Learning Neural Network Controllers with Certified Robust Performance via Adversarial Training

Neelay Junnarkar[1], Yasin Sonmez[1], Murat Arcak[1]

***Abstract*— Neural network (NN) controllers achieve strong empirical performance on nonlinear dynamical systems, yet deploying them in safety-critical settings requires robustness to disturbances and uncertainty. We present a method for jointly synthesizing NN controllers and dissipativity certificates that formally guarantee robust closed-loop performance using adversarial training, in which we use counterexamples to the robust dissipativity condition to guide training. Verification is done post-training using α,β-CROWN, a branch-and-bound-based method that enables direct analysis of the nonlinear dynamical system. The proposed method uses quadratic constraints (QCs) only for characterization of non-parametric uncertainties. The method is tested in numerical experiments on maximizing the volume of the set on which a system is certified to be robustly dissipative. Our method certifies regions up to $78\times$ larger than the region certified by a linear matrix inequality-based approach that we derive for comparison.**

I. Introduction

Neural network (NN) controllers have shown remarkable performance in complex control tasks, but their deployment in safety-critical applications is hindered by the lack of formal guarantees on closed-loop behavior. Unlike classical controllers whose stability and robustness properties can be analyzed with well-established tools, NN controllers are nonlinear and high-dimensional, and hence difficult to certify. While recent works have made progress in certifying nominal closed-loop stability for NNs [1], [2], [3], [4], guaranteeing performance under external disturbances and model uncertainty remains a critical but less explored problem [5].

Dissipativity theory [6], [7] provides a framework that generalizes Lyapunov stability to capture exactly these robustness requirements. By enforcing a dissipation inequality involving a storage function, which is analogous to a Lyapunov function, and a supply rate (e.g., ℓ_2-gain, passivity), dissipativity enables the certification of input-output properties such as disturbance attenuation in addition to properties such as asymptotic stability. We further consider robust dissipativity, which requires dissipativity to hold under model uncertainty. Thus, robust dissipativity extends certificates from stability to robust performance under both external disturbances and model uncertainty.

A widely used method for analyzing such nonlinear and uncertain systems is the framework of integral quadratic constraints (IQCs) [8], which abstracts nonlinearities with quadratic terms and renders conditions into computationally tractable linear matrix inequalities (LMIs) [9], [10]. IQCs have been used to characterize known activation functions of NNs [1], [5], [11], [12]. However, replacing known nonlinearities with bounding IQCs inherently introduces conservativeness. To overcome this, alternative techniques like Satisfiability Modulo Theories (SMT) [13], [14] and branch-and-bound (BaB) methods such as α,β-CROWN [3], [4], [15] have been utilized for tighter verification. Adversarial training [16], [17] has been used in conjunction with these tools to guide synthesis.

The main contribution of this paper is a method to jointly synthesize controllers and certificates that ensure closed-loop robust dissipativity of systems subject to non-parametric uncertainty. We leverage the complementary strengths of α,β-CROWN and QCs: α,β-CROWN is used for direct analysis of known nonlinear dynamics, and QCs are used for characterization of non-parametric model uncertainty. We define a notion of robust dissipativity compatible with bounded sets, derive conditions verifiable via α,β-CROWN, and propose an adversarial training algorithm for joint synthesis of the controller and dissipativity certificate. Final verification is done using α,β-CROWN. Benefits are demonstrated in two numerical examples in which, given a performance specification and a set describing the model uncertainty, controllers are trained to maximize the volume of the state space on which the closed-loop is certified to be robustly dissipative.

The rest of the paper is organized as follows. Section II gives background on modeling problems for verification with α,β-CROWN, on quadratic constraints, and robust dissipativity. Section III derives conditions to be verified with α,β-CROWN to ensure robust dissipativity of the closed-loop system, and derives analogous LMI conditions for comparison in numerical experiments. Section IV details the proposed method for training controllers and verifying robust dissipativity. Section V demonstrates the method in two numerical experiments, including comparison with the LMI-based approach.

A. Notation

$\mathbb{R}_{\geq 0}$ denotes the set of nonnegative real numbers, ℓ_2^n the set of square-summable sequences with elements in $\mathbb{R}^n$, and ℓ_{2e}^n the set of sequences with elements in $\mathbb{R}^n$ that are square-summable on $0,\dots,K$ for all $K \geq 0$. B_r^n denotes the ball in $\mathbb{R}^n$ of radius r centered at the origin. We drop the superscript for the dimension of the codomain when clear from context. $\Omega_{V,\rho}$ denotes the sublevel set $\{x|V(x) \leq \rho\}$ of V.

This work was supported in part by the NSF grant CNS-2111688.

[1]Neelay Junnarkar, Yasin Sonmez, and Murat Arcak are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (neelay.junnarkar@berkeley.edu, yasin_sonmez@berkeley.edu, arcak@berkeley.edu)

II. Background and Problem Setup

A. α,β-CROWN

α,β-CROWN enables verification of the specifications of the form $f(x) > 0$ for all $x \in \mathbb{R}^n$ in an interval $x_{\text{low}} \leq x \leq x_{\text{high}}$ (inequalities taken elementwise) by computing global lower bounds $\underline{f} \leq f(x)$. If $\underline{f} > 0$, then the specification $f(x) > 0$ is certified for all inputs in the domain. The procedure is highly parallelizable and runs on GPUs via the autoLiRPA library [18]. Various logical formulas can equivalently be written as $f(x) > 0$ by defining operations as follows:

$$\begin{aligned}
\max\{x, y\} &= \frac{1}{2}\big(x + y \\
&\quad + \mathrm{ReLU}(x - y) + \mathrm{ReLU}(y - x)\big) \\
\min\{x, y\} &= -\max\{-x, -y\} \\
x > 0 \wedge y > 0 &\iff \min\{x, y\} > 0 \\
x > 0 \vee y > 0 &\iff \max\{x, y\} > 0 \\
\neg(x > 0) &\iff -x \geq 0 \\
x > 0 \implies y > 0 &\iff \neg(x > 0) \vee (y > 0)
\end{aligned}$$

This enables specifications to be simply written as logical formulas.

B. Dissipativity

We consider performance requirements expressed with a robust notion of dissipativity that considers model uncertainty. We define robust dissipativity on subsets of state and external input spaces, allowing us to consider dissipativity on bounded sets that α,β-CROWN can verify.

Consider the system

$$\begin{aligned}
x_{k+1} &= F(x_k, w_k, d_k) \\
v_k &= G(x_k, w_k, d_k) \\
e_k &= H(x_k, w_k, d_k) \\
w_k &= \Delta(v)_k
\end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the system state, $v_k \in \mathbb{R}^{n_v}$ and $w_k \in \mathbb{R}^{n_w}$ are the inputs/outputs of the uncertainty Δ, $d_k \in \mathbb{R}^{n_d}$ is an exogenous input, $e_k \in \mathbb{R}^{n_e}$ is a performance output, $k \geq 0$, and the uncertainty Δ belongs to a set of operators $\mathbf{\Delta}$. Assume the system is well-posed: for any initial condition $x_0 \in \mathbb{R}^n$, $d \in \ell_{2e}$, and $\Delta \in \mathbf{\Delta}$, there exist unique sequences x, v, w, and e satisfying the system equations.

Definition 1 (Robust Dissipativity): Given $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{D} \subseteq \mathbb{R}^{n_d}$, and the class of uncertainty $\mathbf{\Delta}$, the system (1) is robustly dissipative on $(\mathcal{X}, \mathcal{D}, \mathbf{\Delta})$ with respect to a supply rate $s(d, e)$ if the following hold:

1) If $\Delta \in \mathbf{\Delta}, x_0 \in \mathcal{X}$, and $d_k \in \mathcal{D}$ for $k \geq 0$, then $x_k \in \mathcal{X}$ for $k \geq 0$.
2) There exists a storage function $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ with $V(0) = 0$ such that if $\Delta \in \mathbf{\Delta}, x_0 \in \mathcal{X}$, and $d_k \in \mathcal{D}$ for $k \geq 0$, then $V(x_K) - V(x_0) \leq \sum_{k=0}^{K-1} s(d_k, e_k)$ for $K \geq 1$.

The first condition is a robust forward invariance (RFI) condition, and the second condition is the performance condition expressed as a dissipation inequality [7]. Some typical supply rates are $\gamma^2\|d\|^2 - \|e\|^2$, which corresponds to an ℓ_2-gain bound of γ, and $d^\top e$, which corresponds to passivity.

C. Quadratic Constraints

An operator $\Delta : \ell_{2e} \to \ell_{2e}$ satisfies the quadratic constraint (QC) defined by $M = M^\top$ if

$$\begin{bmatrix} v_k \\ w_k \end{bmatrix}^\top M \begin{bmatrix} v_k \\ w_k \end{bmatrix} \geq 0$$

for all $v \in \ell_{2e}$, $k \geq 0$, and $w = \Delta(v)$. QCs of this form can describe properties including sector bounds, gain margins, and multiplication by matrices in a polytope [8], [19].

Given a set $\mathcal{M}$ of symmetric matrices, we define the uncertainty set $\mathbf{\Delta}(\mathcal{M})$ as the set of operators that satisfy the QC defined by M for all $M \in \mathcal{M}$. The simplest family of QCs is $\{\lambda M_0 | \lambda \geq 0\}$ for a fixed $M_0 = M_0^\top$.

III. Main Results

We present sufficient conditions, that can be verified with α,β-CROWN and with LMIs, to ensure robust dissipativity on bounded sets $\mathcal{X}$ and $\mathcal{D}$ for a set of uncertainties $\mathbf{\Delta}(\mathcal{M})$.

Theorem 1: Let $\mathcal{M}$ be a set of QCs, $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ be such that $V(0) = 0$, and $\rho > 0$. Then (1) is robustly dissipative with respect to a supply rate $s(d, e)$ on $(\Omega_{V,\rho}, B_{\overline{d}}^{n_d}, \mathbf{\Delta}(\mathcal{M}))$ if there exist $M_{\text{rfi}}, M_{\text{perf}} \in \mathcal{M}$ such that

$$z^\top M_{\text{rfi}} z \geq 0 \implies V(F(x, w, d)) \leq \rho \tag{2}$$

$$z^\top M_{\text{perf}} z \geq 0 \implies V(F(x, w, d)) - V(x) \leq s(d, e) \tag{3}$$

for all $x \in \Omega_{V,\rho}, w \in \mathbb{R}^{n_w}, d \in B_{\overline{d}}^{n_d}$ where $z = (G(x, w, d), w)$ and $e = H(x, w, d)$.

Proof: Consider a trajectory of the system with $\Delta \in \mathbf{\Delta}(\mathcal{M})$ such that $V(x_0) \leq \rho$ and $d_k \in B_{\overline{d}}^{n_d}$ for $k \geq 0$. Because $w = \Delta(v)$, it holds that $z_k^\top M z_k \geq 0$ for all $k \geq 0$ for all $M \in \mathcal{M}$. This implies, by (2) and induction, that $x_k \in \Omega_{V,\rho}$ for all $k \geq 0$. Therefore, (3) applies for each time step $k \geq 0$ of the trajectory. Summing the right-hand side of the implication in (3) over $k = 0, \ldots, K-1$ implies $V(x_K) - V(x_0) \leq \sum_{k=0}^{K-1} s(d_k, e_k)$ for $K \geq 1$. Therefore both conditions of Definition 1 are met, and (1) is robustly dissipative on $(\Omega_{V,\rho}, B_{\overline{d}}, \mathbf{\Delta}(\mathcal{M}))$. ∎

Remark 1: If the QCs in $\mathcal{M}$ hold locally in the region $\mathcal{V} = \bigcap_i \left\{(x, w, d) | v^\top L_i^\top L_i v \leq \overline{v}_i^2\right\}$ for some matrices (e.g. selection matrices) L_i and bounds $\overline{v}_i \in \mathbb{R}$ where $v = G(x, w, d)$, then we must ensure $\Omega_{V,\rho} \subseteq \mathcal{V}$. This containment holds if there exist $M_{\Delta i} \in \mathcal{M}$ such that

$$z^\top M_{\Delta i} z \geq 0 \implies v^\top L_i^\top L_i v \leq \overline{v}_i^2 \tag{4}$$

for all $x \in \Omega_{V,\rho}, w \in \mathbb{R}^{n_w}$, $d \in B_{\overline{d}}^{n_d}$, and L_i, where $z = (G(x, w, d), w)$.

A. Verification using α,β-CROWN

When $\Omega_{V,\rho}$ is bounded, the robust forward invariance and performance conditions in Theorem 1 are verified using α,β-CROWN by writing them as conditions $\varphi_{\text{rfi}}(x,w,d) > 0$ and $\varphi_{\text{perf}}(x,w,d) > 0$. The condition φ_{rfi} is constructed by first writing (2) as a logical formula, where $\xi = (x,w,d)$:

$$\Big((V(x) \le \rho) \wedge (d^\top d \le \overline{d}^2) \wedge (z^\top M_{\text{rfi}} z \ge 0) \wedge (\xi^\top \xi \ge \epsilon)\Big) \implies -V(F(x,w,d)) + \rho \ge 0$$

Note that α,β-CROWN verifies strict inequalities, so we exclude a small region around the origin using $\xi^\top \xi \ge \epsilon$ and $\epsilon \approx 10^{-3}$. This formula is converted into a function $\mathbb{R}^n \times \mathbb{R}^{n_w} \times \mathbb{R}^{n_d} \to \mathbb{R}$ using the constructions in Section II. A similar procedure is taken to construct φ_{perf}.

The boxes for x and d over which these conditions are verified are selected to contain $\Omega_{V,\rho}$ and $B_{\overline{d}}$, respectively. We describe the method of determining the box for x in Section IV. The box for d is $\{d \| \|d\|_\infty \le \overline{d}\}$.

To construct w—which is in general unbounded—for use in α,β-CROWN, we leverage the properties of the QCs that Δ satisfies to construct w through a differentiable transformation of x, d, and auxiliary parameters. Whether such a transformation exists depends on the QC and the structure of G. If Δ satisfies a gain bound $\gamma^2\|v_k\|^2 - \|w_k\|^2 \ge 0$ and v_k can be expressed in terms of x_k and d_k, then we parameterize w_k as $w_k = (\tilde{w}_1 \gamma \|v_k\| / \|\tilde{w}_2\|)\tilde{w}_2$ where $\tilde{w}_1 \in [-1,1]$ and $\|\tilde{w}_2\|_\infty \le 1$. This can be generalized to G that are affine in w under some additional conditions. We label the domain of the parameters $\tilde{w}$ of w as $\tilde{\mathcal{W}}$. Now the conditions φ_{rfi} and φ_{perf} have domain $\mathcal{X} \times \tilde{\mathcal{W}} \times \mathcal{D}$. We refer to the conditions φ_{rfi} and φ_{perf} as functions of x, w, and d for simplicity of notation.

Parameterizations of w such as the one presented above ensure $z^\top M z \ge 0$ by construction, so the term $z^\top M z \ge 0$ may be removed from $\varphi_{\text{rfi}}(x,w,d)$ and $\varphi_{\text{perf}}(x,w,d)$. For more general sets of QCs $\mathcal{M}$, it may be necessary to assume a bound on w such that $w^\top w \le \overline{w}^2$, sample from the box containing this ball, and integrate this condition into Theorem 1 in a manner similar to the bound on disturbance.

B. Verification using LMIs

For comparison with the use of α,β-CROWN, we now present a QC-based method to analyze robust stability, characterizing both uncertainties and nonlinearities with QCs. Consider the reformulation of (1) as the interconnection of an LTI system and nonlinearities and uncertainties gathered in $\tilde{\Delta}$:

$$\begin{aligned} x_{k+1} &= Ax_k + B_w\tilde{w}_k + B_d d_k \\ \tilde{v}_k &= C_v x_k + D_{vw}\tilde{w}_k + D_{vd} d_k \\ e_k &= C_e x_k + D_{ew}\tilde{w}_k + D_{ed} d_k \\ \tilde{w}_k &= \tilde{\Delta}(\tilde{v})_k \end{aligned} \tag{5}$$

The set of operators $\tilde{\Delta}$ considered will now be defined in terms of a set of QCs $\tilde{\mathcal{M}}$, which is constructed from $\mathcal{M}$ to characterize the original Δ and additional QCs to characterize the nonlinearities.

We now provide conditions amenable to analysis via semidefinite programming that are sufficient for Theorem 1.

Theorem 2: Let $\tilde{\mathcal{M}}$ be a set of QCs, $P \in \mathbb{R}^{n\times n}$ be such that $P \succ 0$, and $\rho > 0$. Consider a quadratic supply rate $s(d,e)$. If there exist $s_\rho \ge 0, s_d \ge 0$, and $M_{\text{rfi}}, M_{\text{perf}} \in \tilde{\mathcal{M}}$ such that (6) and (7) hold, then (1) is robustly dissipative with respect to s on $(\Omega_{V,\rho}, B_{\overline{d}}^{n_d}, \mathbf{\Delta}(\mathcal{M}))$.

In the following, let X, Z, D, S be such that $\xi = (x, \tilde{w}, d)$, $z = (\tilde{v}, \tilde{w})$, $x_{k+1} = X\xi_k$, $z_k = Z\xi_k$, $\begin{bmatrix} d_k^\top & e_k^\top \end{bmatrix}^\top = D\xi_k$ and $s(d_k, e_k) = \xi_k^\top D^\top S D \xi_k$.

$$-X^\top P X + \begin{bmatrix} s_\rho P & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & s_d I \end{bmatrix} - Z^\top M_{\text{rfi}} Z \succeq 0 \tag{6a}$$

$$(1 - s_\rho)\rho - s_d \overline{d}^2 \ge 0 \tag{6b}$$

$$-X^\top P X + \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} - Z^\top M_{\text{perf}} Z + D^\top S D \succeq 0 \tag{7}$$

Proof: Define $V(x) = x^\top P x$. Label the matrix in (6a) as Y and the scalar in (6b) as s. By condition (6), $Y \succeq 0$ and $s \ge 0$, which is equivalent to $\xi^\top Y \xi + s \ge 0$ for all ξ. Expanding, this is equivalent to $-V(F(x,w,d)) + \rho - s_\rho(-V(x) + \rho) - s_d(-d^\top d + \overline{d}^2) - z^\top M_{\text{rfi}} z \ge 0$ for all x, w, d. This is a sufficient condition for (2). Similarly, left- and right-multiplying the matrix in (7) by $\xi^\top$ and ξ shows it is equivalent to $-V(F(x,w,d)) + V(x) + s(d,e) - z^\top M_{\text{perf}} z \ge 0$ for all x, w, d. This is a sufficient condition for (3). Therefore, by Theorem 1, (1) is robustly dissipative on $(\Omega_{V,\rho}, B_{\overline{d}}, \mathbf{\Delta}(\mathcal{M}))$. ∎

Remark 2: If the QCs in $\tilde{\mathcal{M}}$ hold locally, then the following provides a sufficient condition for (4), where $\tilde{V}_i$ is such that $\tilde{L}_i \tilde{v} = \tilde{V}_i \xi$.

$$-\tilde{V}_i^\top \tilde{V}_i + \begin{bmatrix} s_{\rho i} P & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & s_{di} I \end{bmatrix} - Z^\top M_{\Delta i} Z \succeq 0$$

$$\overline{\tilde{v}}_i^2 - s_{\rho i}\rho - s_{di}\overline{d}^2 \ge 0$$

Given a particular P, $\overline{d}$, and $\tilde{\mathcal{M}}$, the problem of finding the largest sub-level set of $V(x) = x^\top P x$ such that (1) is dissipative on $(\Omega_{V,\rho}, B_{\overline{d}}, \mathbf{\Delta}(\tilde{\mathcal{M}}))$ is formulated as the following optimization problem:

$$\begin{aligned} \max_{\rho,\, s_\rho,\, s_d,\, M_{\text{rfi}},\, M_{\text{perf}}} \quad & \rho && \text{(8a)} \\ \text{s.t.} \quad & s_\rho, s_d \ge 0 && \text{(8b)} \\ & M_{\text{rfi}}, M_{\text{perf}} \in \tilde{\mathcal{M}} && \text{(8c)} \\ & (6), (7) && \text{(8d)} \end{aligned}$$

This is an SDP for fixed ρ. To address bilinearity in ρ and decision variable s_ρ, we bisect on ρ. If quadratic constraints $\tilde{\mathcal{M}}$ are local, variables and constraints are used as per Remark 2. It is often useful to parameterize the local QC in terms of the region over which it is valid, and search over this set of local QCs. We do this by sampling different local QCs and, for each, determining the maximum ρ by bisection, and taking the maximum of all ρ found.

IV. Methodology

In this section, we detail our algorithm for training controllers and verifying robust dissipativity of systems of the form (1) using α,β-CROWN. We verify the condition $(\varphi_{\text{rfi}}(x,w,d) > 0) \wedge (\varphi_{\text{perf}}(x,w,d) > 0)$, which we label $\varphi(x,w,d)$ (see Section III-A for construction of the robust forward invariance performance requirement conditions). Algorithm 1 summarizes the training and verification pipeline. It has two stages: the first trains learnable parameters—the controller, storage function, and others—so that φ is satisfied and the certified volume is maximized, and the second uses α,β-CROWN to verify the result of the training.

Algorithm 1 Adversarial Training and Verification

Require: System (1), supply rate s, quadratic constraints $\mathcal{M}$, initial controller π_0, storage function initialization $P \succ 0$, initial bounding box for state $\mathcal{B}_0$, disturbance box $\mathcal{D}$

Training:
Initialize $P_\phi \leftarrow P, V_\phi^{(0)} \leftarrow x^\top P_\phi x, \pi_\theta^{(0)} \leftarrow \pi_0$
Run PGD + α,β-CROWN bisection on $V_\phi^{(0)} \to \rho_0$
Sample anchors $\mathcal{A}$ in $V_\phi^{(0)}(x) \in [\alpha_{\text{exp},1}\rho_0,\ \alpha_{\text{exp},2}\rho_0]$
for epoch $= 1, \ldots, N_{\text{epochs}}$ **do**
  $\tilde{\xi}_i = (x_i, \tilde{w}_i, d_i) \sim \text{Uniform}(\mathcal{B}_j \times \tilde{\mathcal{W}} \times \mathcal{D})$
  $\tilde{\xi}_i^{\text{adv}} \leftarrow$ PGD to estimate $\arg\min_{\tilde{\xi}} \varphi(\tilde{\xi})$
  $\rho \leftarrow \min_{x \in \partial\mathcal{B}_j} V_\phi(x)$
  $\mathcal{L}_{\text{total}} \leftarrow \lambda_{\text{s}}\mathcal{L}_{\text{sample}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{anc}}\mathcal{L}_{\text{anchor}}$
  Update θ, ϕ, other learnable parameters using $\mathcal{L}_{\text{total}}$
  if $\min_i \varphi(x_i, w_i, d_i) \geq 0$ for N_{clean} consecutive epochs **then**
    Simulate trajectories from probe box $\supset \mathcal{B}_j$; keep converging ones
    $\mathcal{B}_{j+1} \leftarrow$ bounding box of reachable envelopes (capped by $\eta_{\max}$)
    break (end training) if $\mathcal{B}_{j+1} = \mathcal{B}_j$
  end if
end for
Verification: Bisect over ρ with α,β-CROWN (Sec. IV-C)
return Certified region Ω_{V_ϕ,ρ^*}, controller π_θ

A. Controller and Certificate

1) Storage Function: We use a neural network storage function that combines a quadratic term and a nonlinear modulation. Let $P_\phi \coloneqq \epsilon I + R^\top R$ where $R \in \mathbb{R}^{n\times n}$ is a trainable matrix and $\epsilon > 0$ is a fixed constant, and let $\psi_\phi \colon \mathbb{R}^n \to \mathbb{R}$ be a neural network. The storage function $V_\phi \colon \mathbb{R}^n \to \mathbb{R}$ is defined as

$$V_\phi(x) = \underbrace{x^\top P_\phi\, x}_{V_{\text{quad}}(x)} \cdot \underbrace{\big(1 + \alpha_{\text{nn}} \tanh\big(\psi_\phi(x) - \psi_\phi(0)\big)\big)}_{m_\phi(x)}, \quad (9)$$

where x is measured relative to the equilibrium x^* and $\alpha_{nn} \in (0,1)$ is a fixed scaling hyperparameter. The function V_ϕ is nonnegative and satisfies $V_\phi(0) = 0$ by construction. Further, V_ϕ is positive definite and radially unbounded because $m_\phi(x) \geq 1 - \alpha_{nn}$ for all x, ensuring $\Omega_{V_\phi,\rho}$ is compact for all ρ.

We initialize this storage function with a quadratic storage function $x^\top P x$ obtained by the LMI method. We find that normalizing P by its Frobenius norm improves numerical stability during training. At initialization, the parameters of the last layer of ψ_ϕ are set to 0 so that $V_\phi(x) = x^\top P_\phi x$. The Cholesky decomposition yields the R matrices to initialize P_ϕ. This initialization provides a warm start that accelerates convergence. During training, both R and the network ψ_ϕ are updated.

2) Controller: The proposed method is compatible with standard neural network and controller architectures such as multi-layer perceptrons, linear state-feedback, and LTI controllers. In this paper, we use a recurrent implicit neural network (RINN) [5] due to its compatibility with LMI methods for robustness analysis, which enables us to use LMI methods to construct initializing controllers. Note that Algorithm 1 can also be used to verify systems with a fixed controller by fixing its parameters.

A RINN controller is a generalization of an LTI controller constructed by interconnecting an LTI system with an implicit neural network [20]. It is of the form

$$\pi_\theta \begin{cases} \dot{x}_K(t) = A_K x_K(t) + B_{Kw} w_K(t) + B_{Ky} y(t) \\ v_K(t) = C_{Kv} x_K(t) + D_{Kvw} w_K(t) + D_{Kvy} y(t) \\ u(t) = C_{Ku} x_K(t) + D_{Kuw} w_K(t) + D_{Kuy} y(t) \\ w_K(t) = \sigma(v_K(t)) \end{cases}$$

where $x_K \in \mathbb{R}^{n_K}$ is the controller state, $w_K \in \mathbb{R}^{n_{Kw}}$ is the output of the implicit neural network, $v_K \in \mathbb{R}^{n_{Kw}}$, y is the output of the plant, u is the control input of the plant, and σ is the activation function, which we take to be ReLU. To simplify the evaluation of the controller, we restrict D_{Kvw} to be strictly upper triangular.

3) QC Multipliers and Supply-Rate Scale: The conditions of Theorem 1 allow for any QC defined by M in a set $\mathcal{M}$. Given a set $\mathcal{M}$ parameterized through a differentiable transformation of parameters Λ, we add these parameters to the set of parameters that are tuned through gradient descent in training. For example, for a set $\mathcal{M} = \{\lambda M_0 | \lambda \geq 0\}$, we introduce and train the parameter $\lambda \geq 0$ (parameterized as $\lambda = \tilde{\lambda}^2$ where $\tilde{\lambda} \in \mathbb{R}$) with λ initialized to 1.0.

Further, the system (1) is dissipative with respect to a supply rate s if and only if it is dissipative with respect to $\alpha_s s$ where $\alpha_s > 0$. To automatically tune this supply rate scale hyperparameter, we introduce it as a learnable parameter (parameterized as $\alpha_s = \exp(\beta_s)$ with $\beta_s \in \mathbb{R}$). In the dissipation inequality condition φ_{perf}, we use the supply rate $\alpha_s s$. To initialize α_s, we use $1/\|P\|_F$, where P is the initial storage function certificate found for certifying the dissipation inequality and $P/\|P\|_F$ is the normalized matrix used to initialize P_ϕ.

B. Training Procedure

We train the controller parameters, the storage function parameters, and the multipliers to satisfy $\varphi(x,w,d) > 0$

through adversarial training.

1) Sub-level Set and Loss: Given a box $\mathcal{B}$ for the state, we estimate $\rho = \min_{x \in \partial\mathcal{B}} V_\phi(x)$, the largest sub-level set of V_ϕ that is contained in $\mathcal{B}$, through projected gradient descent (PGD) [17] on each face of $\mathcal{B}$. PGD is an iterative first-order method that repeats the cycle of taking a gradient step and projecting back onto a constraint set; it was introduced for evaluating adversarial robustness of neural networks [16] and provides an efficient inner maximizer (or minimizer) over bounded domains.

We then attempt to train the learnable parameters to verify the condition φ on $\Omega_{V_\phi,\rho}$. In each training epoch, a batch of state–disturbance pairs $(x_i, \tilde{w}_i, d_i)$ is drawn uniformly at random from the bounding box $\mathcal{B} \times \tilde{\mathcal{W}} \times \mathcal{D}$ and w is computed from $(x_i, \tilde{w}_i, d_i)$ (see Section III-A). The training loss penalizes violations of the verification condition at these samples:

$$\mathcal{L}_{\text{sample}} = \frac{1}{N_{\text{sample}}} \sum_{i=1}^{N_{\text{sample}}} \text{ReLU}\big(-\varphi(x_i, w_i, d_i)\big). \tag{10}$$

2) Adversarial Augmentation Loss: Uniform sampling alone may miss thin violation regions where $\varphi < 0$, since such regions can occupy a negligible fraction of the domain volume. To address this, we augment the training set with PGD adversarial examples: starting from random initializations in $\mathcal{B} \times \tilde{\mathcal{W}} \times \mathcal{D}$, PGD minimizes $\varphi(x, w, d)$ subject to $(x, \tilde{w}, d) \in \mathcal{B} \times \tilde{\mathcal{W}} \times \mathcal{D}$, concentrating samples where violations are most likely. Worst-case points are stored in a ranked replay buffer. We compute a second loss term using adversarial points sampled from this buffer.

$$\mathcal{L}_{\text{adv}} = \frac{1}{N_{\text{adv}}} \sum_{i=1}^{N_{\text{adv}}} \text{ReLU}\big(-\varphi(x_i^{\text{adv}}, w_i^{\text{adv}}, d_i^{\text{adv}})\big). \tag{11}$$

3) Growth Incentive via Anchor Sampling Loss: To encourage the sub-level set $\Omega_{V_\phi,\rho}$ to grow, we sample *anchor* states in a band around the initial verified region and penalize high storage-function values at those states. Specifically, let ρ_0 denote the initial PGD-verified sub-level set value and $V_\phi^{(0)}$ the initial storage function. We sample anchor states whose initial storage-function values lie in $[\alpha_{\text{exp},1}\,\rho,\ \ \alpha_{\text{exp},2}\,\rho]$, where $\alpha_{\text{exp},1} < 1$ penalizes for shrinking below the initial certified region and $\alpha_{\text{exp},2} > 1$ rewards expansion beyond it. The regularization term is

$$\mathcal{L}_{\text{anchor}} = \frac{1}{|\mathcal{A}|} \sum_{x \in \mathcal{A}} \text{ReLU}\left(\frac{V_\phi(x)}{\rho} - 1\right), \tag{12}$$

where $\mathcal{A}$ is the set of anchor states and ρ is the current sub-level set value. This term penalizes anchor states that lie outside the current sublevel set, encouraging the optimizer to reshape V_ϕ so that $\Omega_{V_\phi,\rho}$ expands to include them. Hyperparameters are listed in Table II.

The total loss is then a weighted sum of individual loss terms.

$$\mathcal{L}_{\text{total}} = \lambda_{\text{s}}\mathcal{L}_{\text{sample}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{anc}}\mathcal{L}_{\text{anchor}} \tag{13}$$

4) Simulation-Based Domain Expansion: We increase the size of $\mathcal{B}$, and therefore the size of the sub-level set we aim to verify, using trajectory simulations, following the approach of [4]. Expansion is triggered when PGD finds no violations for N_{clean} consecutive epochs, indicating that the adversarial sampler can no longer falsify the condition over the current region. At each expansion step, we sample initial conditions from a *probe box* that extends beyond the current domain $\mathcal{B}_j$, simulate the system for K_{sim} steps, and retain only those trajectories that converge to the origin (measured by the terminal state norm falling below a threshold δ_{conv}). The new domain $\mathcal{B}_{j+1}$ is the element-wise bounding box of the reachable envelopes of the converging trajectories, capped by a maximum growth factor $\eta_{\max}$. This procedure expands the domain where the current controller stabilizes the system, while avoiding regions where it does not. Training terminates when the domain no longer grows (no converging trajectories reach beyond the current box).

C. Formal Verification

After training, we fix the trained parameters and use α,β-CROWN to verify $\varphi_{\text{rfi}}(x, w, d) > 0$ and $\varphi_{\text{perf}}(x, w, d) > 0$ through bisection on the level set ρ to maximize the volume of the verified region. First, given the final box $\mathcal{B}$ on state from training, we estimate $\rho_{\max} = \min_{x \in \partial\mathcal{B}} V_\phi(x)$. Then, the bisection procedure is as follows.

1) Initialize the search bracket: first test $\rho_0 = \beta\rho_{\max}$ where $\beta < 1$ (we use $\beta \sim 0.95$). If α,β-CROWN verifies ρ_0: bracket $= [\rho_0, \mu\rho_{\max}]$. If not: bracket $= [\rho_{\max}/\mu, \rho_0]$ where $\mu > 1$ is a multiplier.
2) At each bisection step, PGD pre-screens the candidate ρ; if a counterexample is found, the candidate is rejected without calling α,β-CROWN.
3) α,β-CROWN verifies the sub-problem $\varphi_{\text{rfi}}(x, w, d) > 0$. If it fails, the upper bound is lowered, and we go to step (2).
4) α,β-CROWN verifies the sub-problem $\varphi_{\text{perf}}(x, w, d) > 0$. If it passes, the lower bound is raised; otherwise, the upper bound is lowered.
5) Repeat from step (2) until the bracket width is below a tolerance δ.

Remark 3 (Tight bounding boxes): For each candidate ρ, we compute a tight bounding box for $\Omega_{V_\phi,\rho}$ by sampling and finding the element-wise extrema of points satisfying $V(x) \le \rho$, expanded by a small margin. This replaces the full domain $\mathcal{B}$ in the specification and tightens the α,β-CROWN bounds.

As a baseline, we run the full bisection procedure on the initial quadratic storage function $V_\phi^{(0)}(x) = x^\top P_\phi\, x$ (with zero neural correction) with the initial controller. Comparing these pre-training certified regions to the post-training results isolates the improvement due to the nonlinear storage function and controller training.

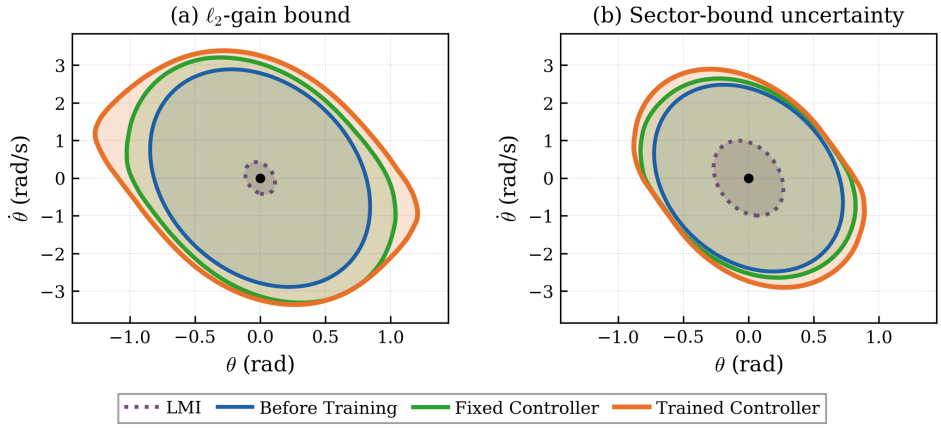

Fig. 1. Certified regions projected onto the plant state $(\theta, \dot{\theta})$ for (a) the ℓ_2-gain bound experiment and (b) the robust stability experiment under sector-bound uncertainty ($\alpha = 0.25$). Four methods are compared: LMI baseline (purple, dotted), before training (blue), training only the storage function with a fixed controller (green), and jointly training both controller and storage function (orange).

V. Numerical Experiments

We evaluate the proposed framework on a torque-actuated inverted pendulum[1]. In two experiments, we present two results using the method presented in Section IV: one where we train both the controller and dissipativity certificates to maximize the certified volume, and a second where we train only the dissipativity certificates and leave the controller fixed. Where applicable, we compare the results with the regions certified using the LMI techniques presented in Section III-B.

A. Certifying ℓ_2-Gain Bound

Consider the following model of an inverted pendulum with a disturbance entering into the control input.

$$\begin{aligned}\dot{x}_{P1}(t) &= x_{P2}(t)\\ \dot{x}_{P2}(t) &= -\frac{\mu}{m\ell^2}x_{P2}(t) + \frac{g}{\ell}\sin(x_{P1}(t))\\ &\quad + \frac{1}{m\ell^2}(\mathrm{sat}_{\overline{u}}(u(t)) + d(t))\\ y(t) &= x_P(t)\end{aligned}$$

The control input u is saturated in the interval $[-\overline{u}, \overline{u}]$ and the disturbance is bounded in absolute value by $0.1\overline{u}$. All parameters are given in Table II.

We certify dissipativity of the closed-loop of the plant and controller from disturbance d to performance output $e = x_P$ with respect to the ℓ_2 gain supply rate $s(d, e) = \gamma^2\|d\|^2 - \|e\|^2$. We initialize the RINN controller with one synthesized using LMI methods similar to [5]. We use no saturation on the controller, no bounds on the disturbance, and a quadratic constraint on sin that is valid for $x_{P1} \in [-\pi, \pi]$. This comes with an associated quadratic storage function. The plant and controller are discretized individually using Euler integration, and Theorem 1 is applied to the closed-loop system, which has state $x = (x_P, x_K)$.

We compute four certified regions $\Omega_{V,\rho}$:

1) Using the initial controller and its associated storage function using the LMI method in Section III-B.
2) Using the initial controller and its associated storage function using α,β-CROWN as in Section IV-C.
3) Using the initial controller and training the certificates with Algorithm 1.
4) Training both the controller and the certificates with Algorithm 1.

For the LMI method, we apply local sector constraints to sin and to the saturation function $\mathrm{sat}(v) = \min\{1, \max\{-1, v\}\}$ (the function $\mathrm{sat}_{\overline{u}}(u) = \overline{u}\ \mathrm{sat}(u/\overline{u})$). We grid search over local sector bounds on sin valid for $|x_{P1}| \le \overline{x}_{P1}$ with $\overline{x}_{P1} \in [0, \pi]$ and similarly grid search over local sector bounds on sat valid for $|v| \le \overline{v}$ with $\overline{v} \in [1.0, 5.0]$ using a resolution of 0.1.

Table I reports the volumes of the certified sub-level sets $\Omega_{V,\rho}$ projected onto the plant state x_P, and the computation times. Figure 1 visualizes the projections of $\Omega_{V_\phi,\rho}$ onto the two-dimensional plane of plant states, given as:

$$\mathcal{P}(\Omega_{V,\rho}) = \{x_P | \min_{x_K} V(x_P, x_K) \le \rho\} \tag{14}$$

and is estimated by minimizing V over the controller states at each point of a fine grid. All experiments are run on a computer with a Nvidia RTX 5090 and an AMD 9950 X3D.

The LMI baseline yields the smallest certified volume. Even before training, verifying the same closed-loop system using the same storage function with α,β-CROWN gives a $51\times$ larger region, illustrating the conservatism of using QCs to characterize the nonlinearities. Training the storage function increases the verified volume to $67.5\times$ the LMI baseline and $1.32\times$ the region verified with α,β-CROWN without training. Joint training of the controller and the storage function results in the largest verified volume at $78.1\times$ the LMI baseline, $1.52\times$ the region verified with α,β-CROWN without training, and $1.16\times$ the region obtained by training the storage function alone. This demonstrates that substantial enlargement of the certified region is achievable

[1] https://github.com/neelayjunnarkar/local-robust-dissipativity

TABLE I

VOLUMES AND COMPUTATION TIMES

Method	Volumes $\mathcal{P}(\Omega_{V,\rho})$ (rad·rad/s)			Times (min.)		
	Abs.	vs. LMI	vs. Before train	Train	Verif.	Total
ℓ_2-Gain Bound						
1) LMI	0.16	1.0×	—	N/A	0.03	0.03
2) Before Training	8.2	51.2×	1.0×	N/A	9	9
3) Fixed Controller	10.8	67.5×	1.32×	3	33	36
4) Trained Controller	12.5	78.1×	1.52×	6	26	32
Robust Stability						
1) LMI	0.80	1.0×	—	N/A	0.03	0.03
2) Before Training	6.0	7.6×	1.0×	N/A	21	21
3) Fixed Controller	7.1	8.9×	1.17×	7	99	106
4) Trained Controller	8.1	10.2×	1.35×	19	96	115

even without modifying the controller, and that co-training provides additional benefit.

A note on the gap between the volume that is not falsified by PGD attacks and the volume that is certified with α,β-CROWN: With the initial controller and storage function, PGD and α,β-CROWN areas are approximately identical (8.2 vs. 8.2). After training, a modest gap appears (e.g., 13.1 vs. 12.5 for the trainable case), reflecting the inevitable relaxation in α,β-CROWN's bound propagation through the more expressive trained storage function.

B. Robust Stability under Model Uncertainty

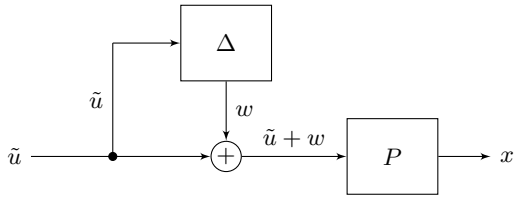

Fig. 2. Uncertainty structure: Plant input is sum of control $\tilde{u}$ and $w = \Delta(\tilde{u})$, where Δ is uncertain and satisfies $\|w\| \le \alpha\|\tilde{u}\|$ pointwise in time.

We now apply the framework to verify robust stability (as in, use supply rate $s(d,e) = 0$) for the same pendulum subject to non-parametric uncertainty Δ on the input channel (Figure 2). The uncertain dynamics take the form

$$\begin{aligned}
\dot{x}_{P1}(t) &= x_{P2}(t) \\
\dot{x}_{P2}(t) &= -\frac{\mu}{m\ell^2}x_{P2}(t) + \frac{g}{\ell}\sin(x_{P1}(t)) \\
&\quad + \frac{1}{m\ell^2}\big(\tilde{u}(t) + \Delta(\tilde{u}(t))\big) \\
\tilde{u}(t) &= \mathrm{sat}_{\overline{u}}(u(t)) \\
y(t) &= x_P(t)
\end{aligned}$$

where Δ is characterized by the sector-bound QC

$$\begin{bmatrix} v \\ w \end{bmatrix}^{\top} \underbrace{\begin{bmatrix} \alpha^2 & 0 \\ 0 & -1 \end{bmatrix}}_{M_0} \begin{bmatrix} v \\ w \end{bmatrix} \ge 0, \tag{15}$$

so that $\|w(t)\| \le \alpha\|u(t)\|$ for all t. We use the same initializing controller and storage function.

Note that verifying stability of this uncertain system implies that, among other uncertainties, the controller stabilizes the plants gP for all $g \in [1-\alpha, 1+\alpha]$ where P is the nominal plant with $\Delta = 0$. Therefore, this uncertainty is sufficient for analyzing gain margin conditions.

To construct a bounded parameterization suitable for α,β-CROWN, we write $w_k = \alpha\,\tilde{w}_k\,v_k, \quad \tilde{w}_k \in [-1,1]$, which satisfies (15) by construction. The verification conditions φ_{rfi} and φ_{perf} are then defined over the bounded domain $\mathcal{X}\times[-1,1]$. Results are reported in Table I. The LMI baseline certifies $0.80\,\text{rad}\cdot\text{rad/s}$ with the uncertainty characterized by a sector-bound QC. Before training, α,β-CROWN certifies a projected area of $6.0\,\text{rad}\cdot\text{rad/s}$ (7.6× the LMI baseline), smaller than the ℓ_2-gain case due to the cost of certifying stability for all plants in the uncertainty set. Training the storage function with a fixed controller increases the α,β-CROWN-verified area to 7.1 (+17% over the initial certificate). Joint training of the controller and storage function yields $8.1\,\text{rad}\cdot\text{rad/s}$ (+35% over the initial, 1.14× the fixed-controller result). Figure 1 visualizes the certified regions.

TABLE II

PHYSICAL AND ALGORITHM PARAMETERS.

Parameter	Symbol	ℓ_2-Gain Bound	Robust stability
Plant (shared)			
Mass	m	0.15 kg	
Length	ℓ	0.5 m	
Damping	μ	0.1 Nm s/rad	
Gravity	g	9.81 m/s^2	
Time step	Δt	0.01 s	
Torque limit	$\overline{u}$	0.75 Nm ($\approx 1.02\,mgl$)	
Supply rate / uncertainty			
ℓ_2-gain bound	γ	100	—
Disturbance bound	$\overline{d}$	0.075 Nm	—
Uncertainty bound	α	—	0.25
Auxiliary variable	$\tilde{w}$	—	$\in [-1, 1]$
Supply rate	$s(d,e)$	$\gamma^2\|d\|^2 - \|e\|^2$	0
Controller (shared)			
RINN internal states	n_K	2	
RINN implicit nodes	n_{Kw}	8	
Storage function			
Storage NN hidden	—	[128, 128]	[32, 32]
Hidden activation	—	LeakyReLU	
Neural scale	α_{nn}	0.5	0.25
Domain (initial)	$\mathcal{B}_0$	$[-3,3]\times[-9,9]\times[-4,4]^2$	
Anchor sampling			
Anchor count	$\|\mathcal{A}\|$	1,024	
Inner anchor factor	$\alpha_{\text{exp},1}$	0.75	
Outer anchor factor	$\alpha_{\text{exp},2}$	1.3 / 1.2 (train/fix)	1.3 / 1.1 (train/fix)
Anchor weight	λ_a	0.1	
Loss weights	$\lambda_{\text{s}}, \lambda_{\text{adv}}$	1	

VI. CONCLUSION

This paper presented a framework for training neural network controllers and corresponding certificates to ensure robust dissipativity of the closed-loop system. A key advantage of this approach is its ability to leverage the complementary strengths of α,β-CROWN for direct analysis of known

nonlinear dynamics and QCs for non-parametric model uncertainty. As demonstrated in numerical experiments, this method significantly reduces conservativeness compared to LMI-based techniques, yielding substantially larger certified volumes, although at the expense of higher computation time. Future work includes extending the framework from QCs to integral quadratic constraints.

REFERENCES

[1] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 1980–1987, Apr. 2022.

[2] R. Wang and I. R. Manchester, "Youla-REN: Learning nonlinear feedback policies with robust stability guarantees," in *2022 American Control Conference (ACC)*, Jun. 2022, pp. 2116–2123.

[3] 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 *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235, PMLR, 21–27 Jul 2024, pp. 56 033–56 046.

[4] H. Li, X. Zhong, B. Hu, and H. Zhang, "Two-stage learning of stabilizing neural controllers via Zubov sampling and iterative domain expansion," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[5] N. Junnarkar, M. Arcak, and P. Seiler, "Stability margins of neural network controllers," in *2025 American Control Conference (ACC)*, IEEE, 2025, pp. 1355–1360.

[6] J. C. Willems, "Dissipative dynamical systems part I: General theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pp. 321–351, Jan. 1, 1972.

[7] M. Arcak, C. Meissen, and A. Packard, *Networks of Dissipative Systems* (SpringerBriefs in Electrical and Computer Engineering). Cham: Springer International Publishing, 2016.

[8] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 42, no. 6, pp. 819–830, Jun. 1997.

[9] J. Veenman, C. W. Scherer, and H. Köroğlu, "Robust stability and performance analysis based on integral quadratic constraints," *European Journal of Control*, vol. 31, pp. 1–32, Sep. 1, 2016.

[10] P. Seiler, "Stability analysis with dissipation inequalities and integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1704–1709, Jun. 2015, Conference Name: IEEE Transactions on Automatic Control.

[11] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 1–15, Jan. 2022.

[12] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer, "Training robust neural networks using Lipschitz bounds," *IEEE Control Systems Letters*, vol. 6, pp. 121–126, 2022.

[13] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo, "Formal synthesis of Lyapunov neural networks," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 773–778, 2021.

[14] R. Zhou, T. Quartz, H. De Sterck, and J. Liu, "Neural Lyapunov control of unknown nonlinear systems with stability guarantees," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, Red Hook, NY, USA: Curran Associates Inc., Apr. 3, 2024, pp. 29 113–29 125.

[15] S. Wang et al., "Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21, Red Hook, NY, USA: Curran Associates Inc., 2021.

[16] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[17] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[18] K. Xu et al., "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1129–1141, 2020.

[19] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 57–95, Jan. 2016.

[20] L. El Ghaoui, F. Gu, B. Travacca, A. Askari, and A. Tsai, "Implicit deep learning," *SIAM Journal on Mathematics of Data Science*, vol. 3, no. 3, pp. 930–958, Jan. 2021.