theorem [48]) If there exists a sequence of pairs of real numbers (a_n, b_n)

such that $a_n > 0$ and $lim_{n\to\infty}F_Y^{(n)} = G(y)$, where $G(y)$ is a non-degenerate cumulative distribution function, then G must correspond to one of the following types of CDFs:

$$Gumbel\ class\ (Type\ I):$$

$$G(y) = exp\{-exp\{-\frac{y-a_W}{b_W}\}\}$$

$$Frechet\ class\ (Type\ II):$$

$$G(y) = \begin{cases} 0, \text{ if } y < a_W, \\ exp\{-\frac{y-a_W}{b_W}\}^{-c_W}, \text{ if } y \geq a_W. \end{cases}$$

$$Reverse\ Weibull\ class\ (Type\ III):$$

$$G(y) = \begin{cases} exp\{-\frac{y-a_W}{b_W}\}^{-c_W}, \text{ if } y < a_W, \\ 1, \text{ if } y \geq a_W. \end{cases}$$

where $a_W \in \mathbb{R}$, $b_W > 0$, and $c_W > 0$ are the location, scale, and shape parameter, respectively.

Because the CDF of $L_q = ||\nabla g(\boldsymbol{\xi})||_q$ is right-bounded, as proved in the limit property of Theorem 3, we postulate that $max_{\boldsymbol{z}_0 \neq \boldsymbol{x}_0} L_q$ adheres to the Reverse Weibull class. This is because the CDF of the Reverse Weibull distribution also has a right endpoint, which corresponds to the extreme value. This extreme value represents the unknown local cross Lipschitz constant $max_{\boldsymbol{z}_0 \neq \boldsymbol{x}_0} L_q$ that we aim to estimate in REN. When the time step t is sufficiently large, all sampled L_q converge to the same constant c as they adhere to a One-point distribution.

We can find that if a NNCS satisfies the Lemma 1, 2, 3, and Assume 1, 2, the lower bound on the minimum distortion for NNCS can be determined using the REN framework.

B. Algorithm for REN

To estimate robustness evaluation of REN and the $max_{\boldsymbol{z}_0 \neq \boldsymbol{x}_0} L_q$ for NNCS, we provide the algorithm 1. Given a NNCS which satisfies Assume 1 and 2, the input state of interest region $\mathcal{B}$, a K-S test confidence interval α to detect whether the L_q adhere to a non-degenerate CDF or an One-point distribution, barrier function of NNCS, and the maximum perturbation R with the norm ball $\mathbb{B}_p(\boldsymbol{x}_0, R)$.

We initially generate N_b batches of perturbed states. Within each batch, we sample N_s instances of $\boldsymbol{z}_0$. Uniformly and randomly select a pure data $\boldsymbol{x}_0 \in \mathcal{A}$ within $g(\boldsymbol{\xi}_t) > 0$ in the set time steps t (line 5). For the initial state $\boldsymbol{x}_0$, if $\boldsymbol{x}_0 \in \mathcal{S}$, we compute the minimum distance between $\boldsymbol{x}_0$ and barrier function $g(\boldsymbol{x})$ by optimization (line 8 to 10). To ensure that $\boldsymbol{z}_t \in \mathcal{S}$, we check the condition $g(\tilde{\boldsymbol{z}}_t) < 0$. If this condition is not met, the perturbed sample is resampled (line 15). Subsequently, we compute $||\nabla g(\tilde{\boldsymbol{z}}_t^{(ik)})||_q$ according to distinct distributions. The sampled $L_q = ||\nabla g(\tilde{\boldsymbol{z}}_t^{(ik)})||_q$ is calculated as a candidate maximum Lipschitz constant in line 17. Subsequently, we extract the maximum value from the candidates in line 19. Thereafter, if the N_b sampled Lipschitz constants in K adhere to Reverse Weibull contribution, the result of K-S is more than α. We perform maximum likelihood estimation of the Reverse Weibull distribution parameters, with the location parameter $\hat{a}_W$ serving as a potential estimate for $max_{\boldsymbol{z}_0 \neq \boldsymbol{x}_0} L_q$ (line 22). It is important to note that Lemma 4 is applicable when the count of samples is sufficient. In cases where the number of samples is inadequate or the samples adhere to One-point contribution, we resort to peak detection (line 24). Ultimately, we use the maximum L_q to compute the robustness estimate in REN (lines 26). Notably, there is no scenario where samples from two different distributions would be mixed together due to the line 19 Algorithm.1

Algorithm 1 Algorithm of REN

```
1: Input: the NNCS which satisfies Assume 1 and 2, the input state of interest region $\mathcal{B}$, equilibrium state $\boldsymbol{x}^\star$, perturbation norm $p$, K-S test confidence interval $\alpha$, barrier function $g(\boldsymbol{\xi}) = \rho - V(\boldsymbol{\xi})$, batch size $N_b$, number of samples per batch $N_s$, and maximum perturbation $R$.
2: Output: REN robustness evaluation $\mu \in \mathbb{R}^+$ which is the lower bound on the minimum distortion.
3: initialize: $K \leftarrow \{\emptyset\}$, $q \leftarrow \frac{p}{p-1}$
4: while $\boldsymbol{x}_0$ is $\emptyset$ or $g(\boldsymbol{\xi}_t) < 0$ do
5:     uniform and ramdom select the pure sample: $\boldsymbol{x}_0 \in \mathcal{B}$
6: end while
7: if $g(\boldsymbol{\xi}_0) > 0$ then
8:     Inner-approximate the minimum distortion in $\mathcal{S}$: $L \leftarrow \mathrm{D}(\boldsymbol{x}_0, g(\boldsymbol{x}))$
9: else
10:    $L \leftarrow NaN$
11: end if
12: for $i \leftarrow 1$ to $N_b$ do
13:    for $k \leftarrow 1$ to $N_s$ do
14:        while $\boldsymbol{z}_0^{(ik)}$ is $\emptyset$ or $g(\tilde{\boldsymbol{z}}_t) < 0$ do
15:            ramdom select the perturbed sample: $\boldsymbol{z}_0^{(ik)} \in \mathbb{B}_p(\boldsymbol{x}_0, R)$
16:        end while
17:        compute robustness evaluations: $b_{ik} \leftarrow ||\nabla g(\tilde{\boldsymbol{z}}_t^{(ik)})||_q$
18:    end for
19:    Compile maximum Lipschitz constant: $K \leftarrow K \cup \{max\{max_k\{b_{ik}\}, L\}\}$
20: end for
21: if $ks\_test(K) \geq \alpha$ then
22:    $\hat{a}_W \leftarrow$ MLE $a_W$ in Type III distribution on $K$
23: else
24:    $\hat{a}_W \leftarrow$ peak detection on $K$
25: end if
26: $\mu \leftarrow min(\frac{g(\boldsymbol{\xi}_t)}{\hat{a}_W}, R)$
```

V. Experiments

We develope a framework, termed Robustness Evaluation for NNCS (REN), to estimate the lower bound of minimum distortion in NNCS. The existing studies on ROA provides a comprehensive setup for both the state and output secnarios, and is notably general. Consequently, our experiments mainly employ the models and scenarios derived from ROA studies [32]. We first delineate the configurations in Exp V-A. In Exp V-B, we demonstrate that the Reverse Weibull and the One-point distributions adequately fit the empirical distributions of the cross Lipschitz constant L_q. Subsequently, in Exp V-C, we involve a difference experiment to distinguish the REN robustness evaluation from the robustness evaluation computed by barrier functions. Finally, in Exp V-D, we compare the lower bound provided by REN with the minimum distortions achieved by baselines to show the soundness and applications of REN.

A. Experiment Configurations

a) Networks and tasks: The assessment is carried out on three distinct tasks: pathtracking [49], pendulum [50], and 2D quadrotor [51]. These tasks are incorporated into VNN-COMP 2024 [52] as competitive scenarios for participants. All the NNCS come from [32] with multiple activate functions. Their specifics are delineated in Table I. The neurons and layer imply the number of neurons and layers in one time interval. As time step t increases, so do the total neurons and layers of the whole NNCS. In the following experiments, all of the specifications such as state domain $\mathcal{B}$ come from the spec.file in [32]. Because [32] contains that it is better to calculate l_∞ norm ($p = \infty$) for PGD and $\alpha\beta$-CROWN, we analysis the minimum distortion with $p = \infty$ in the following experiments. All samples use the same random seed.

b) Baselines: We employ iterative-based attack methodologies, projected gradient descent (PGD) [8], alongside the SOTA network verifier $\alpha\beta$-CROWN, to determine the minimum distortion of NNCS. Iterative-based attack techniques are predominantly designed for classification networks [8], [53]. In the [32], PGD is applied to the NNCS setting to generate adversarial examples for training. The generability of abcrown leads to its use in conjunction with PGD in [32] to identify adversarial examples within the NNCS domain. The timeout settings are setted as 360000s for verification. The minimum distortion lower bound, computed via the barrier function associated with the inner-approximation $\mathcal{S} \in \mathcal{A}$, referred to the robustness evaluation L for NNCS. Similarly, the one computed by Lipschitz constant proven in our work is referred to REN robustness evaluation.

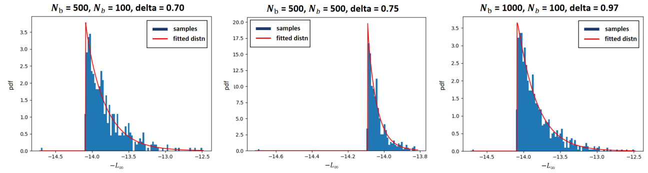
c) Machine and Software: All experiments are conducted on a 40-core 2.20GHz Intel Xeon Silver 4210 platform with 128GB memory and NVIDIA GeForce RTX 3090 platform with 24GB video memory. REN is implemented in Python 3.10.

TABLE I: Models used in the experiments. All the models are trained through PGD [8] and $\alpha\beta$-CROWN [54] based on Lyapunov conditions. The Neurons and Layers imply the number of neurons and layers in one time interval. The terms "-state" or "-output" in Model denotes that the scenario of the task involves state feedback or output feedback. The 'small' means that the NNCS is trained with a small torque.

Model	Neurons	Layers
Pathtracking-state	51	5
Pathtracking-state(small)	51	5
Pendulum-state	51	5
Pendulum-state(small)	51	5
Pendulum-output	73	8
Quadrotor2d-state	24	3
Quadrotor2d-output	54	6

B. Validation of distinct distributions

To validate Reverse Weibull and the One-point distributions adequately fit the empirical distributions of the cross Lipschitz constant L_q based on various value of t, we conduct Kolmogorov-Smirnov goodness-of-fit test (K-S test [55]) to estimate the fittness through p-values based on various batch size N_b, number of samples per batch N_s, and time step t. The time interval is 0.05 second as the same as the one in Yang et al. [32]. If $p\_value \geq \alpha$, $\alpha = 0.95$, the Reverse Weibull distribution hypothesis cannot be rejected. Conversely, it implies that almost of the samples adhere to One-point distribution. The Fig 3 shows the results of K-S test in pathtracking within various N_b, N_s, and the constant t. Specifically, Figures (a), (b), and (c) display the K-S test results for samples with various $N_b \times N_s$ (500×100, 500×500, and 1000×100).

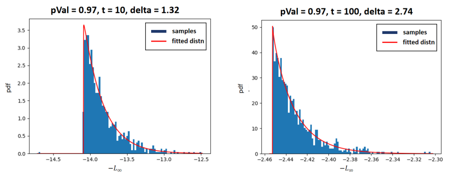


(a) (b) (c)

Fig. 3: The cross Lipschitz constant sampled in the pathtracking tasks, varying the batch size (N_b), the number of samples per batch (N_s), while maintaining a constant time step ($t = 10$). The values of N_b, N_s, and p-values (pVal) from K-S test are presented atop each subfigure. The blue parts represent the empirical distribution of the Lipschitz constants, whereas the red curve denotes the theoretical Reverse Weibull distribution.

It should be noted that $t = 10$ was selected to ensure that the sampled states never converge to the equilibrium in pathtracking tasks. The figures indicate that as both N_b and N_s increase, the sampled L_q values are more likely (pVal $=$ 0.70, 0.75, and 0.97, respectively) to follow the Reverse Weibull distribution. This result supports the assumption stated in Lemma 4, which posits that the sample size must be sufficiently large. Furthermore, the analysis reveals that the effect of N_b on the pVal is more pronounced than that of N_s. Consequently, to enhance the accuracy of the REN while maintaining efficiency, we prefer to increasing in N_b rather than N_s.

Figure 4 presents the K-S test for pathtracking as $t \in \{10, 100, 10000\}$, while keeping $N_b \times N_s = 1000 \times 100$. Figure 4 (a) to (c) confirm Theorem 3, which posits that the number of time steps t can alter the distribution of the Lipschitz constant in NNCS. As t increases, the robustness evaluation (delta) also increases, indicating that NNCS exhibit greater robustness at larger t. This increased robustness is attributed to the ability of NNCS to partially rectify perturbations within each time interval. Otherwise, the result in PathTracking-state at $t = 10000$ also demonstrate the application of REN on the large networks (about 510000 neruons).



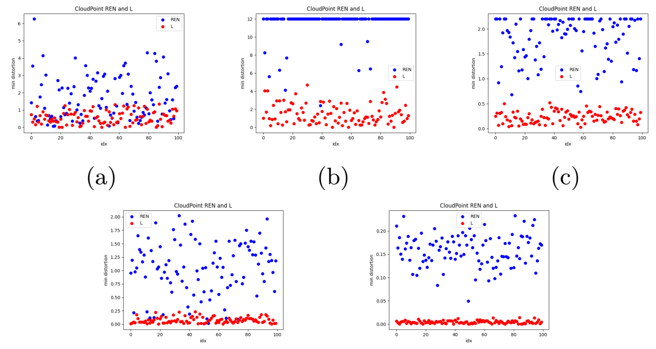
(a) (b) (c)

Fig. 4: The cross Lipschitz constant sampled in the pathtracking tasks, varying the time step t, while maintaining a constant $N_b \times N_s$. The values of t, the p-values (pVal) from K-S test, and robustness evaluation of REN (delta) are presented atop each subfigure.

C. Comparison between robustness evaluation from REN and barrier functions

To differentiate the REN robustness evaluation from that computed via barrier functions, we compare the lower bounds of minimum distortion they produce among 100 samples in Figure 5, which displays cloudpoint images. The cloudpoint figures (a) to (e) correspond to Pathtracking-state, Pendulum-state, Pendulum-output, Quadrotor2d-state, and Quadrotor2d-output, respectively. Red points represent the L robustness evaluation, while blue points denote the REN robustness evaluation. The horizontal axis indexes the samples, and the vertical axis denotes the lower bound on the minimum distortion.

In Figure 5, the REN robustness evaluation is observed to outperform the L robustness evaluation across the majority of tasks. Specifically, the REN robustness evaluation exceeds the L robustness evaluation for 97, 99, 100, 99, and 100 samples, respectively, in each task. Notably, barrier functions were unable to handle 3 samples in the Pathtracking-state task. One reason for this difference is



(a) (b) (c)

(d) (e)

Fig. 5: Both REN and L robustness evaluations are derived from the same set of 100 samples. Red points represent the L robustness evaluation, while blue points denote the REN robustness evaluation. The horizontal axis indexes the samples, and the vertical axis denotes the lower bound on the minimum distortion.

that the state considered in the L robustness evaluation is limited to the barrier function corresponding to the inner-approximation, as opposed to the RA or ROA region which is considered in the REN robustness evaluation. Another reason is the difference in discussed perturbed sample $\boldsymbol{z}_0$ in $\|\boldsymbol{z}_0 - \boldsymbol{x}_0\|$. Specifically, the L robustness evaluation focuses on $\boldsymbol{z}_0$ in the inner-approximations from barrier functions, whereas the REN robustness evaluation considers $\boldsymbol{z}_t$ in the inner-approximation while $\boldsymbol{z}_0$ lies in the exact RA or ROA.

The experimental results indicate that the REN robustness evaluation outperforms the L robustness evaluation in the majority of tasks for NNCS. To enhance the tightness of the robustness evaluation provided by the REN framework, we propose a combination of REN with the L robustness evaluation, as detailed in Algorithm 1.

D. Comparison among REN, verification, and attack

In this study, we compare REN framework in algorithm 1 against the minimal distortions attained by the baseline methods with time step $t = 10$ among the same 100 samples. Table II facilitates a comparative analysis of average l_∞ distortions detected by the PGD, PGD[REN] and $\alpha\beta$-CROWN relative to REN robustness evaluations. PGD[REN] refers to the PGD within the REN framework to reduce its detection domain for identifying adversarial examples. We determine that the lower bound on minimum distortion corresponds to the maximum potential perturbation for unsuccessful attack instances. To detect the precise minimum distortion through verification $\alpha\beta$-CROWN as a standard, we utilize the dichotomy [56] which is a typical method for detection. The dichotomy iteration and threshold are setted as 30 and $1e^{-6}$, respectively, ensuring that the accuracy and efficiency of dichotomy. Term '-' means timeout issue.

As anticipated by comparison with $\alpha\beta$-CROWN, the results of attacks are regarded as the upper bound for the minimum distortion, our REN framework establishes a sound lower bound on minimum distortion. In Table II, the robustness estimation identified by PGD[REN] surpasses that of PGD by TODO: Number on average. Specifically, all cases that are unsuccessfully attacked by PGD in TODO: Tasks, obtaining the robustness evaluation as TODO: Number. Meanwhile, the value evaluated by PGD[REN] is TODO: Number which is closer to the exact minimum distortion attained by $\alpha\beta$-CROWN. The experimental findings suggest that the REN framework can provide a lower bound for minimum distortion, thereby reducing the scope of search for attacks and increasing the possibility of detecting tighter adversarial examples. Although PGD[REN] incurs an average time cost that is TODO: Number% higher than that of PGD, it consumes TODO: Number% less time on average compared to the total time of the REN framework and PGD attack. This reduction in time is attributable to the decreased scope of the research, which facilitates the detection of adversarial examples in some cases, then attcak terminates earliy.

TABLE II: Comparisons on average REN robustness evaluations and distortions fonund by PGD, PGD[REN] and $\alpha\beta$-CROWN. The term '-' means timeout issue.

Network	Method	Time(s)	δ
PathTracking-state	PGD	15.608	2.250
	PGD[REN]	-	2.031
	$\alpha\beta$-CROWN	1.253k	1.784
	REN	22.131	1.720
PathTracking-state(small)	PGD	15.608	2.250
	PGD[REN]	-	2.031
	$\alpha\beta$-CROWN	1.253k	1.784
	REN	22.131	1.720
Pendulum-state	PGD	22.293	12.000
	PGD[REN]	-	12.000
	$\alpha\beta$-CROWN	2.731k	12.000
	REN	23.951	11.459
Pendulum-state(small)	PGD	22.293	12.000
	PGD[REN]	-	12.000
	$\alpha\beta$-CROWN	2.731k	12.000
	REN	23.951	11.459
Pendulum-output	PGD	48.659	1.283
	PGD[REN]	-	1.007
	$\alpha\beta$-CROWN	-	-
	REN	33.586	0.341
Quadrotor2d-state	PGD	45.849	2.881
	PGD[REN]	-	2.093
	$\alpha\beta$-CROWN	3.085k	1.363
	REN	10.782	1.107
Quadrotor2d-output	PGD	57.631	0.450
	PGD[REN]	-	0.236
	$\alpha\beta$-CROWN	-	-
	REN	22.105	0.158

Furthermore, torque is a critical parameter in the training of controller systems. Utilizing a small torque can augment the expressivity, convergence, and interpretability of systems. However, it may impact the robustness of systems [32]. Comparing NNCS with their versions trained with small torque, we observe that the lower bound on minimum distortion achieved by REN in the PathTracking-state is TODO: Number greater than that in the PathTracking-state(small). This finding aligns with the fact presented by Yang et al. [32], which indicates that the $\mathcal{S}$ in PathTracking-state encompasses the $\mathcal{S}$ in PathTracking-state(small). The comparisons show the application of REN framework on evaluation for NNCS.

Moreover, we find that verification methods encountered instances of failure. For example, the $\alpha\beta$-CROWN reaches a timeout issue during the Quadrotor2d-output verification. This is because $\alpha\beta$-CROWN invokes much more rounds of the gradient calculation of the NNCS than attacks, and more complex non-linear functions incur a higher computational cost, requiring more time to compute gradients and perform the verification.

It is important to recognize the distinction in input requirements between NNCS equipped with observers and those without. For instance, while the inputs for the Quadrotor2d-state system encompass the quadrotor's complete position data, the inputs of Quadrotor2d-output system is limited to angle or lidar measurements, excluding location information which should be estimated by observers. This difference in inputs precludes a direct comparison of robustness using the minimum distortion of input between NNCS with and without observers.

VI. Conclusion

This paper proposes a framework named REN. Our work firstly determines one of the properties in Neural Network Controlled Systems (NNCS), its lower bound on minimum distortion, for all states of interest by leveraging the Lipschitz constants of NNCS. We establish the Lipschitz continuity of NNCS and investigate the different distributions adhered to by the Lipschitz constants. Experimental results validate that the REN framework provides the sound and sufficiently tight lower bound on minimum distortion for NNCS with theoretical guarantees and practical applicability. The REN framework enables us to contemplate the training, evaluation, enhanced attak or defence, and other applications on NNCS, as well as the various ones on classification NNs, with REN as a benchmark, in future works. We also hope for a mature dataset to standardize future works in NNCS, like MNIST and CIFAR-10 for classification NNs.

References

[1] J. Pacheco, S. Satam, S. Hariri, C. Grijalva, and H. Berkenbrock, "Iot security development framework for building trustworthy smart car services," in 2016 IEEE Conference on Intelligence and Security Informatics (ISI). IEEE, 2016, pp. 237–242.

[2] H. Naveed, J. Grundy, C. Arora, H. Khalajzadeh, and O. Haggag, "Towards runtime monitoring for responsible machine learning using model-driven engineering," in Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, 2024, pp. 195–202.

[3] S. Morales, R. Clarisó, and J. Cabot, "A dsl for testing llms for fairness and bias," in Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, 2024, pp. 203–213.

[4] K. Chen, Y. Yang, B. Chen, J. A. H. López, G. Mussbacher, and D. Varró, "Automated domain modeling with large language models: A comparative study," in 2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS). IEEE, 2023, pp. 162–172.

[5] N. Ayoughi, S. Nejati, M. Sabetzadeh, and P. Saavedra, "Enhancing automata learning with statistical machine learning: A network security case study," in Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, 2024, pp. 172–182.

[6] T. Weber and S. Weber, "Model everything but with intellectual property protection-the deltachain approach," in Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, 2024, pp. 49–56.

[7] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Deep drone acrobatics," arXiv preprint arXiv:2006.05768, 2020.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017.

[9] N. Morgulis, A. Kreines, S. Mendelowitz, and Y. Weisglass, "Fooling a real car with adversarial traffic signs," arXiv preprint arXiv:1907.00374, 2019.

[10] T.-W. Weng, H. Zhang, P.-Y. Chen, J. Yi, D. Su, Y. Gao, C.-J. Hsieh, and L. Daniel, "Evaluating the robustness of neural networks: An extreme value theory approach," arXiv preprint arXiv:1801.10578, 2018.

[11] S. H. Silva and P. Najafirad, "Opportunities and challenges in deep learning adversarial robustness: A survey," arXiv preprint arXiv:2007.00753, 2020.

[12] A. Amini, G. Liu, and N. Motee, "Robust learning of recurrent neural networks in presence of exogenous noise," in 2021 60th IEEE Conference on Decision and Control (CDC). IEEE, 2021, pp. 783–788.

[13] S. Zhou, C. Liu, D. Ye, T. Zhu, W. Zhou, and P. S. Yu, "Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity," ACM Computing Surveys, vol. 55, no. 8, pp. 1–39, 2022.

[14] J. Wang, J. Ai, M. Lu, H. Su, D. Yu, Y. Zhang, J. Zhu, and J. Liu, "A survey of neural network robustness assessment in image recognition," arXiv preprint arXiv:2404.08285, 2024.

[15] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," arXiv preprint arXiv:2102.01356, 2021.

[16] Z. Zhao, G. Chen, T. Liu, T. Li, F. Song, J. Wang, and J. Sun, "Attack as detection: Using adversarial attack methods to detect abnormal examples," ACM Transactions on Software Engineering and Methodology, vol. 33, no. 3, pp. 1–45, 2024.

[17] M. Sathiyanarayanan, S. Mahendra, and R. B. Vasu, "Smart security system for vehicles using internet of things (iot)," in 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, 2018, pp. 430–435.

[18] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," Progress in Aerospace sciences, vol. 91, pp. 99–131, 2017.

[19] J. Saunders, S. Saeedi, and W. Li, "Autonomous aerial robotics for package delivery: A technical review," Journal of Field Robotics, vol. 41, no. 1, pp. 3–49, 2024.

[20] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust mpc and neural network control," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 3050–3057, 2020.

[21] M. Abu-Ali, F. Berkel, M. Manderla, S. Reimann, R. Kennel, and M. Abdelrahem, "Deep learning-based long-horizon mpc: robust, high performing, and computationally efficient control for pmsm drives," IEEE transactions on power electronics, vol. 37, no. 10, pp. 12 486–12 501, 2022.

[22] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," IEEE Transactions on Automatic Control, vol. 66, no. 2, pp. 794–801, 2020.

[23] Z. Liu, K. Peng, L. Han, and S. Guan, "Modeling and control of robotic manipulators based on artificial neural networks: a review," Iranian Journal of Science and Technology, Transactions of Mechanical Engineering, vol. 47, no. 4, pp. 1307–1347, 2023.

[24] R. Ivanov, T. Carpenter, J. Weimer, R. Alur, G. Pappas, and I. Lee, "Verisig 2.0: Verification of neural network controllers using taylor model preconditioning," in International Conference on Computer Aided Verification. Springer, 2021, pp. 249–262.

[25] C. Huang, J. Fan, X. Chen, W. Li, and Q. Zhu, "Polar: A polynomial arithmetic framework for verifying neural-network controlled systems," in International Symposium on Automated Technology for Verification and Analysis. Springer, 2022, pp. 414–430.

[26] M. Sha, X. Chen, Y. Ji, Q. Zhao, Z. Yang, W. Lin, E. Tang, Q. Chen, and X. Li, "Synthesizing barrier certificates of neural network controlled continuous systems via approximations," in 2021 58th ACM/IEEE Design Automation Conference (DAC). IEEE, 2021, pp. 631–636.

[27] B. Xue, N. Zhan, and M. Fränzle, "Inner-approximating reach-avoid sets for discrete-time polynomial systems," in 2020 59th IEEE Conference on Decision and Control (CDC). IEEE, 2020, pp. 867–873.

[28] B. Xue, R. Li, N. Zhan, and M. Fränzle, "Reach-avoid analysis for stochastic discrete-time systems," in 2021 American Control Conference (ACC). IEEE, 2021, pp. 4879–4885.

[29] C. Zhang, W. Ruan, and P. Xu, "Reachability analysis of neural network control systems," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 12, 2023, pp. 15 287–15 295.

[30] H. Dai, B. Landry, L. Yang, M. Pavone, and R. Tedrake, "Lyapunov-stable neural-network control," arXiv preprint arXiv:2109.14152, 2021.

[31] J. Wu, A. Clark, Y. Kantaros, and Y. Vorobeychik, "Neural lyapunov control for discrete-time systems," Advances in neural information processing systems, vol. 36, pp. 2939–2955, 2023.

[32] L. Yang, H. Dai, Z. Shi, C.-J. Hsieh, R. Tedrake, and H. Zhang, "Lyapunov-stable neural control for state and output feedback: A novel formulation," in Forty-first International Conference on Machine Learning.

[33] A. Chakraborty, P. Seiler, and G. J. Balas, "Nonlinear region of attraction analysis for flight control verification and validation," Control Engineering Practice, vol. 19, no. 4, pp. 335–345, 2011.

[34] Y.-J. Chen, M. Tanaka, K. Tanaka, and H. O. Wang, "Stability analysis and region-of-attraction estimation using piecewise polynomial lyapunov functions: Polynomial fuzzy model approach," IEEE Transactions on Fuzzy Systems, vol. 23, no. 4, pp. 1314–1322, 2014.

[35] M. Korda, D. Henrion, and C. N. Jones, "Controller design and region of attraction estimation for nonlinear dynamical systems," IFAC Proceedings Volumes, vol. 47, no. 3, pp. 2310–2316, 2014.

[36] A. Mauroy and I. Mezić, "Global stability analysis using the eigenfunctions of the koopman operator," IEEE Transactions on Automatic Control, vol. 61, no. 11, pp. 3356–3369, 2016.

[37] A. M. Lyapunov, "The general problem of the stability of motion," International journal of control, vol. 55, no. 3, pp. 531–534, 1992.

[38] J. J. Vidal, "Implementing neural nets with programmable logic," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 36, no. 7, pp. 1180–1190, 1988.

[39] D. Luenberger, “An introduction to observers,” IEEE Transactions on automatic control, vol. 16, no. 6, pp. 596–602, 1971.

[40] S. Meyn, Control systems and reinforcement learning. Cambridge University Press, 2022.

[41] J. Crary, Techniques of the Observer. MIT press Cambridge, MA, 1990.

[42] W.-H. Chen, J. Yang, L. Guo, and S. Li, “Disturbance-observer-based control and related methods—an overview,” IEEE Transactions on industrial electronics, vol. 63, no. 2, pp. 1083–1095, 2015.

[43] G. R. Wood and B. Zhang, “Estimation of the lipschitz constant of a function,” Journal of Global Optimization, vol. 8, pp. 91–103, 1996.

[44] A. C. Ponce and J. Van Schaftingen, “The continuity of functions with n-th derivative measure.”

[45] J. De Groot, “Subcompactness and the baire category theorem,” Indag. Math, vol. 25, pp. 761–767, 1963.

[46] A. S. Kechris and A. Louveau, “A classification of baire class 1 functions,” Transactions of the american mathematical society, vol. 318, no. 1, pp. 209–236, 1990.

[47] I. W. Burr, “Cumulative frequency functions,” The Annals of mathematical statistics, vol. 13, no. 2, pp. 215–232, 1942.

[48] R. L. Smith, “Extreme value theory,” Handbook of applicable mathematics, vol. 7, no. 437-471, p. 18, 1990.

[49] J. M. Snider et al., “Automatic steering methods for autonomous automobile path tracking,” Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08, 2009.

[50] S. Mori, H. Nishihara, and K. Furuta, “Control of unstable mechanical system control of pendulum,” International Journal of Control, vol. 23, no. 5, pp. 673–692, 1976.

[51] S. Bouabdallah and R. Siegwart, “Full control of a quadrotor,” in 2007 IEEE/RSJ international conference on intelligent robots and systems. Ieee, 2007, pp. 153–158.

[52] C. Brix, S. Bak, T. T. Johnson, and H. Wu, “The fifth international verification of neural networks competition (vnn-comp 2024): Summary and results,” arXiv preprint arXiv:2412.19985, 2024.

[53] C. Du, C. Huo, L. Zhang, B. Chen, and Y. Yuan, “Fast c&w: A fast adversarial attack algorithm to fool sar target recognition with deep convolutional neural networks,” IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2021.

[54] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter, “Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification,” Advances in Neural Information Processing Systems, vol. 34, pp. 29 909–29 921, 2021.

[55] F. J. Massey Jr, “The kolmogorov-smirnov test for goodness of fit,” Journal of the American statistical Association, vol. 46, no. 253, pp. 68–78, 1951.

[56] N. Jay, “Gender and dichotomy,” Feminist studies, vol. 7, no. 1, pp. 38–56, 1981.