

Learning Certified Neural Network Controllers Using Contraction and Interval Analysis

Akash Harapanahalli¹, Samuel Coogan¹, and Alexander Davydov²

Abstract—We present a novel framework that jointly trains a neural network controller and a neural Riemannian metric with rigorous closed-loop contraction guarantees using formal bound propagation. Directly bounding the symmetric Riemannian contraction linear matrix inequality causes unnecessary overconservativeness due to poor dependency management. Instead, we analyze an asymmetric matrix function G , where 2^n GPU-parallelized corner checks of its interval hull verify that an entire interval subset X is a contraction region in a single shot. This eliminates the sample complexity problems encountered with previous Lipschitz-based guarantees. Additionally, for control-affine systems under a Killing field assumption, our method produces an explicit tracking controller capable of exponentially stabilizing any dynamically feasible trajectory using just two forward inferences of the learned policy. Using JAX and `immrnx` for linear bound propagation, we apply this approach to a full 10-state quadrotor model. In <10 minutes of post-JIT training, we simultaneously learn a control policy π , a neural contraction metric Θ , and a verified 10-dimensional contraction region X .

Index Terms—Neural networks, contraction theory, formal methods

I. INTRODUCTION

LEARNING-based methods have demonstrated potential for controlling complex nonlinear systems, owing to their ability to represent and learn highly expressive policies that can be difficult to obtain through classical control design. However, deploying learned controllers in safety-critical applications demands rigorous guarantees on closed-loop behavior that such policies do not inherently provide.

This gap between expressiveness and certifiability has motivated a growing body of work on formally verifying closed-loop behavior of systems with neural network controllers. For instance, one can certify neural Lyapunov functions for stabilization or neural control barrier functions for safety, using satisfiability modulo theories (SMT) or branch-and-bound optimization to rigorously conclude the lack of counterexamples [1]. These exact verification approaches, however, can be difficult to scale to large state spaces and deeper networks.

*This work was supported in part by the National Science Foundation under awards #2219755 and #2440387, and by the Air Force Office of Scientific Research under Grant FA9550-23-1-0303.

¹Akash Harapanahalli and Samuel Coogan are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA. {aharapan,sam.coogan}@gatech.edu

²Alexander Davydov is with the Department of Mechanical Engineering and Ken Kennedy Institute, Rice University, Houston, TX, 77005, USA. davydov@rice.edu

A scalable alternative uses *bound propagation* [2] to verify guaranteed overapproximations of the learned components.

Beyond stabilization and invariance, contraction theory has provided a suite of tools suitable to tackle other challenges like online tracking [3], and recent work has developed neural contraction metrics [4] to help scale these methods. However, formally verifying these contraction metrics remains a significant challenge, relying on Lipschitz-based sampling arguments or over-conservative bounding techniques [5], [6]. In this work, we use off-the-shelf neural network bound propagators, careful dependency management through the contraction condition, and an efficient exact characterization of the ℓ_2 -logarithmic norm of an interval matrix to jointly train neural network controllers and contraction metrics with formal guarantees.

A. Problem Formulation and Contributions

Consider a nonlinear control system of the form

$$\dot{x} = f_{ol}(t, x, u), \quad (1)$$

where $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^m$ is a control input, and $f_{ol} : [0, \infty) \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a C^1 smooth mapping. Our primary goal is to jointly synthesize an explicit feedback policy $\pi(x)$ and a Riemannian metric $M(x)$ such that the closed-loop system

$$\dot{x} = f_{\pi}(t, x) := f_{ol}(t, x, \pi(x)) \quad (2)$$

is contracting. We also consider the special case of control-affine systems, $\dot{x} = f_d(x) + g(x)u$, where the goal extends to tracking any dynamically feasible trajectory.

To address these problems, we develop a novel framework using bound propagation to jointly train a neural contraction metric and a neural network controller for nonlinear systems with formal guarantees of closed-loop contractivity. Our specific contributions are outlined:

- **Asymmetric Contraction Condition.** We propose an asymmetric Riemannian contraction condition, equivalent to the standard linear matrix inequality (LMI) provided in [7], which bounds the ℓ_2 -logarithmic norm of a matrix directly involving the factored parameterization of the metric. While mathematically equivalent on point-wise evaluations, our asymmetric condition can dramatically reduce overestimation error when using bound propagation tools due to better dependency management.
- **Certification and Training via Bound Propagation.** Using off-the-shelf bound propagators with automatic differentiation, we algorithmically obtain an interval hull of this

asymmetric matrix. Applying a result from [8], we show how the maximum logarithmic norm within this interval hull can be checked with 2^n GPU parallelized corner checks. We design a loss function in Algorithm 1 which, if nonpositive, certifies the contraction condition over an entire region X without exhaustive sampling.

- **Explicit Tracking Control.** In the special case where the system is control affine and the metric satisfies a Killing field condition (matching the strong control contraction metric (CCM) condition from [3, C2]), we propose a simple, explicit feedback controller $u(t, x) = \pi(x) - \pi(x'(t)) + u'(t)$, capable of tracking any open-loop dynamically feasible pair $(x'(t), u'(t))$ using only two forward inferences of the learned policy π . Notably, our method eliminates the need to search for a geodesic of the metric during online execution, and does not require any state augmentation into error coordinates.

One of the crucial distinctions between our approach and previous verification frameworks such as [9], which certifies neural contraction metrics after training using SMT solvers, is that we embed a differentiable certificate from the efficient bound propagator CROWN [10] directly in the training loss. This allows us to *learn* controllers and neural contraction metrics that are certified by construction, rather than *post hoc* verification, and then optimize directly against the overestimation error to help mitigate wrapping effects.

Finally, we illustrate through a 10-state quadrotor benchmark how the proposed method can certify and train explicit contracting tracking controllers, where sampling-based certification would suffer from the curse of dimensionality.

B. Background, Notation, and Related Work

Riemannian Metrics and Contraction Theory: Contraction theory provides a geometric framework for translating infinitesimal linear properties into incremental exponential stability bounds between trajectories of the system. There is a rich literature in contraction analysis, including [7] for Riemannian spaces, [11] for a differential Lyapunov function oriented framework, [12] for distances defined by possibly non-Euclidean norms, and [13], [3] for controller design. In this work, we focus on Riemannian geometry in $\mathbb{R}^n$.

Let $\mu_2(\cdot)$ denote the induced ℓ_2 logarithmic norm $\mu_2(A) = \lambda_{\max}(\frac{A+A^T}{2})$. For a matrix function $A : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$, let $\partial_v A(x) = \sum_{i=1}^n \frac{\partial A}{\partial x_i}(x) v_i$ denote the directional derivative. Let $X \subseteq \mathbb{R}^n$ be a set whose interior $\text{int } X$ is nonempty and connected. A *Riemannian metric* M on X is a smooth, symmetric, positive-definite matrix function that is *uniformly bounded*, meaning there exist $a, b > 0$ such that $aI \preceq M(x) \preceq bI$ for every $x \in X$.

M defines a length for piecewise-smooth paths $\gamma : [0, 1] \rightarrow \text{int } X$ by integrating the Riemannian norm of the velocity: $\ell(\gamma) = \int_0^1 \sqrt{\gamma'(s)^T M(\gamma(s)) \gamma'(s)} ds$. This equips $\text{int } X$ with a distance $d(x, y)$, defined as the infimum of the lengths of all such paths in $\text{int } X$ connecting x and y (we do not assume $\text{int } X$ is geodesically complete, so this infimum may not be achieved by a geodesic within the space).

f_{o1}	Open-loop system (1)
f_π	Closed-loop system (2) under policy $u = \pi(x)$
f_d	Drift term for the control-affine system from (5)
g	Input vector fields for the control-affine system from (5)
f_{c1}	Closed-loop system (8) under policy $u = \pi(x) - \pi(x') + u'$
X	Region of the state space $\mathbb{R}^n$ with nonempty interior $\text{int } X$
M	Riemannian metric over X
Θ	Upper triangular factorization satisfying $M(x) = \Theta(x)^T \Theta(x)$
d	Metric geodesic distance associated to M
π	Feedback policy
S	Left-hand side of the contraction LMI (3)
G	Asymmetric contraction matrix from Definition 2

TABLE 1: Symbols and Notation

Definition 1 (Contraction Region [7, Def. 2]). A region of the state space X is called a *contraction region* of the vector field f_π from (2) with respect to uniformly bounded metric $M(x)$ if there exists $c > 0$ such that the linear matrix inequality $S(t, x; c) \preceq 0$ holds uniformly over $t \in [0, \infty)$, $x \in X$, where

$$S(t, x; c) := M(x) \frac{\partial f_\pi}{\partial x}(t, x) + \frac{\partial f_\pi}{\partial x}(t, x)^T M(x) + \partial_{f_\pi(t, x)} M(x) + 2cM(x). \quad (3)$$

We note that Definition 1 does not require X to be forward invariant. Definition 1 provides an important infinitesimal characterization of the following property: if two trajectories x, x' of f_π (2), with initial conditions x_0, x'_0 respectively at time t_0 , are contained within $\text{int } X$ and have a minimizing geodesic connecting them fully contained within $\text{int } X$, then

$$d(x(t), x'(t)) \leq e^{-c(t-t_0)} d(x_0, x'_0).$$

This bound is useful in a number of contexts—for instance, for autonomous systems, if a contraction region is forward invariant, geodesically convex and complete, the flow map ϕ_τ for fixed $\tau > 0$ is a contraction mapping, and Banach’s fixed point theorem concludes that the system exponentially converges to a unique equilibrium.

Interval Analysis and Bound Propagation: Let $\mathbb{IR}$ denote the set of closed intervals of $\mathbb{R}$, of the form $[a, \bar{a}] := \{a \in \mathbb{R} : a \leq a \leq \bar{a}\}$ for $a, \bar{a} \in \mathbb{R}$. An interval matrix $[A] = [\underline{A}, \bar{A}] \in \mathbb{IR}^{m \times n}$ is a matrix where $[A]_{ij} = [\underline{A}_{ij}, \bar{A}_{ij}] \in \mathbb{IR}$. $[A]$ is a subset of $\mathbb{R}^{m \times n}$ in the entrywise sense, so $A \in [A]$ if $A_{ij} \in [A]_{ij}$ for every $i = 1, \dots, m$ and $j = 1, \dots, n$. For the interval matrix $[A]$, we notate its center $A^c \in \mathbb{R}^{m \times n}$ as $A^c = (\underline{A} + \bar{A})/2$ and its radius $A^\Delta \in \mathbb{R}^{m \times n}$ as $A^\Delta = (\bar{A} - \underline{A})/2$. For a vector $v \in \mathbb{R}^n$, let $\text{diag}(v) \in \mathbb{R}^{n \times n}$ be the matrix $\text{diag}(v)_{ii} = v_i$ and $\text{diag}(v)_{ij} = 0$ for $i \neq j$.

Given a matrix function $A : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times n}$, an *interval hull* of A over an input set $X \subseteq \mathbb{R}^n$ is an interval matrix $[A] \in \mathbb{IR}^{m \times n}$ such that $A(x) \in [A]$ uniformly for every $x \in X$. Obtaining the smallest interval hull of A is generally an intractable nonconvex optimization problem. Instead, a general way to obtain larger valid interval hulls is through *bound propagation*, where (i) a computation tree for A is built through an abstract evaluation, and (ii) the input bound X is successively propagated through the computation tree. These existing approaches have varying tradeoffs of accuracy and runtime: interval bound propagation [14], [15], which efficiently propagates lower and upper scalar bounds; linear bound propagation [10], which propagates linear lower and upper bounding functions; set-based methods [16] including

zonotopes and star sets [17], which propagate set overapproximations forward through the network; Taylor models [18] and other polynomial approximators, which propagate tighter but more computationally expensive representations.

A fundamental issue often encountered by these approaches is the *dependency problem*. When a variable appears multiple times in a computation graph, these methods treat each instance as an independent variable, failing to capture algebraic interactions between terms. We demonstrate this phenomenon in Example 1 in the next Section, and show how our asymmetric formulation mitigates these issues.

II. RIEMANNIAN CONTRACTION VIA BOUND PROPAGATION

In prior contraction synthesis works [4], the LMI (3) is incorporated into the training by sampling points x_n across the desired contraction region X and backpropagating against $\lambda_{\max}(S(t, x_n; c))$. In this section, we use bound propagation to directly verify the contraction condition over the region X without sampling, which we use to design an explicit tracking control policy for control-affine systems.

A. The Asymmetric Contraction Matrix

We first propose an asymmetric matrix that reduces the overestimation error from bound propagation.

Definition 2 (Asymmetric Contraction Matrix). Consider the vector field f_π from (2), a Riemannian metric $M(x) = \Theta(x)^T \Theta(x) + aI$ for $a \geq 0$, and a fixed $c > 0$. Define the *asymmetric contraction matrix* as the following:

$$G(t, x; a, c) := a \left(\frac{\partial f_\pi}{\partial x}(t, x) + cI \right) + \Theta(x)^T \left[\frac{\partial f_\pi}{\partial x}(t, x) \Theta(x) + \Theta(x) \left(\frac{\partial f_\pi}{\partial x}(t, x) + cI \right) \right] \quad (4)$$

The following lemma demonstrates how an ℓ_2 -logarithmic norm bound on G is equivalent to the LMI (3), building off the fact that the symmetrization of G exactly equals $\frac{1}{2}S$.

Lemma 1 (Equivalence). Consider the vector field f_π from (2), let $M(x) = \Theta(x)^T \Theta(x) + aI$ for $a \geq 0$ be a uniformly bounded Riemannian metric, and $c > 0$ be fixed. The following are equivalent:

- (i) $S(t, x; c) \preceq 0$, with S defined in (3);
- (ii) $\mu_2(G(t, x; a, c)) \leq 0$, with G defined as the asymmetric contraction matrix (4).

Thus, X is a contraction region if and only if $\mu_2(G(t, x; a, c)) \leq 0$ uniformly over $t \in [0, \infty)$, $x \in X$.

Proof. Expanding G and adding its transpose reveals

$$\begin{aligned} G(t, x; a, c) + G(t, x; a, c)^T &= 2c\Theta(x)^T \Theta(x) \\ &+ \Theta(x)^T \frac{\partial f_\pi}{\partial x}(t, x) \Theta(x) + \left(\frac{\partial f_\pi}{\partial x}(t, x) \Theta(x) \right)^T \Theta(x) \\ &+ \Theta(x)^T \Theta(x) \frac{\partial f_\pi}{\partial x}(t, x) + \frac{\partial f_\pi}{\partial x}(t, x)^T \Theta(x)^T \Theta(x) \\ &+ a \frac{\partial f_\pi}{\partial x}(t, x) + a \frac{\partial f_\pi}{\partial x}(t, x)^T + 2acI \end{aligned}$$

Substituting $M(x) = \Theta(x)^T \Theta(x) + aI$ and $\frac{\partial f_\pi}{\partial x}(x) M(x) = \left(\frac{\partial f_\pi}{\partial x}(x) \Theta(x) \right)^T \Theta(x) + \Theta(x)^T \frac{\partial f_\pi}{\partial x}(x) \Theta(x)$ using the matrix product rule shows that $S(t, x; c) = G(t, x; a, c) +$

$G(t, x; a, c)^T$. Thus, $\mu_2(G(t, x; a, c)) = \lambda_{\max}\left(\frac{S(t, x; c)}{2}\right) \leq 0$ if and only if $S(t, x; c)$ has nonpositive eigenvalues. $\square$

We note that the condition $\mu_2(G(t, x; a, c)) \leq 0$ is different from the condition $\mu_2(F(t, x)) \leq -c$ provided in [7, Def. 2], which bounds the logarithmic norm of the *generalized Jacobian* $F(t, x) = (\frac{\partial f_\pi}{\partial x}(t, x) \Theta(x) + \Theta(x) \frac{\partial f_\pi}{\partial x}(t, x)) \Theta(x)^{-1}$ (for the $a = 0$ case). Bounding F would require propagating bounds through a matrix inversion, which is difficult. Instead, the asymmetric contraction matrix (4) requires only a transpose of Θ .

The following analytical example demonstrates how while the two conditions are theoretically equivalent, applying bound propagation strategies to $G(t, x; a, c)$ instead of $S(t, x; c)$ can dramatically reduce overestimation error due to lost information in the symmetrization.

Example 1 (Dependency management). In the scalar case, we can easily compare the expressions (3) and (4),

$$\begin{aligned} S &= \Theta^2 J + J \Theta^2 + \Theta \dot{\Theta} + \dot{\Theta} \Theta + 2c \Theta^2, \\ G &= \Theta(\dot{\Theta} + \Theta(J + c)), \end{aligned}$$

with the notation $\dot{\Theta} = \frac{\partial f_{cl}}{\partial t} \Theta$, and $a = 0$. Since $S = 2G$, this may seem like a simple notational difference. However, recall from Section I-B that bound propagators like interval analysis and CROWN directly trace the above expressions into computation graphs—the resulting computations and uncertainty propagation can yield significant differences between S and G . The expression for G captures interactions between $\dot{\Theta}$ and ΘJ before premultiplying by Θ , while the expression for S multiplies first, introducing redundancies.

For instance, evaluation of the above expressions over $\Theta \in [0.5, 1]$, $\dot{\Theta} \in [-2, -1.5]$, $J \in [-1, 1]$ and $c = 0.5$ gives

$$S \in [-5.75, 1.5] \quad \text{and} \quad 2G \in [-5, 0],$$

which is a significant difference when trying to verify non-positivity (contraction) using Lemma 1.

B. Efficient Parallel Corner Checks

Given an interval matrix $[A]$, one way to solve $\max_{A \in [A]} \mu_2(A)$ is to (i) write $[A] = \text{co}\{A_k\}_{k=1}^{2^{n^2}}$ (co is the convex hull), where the A_k consist of all 2^{n^2} matrices of the form $(A_k)_{ij} \in \{\underline{A}_{ij}, \bar{A}_{ij}\}$, and (ii) note that μ_2 is convex, so $\max_{A \in [A]} \mu_2(A) = \max_{k \in \{1, \dots, 2^{n^2}\}} \mu_2(A_k)$. While this exactly computes the largest logarithmic norm of the set, it scales very poorly in n . Instead, we apply a result from [8] to reduce the number of corner checks down to 2^n . For the $n = 10$ quadrotor in Section IV, for instance, 2^{10^2} is over 10^{30} corners, while 2^{10} is far more practical at 1024 corners.

Lemma 2 ([8, Thm 1]). For an interval matrix $[A] \in \mathbb{IR}^{n \times n}$,

$$\begin{aligned} \max_{A \in [A]} \mu_2(A) \\ = \max \{ \mu_2(A^c + \text{diag}(s) A^\Delta \text{diag}(s)) : s \in \{-1, +1\}^n \}. \end{aligned}$$

The proof is in [8] in the context of verifying the positive definiteness of an interval matrix—we make slight modifications to bound the function $\mu_2(A) = \lambda_{\max}\left(\frac{A+A^T}{2}\right)$ instead.

The essential idea is to decompose $v^T A v$ into $v^T A^c v$ and bound the difference using $|v|^T A^\Delta |v|$, where $|\cdot|$ is the entrywise absolute value. Then, we consider possible choices of $|v|$ instead of possible choices of $A - A^c$.

Proof. We first recall $\mu_2(A) = \max_{v: |v|_2=1} (v^T \frac{A+A^T}{2} v) = \max_{v: |v|_2=1} v^T A v$. Let $A \in [A]$. We can write $v^T A v = v^T A^c v + v^T (A - A^c) v$. Note that for any row a^T and vector v , $|a^T v| \leq |a^T| |v|$. Thus, $|v^T (A - A^c) v| \leq |v^T (A - A^c)| |v| \leq |v|^T |A - A^c| |v| \leq |v|^T A^\Delta |v|$, and therefore $v^T A v \leq v^T A^c v + |v|^T A^\Delta |v|$. Define a sign vector $s(v) \in \{-1, +1\}^n$ by $(s(v))_j = \text{sign}(v_j)$ —then $|v| = \text{diag}(s(v))v$ and

$$\begin{aligned} v^T A v &\leq v^T A^c v + v^T \text{diag}(s(v)) A^\Delta \text{diag}(s(v)) v \\ &= v^T (A^c + \text{diag}(s(v)) A^\Delta \text{diag}(s(v))) v \\ &\leq \max\{v^T (A^c + \text{diag}(s) A^\Delta \text{diag}(s)) v : s \in \{-1, +1\}^n\}. \end{aligned}$$

Taking a max over v with $|v|_2 = 1$ on either side and swapping the order of max, $\mu_2(A) \leq \mu_2(A^c + \text{diag}(s) A^\Delta \text{diag}(s))$ for some sign vector s . The containment $A^c + \text{diag}(s) A^\Delta \text{diag}(s) \in [A]$ implies equality. $\square$

Remark 1. In prior work [6], a sufficient condition to check the negative (semi-)definiteness of an interval matrix $[A]$ constructs a matrix B bounding the Metzler majorant $[A]_{\text{Mzr}}^{-1}$ of any matrix $A \in [A]$, and verifies that this matrix has all eigenvalues with negative real part. However, this sufficient condition is inherently conservative, even for single matrices. For example, $A = -t v v^T$ with $t > 0$ and $v = [1, \dots, 1]^T \in \mathbb{R}^n$ is negative semidefinite yet its Metzler majorant has maximum eigenvalue equal to $t(n-2)$ which is positive for all $n > 2$ and grows unbounded as $t \rightarrow \infty$.

Applying Lemma 2 with an interval hull of the asymmetric contraction matrix over a region X , we can verify that X is a contraction region through 2^n corner checks.

Theorem 1. Let $[G] \in \mathbb{R}^{n \times n}$ be an interval hull of the asymmetric contraction matrix G from (4) over $[0, \infty) \times X$, for fixed $c > 0$ and region $X \subseteq \mathbb{R}^n$. If

$$\max\{\mu_2(G^c + \text{diag}(s) G^\Delta \text{diag}(s)) : s \in \{-1, +1\}^n\} \leq 0,$$

then X is a contraction region for the nonlinear system f_π from (2) at rate c .

Proof. Lemma 2 implies $\max_{x \in X} \mu_2(G(t, x; a, c)) \leq 0$, and Lemma 1 completes the proof. $\square$

C. Loss for Certified Metric and Policy Training

Corollary 1. If $\text{LOSS}(\Theta, \pi; X, a, b, c) \leq 0$ in Algorithm 1, for $a, b, c > 0$, then

- (i) $M(x) = \Theta(x)^T \Theta(x) + aI$ satisfies $aI \preceq M(x) \preceq bI$;
- (ii) X is a contraction region for the closed-loop system $\dot{x} = f_\pi(x) = f_{\text{ol}}(x, \pi(x))$ from (2) at rate c .

Proof. $aI \preceq M(x)$ as $\Theta^T \Theta \succeq 0$. $\text{LOSS}(\Theta, \pi; X, a, b, c) \leq 0$ implies $\lambda \leq 0$ and $\hat{b} - b \leq 0$. By Lemma 2, for every $M \in [M]$, $\lambda_{\max}((M + M^T)/2) \leq \hat{b}$, implying part (i). Part (ii) follows by application of Theorem 1. $\square$

¹ $([A]_{\text{Mzr}})_{ij} = |A_{ij}|$ if $i \neq j$ and $([A]_{\text{Mzr}})_{ii} = A_{ii}$.

Algorithm 1 Certified Neural Contraction Metric Loss

- 1: **Input:** $a, b > 0$ metric bounds, $c > 0$ contraction rate
- 2: **function** $\text{LOSS}(\Theta, \pi; X, a, b, c)$
- 3: $[G] \leftarrow$ interval hull of G from Definition 2 over X
- 4: $\lambda \leftarrow \max_{s \in \{-1, +1\}^n} \mu_2(G^c + \text{diag}(s) G^\Delta \text{diag}(s))$
- 5: $[M] \leftarrow$ interval hull of $\Theta^T \Theta + aI$ over X
- 6: $\hat{b} \leftarrow \max_{s \in \{-1, +1\}^n} \mu_2(M^c + \text{diag}(s) M^\Delta \text{diag}(s))$
- 7: **return** $\max\{\lambda, 0\} + \max\{\hat{b} - b, 0\}$

A crucial step in Algorithm 1 is obtaining the interval hulls $[G]$ and $[M]$ containing the output of (4) over an input set X . In practice, we found that linear bound propagation in the style of CROWN [10], [19] provides the right balance of accuracy, scalability, and trainability. We use the implementation provided in our toolbox `immrax` [20]² in JAX [21]. JAX allows us to automatically (i) construct the asymmetric contraction matrix (4) using automatic differentiation, (ii) trace the computation graph applying linear bound propagation, and (iii) backpropagate the loss from Algorithm 1 and dispatch the neural network training to the GPU.

III. EXPLICIT TRACKING CONTROL DESIGN

Consider the following control-affine nonlinear system,

$$\dot{x} = f_{\text{ol}}(x, u) = f_d(x) + g(x)u, \quad (5)$$

where $f_d : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a C^1 drift term and $g(x) \in \mathbb{R}^{n \times m}$ is a C^1 state dependent matrix. With some restrictions on the choice of metric $M(x)$, we can apply the methodology from the previous section to design a simple feedback controller to exponentially track any dynamically feasible trajectory within the contraction region X . We assume this dynamically feasible trajectory is precomputed using a motion planning algorithm. Notably, we do not require any online geodesic computations like [3], [22], or state augmentation with the trajectory to be tracked like previous learning methods [4].

Corollary 2. Suppose $\text{LOSS}(\Theta, \pi; X, a, b, c) \leq 0$ for the nonlinear system $\dot{x} = f_d(x) + g(x)u$ with hyperparameters $a, b, c > 0$, satisfying the following condition: for every $j = 1, \dots, m$ and every $x \in X$,

$$\Theta(x)^T \partial_{g_j(x)} \Theta(x) + (\Theta(x)^T \Theta(x) + aI) \frac{\partial g_j}{\partial x}(x) = 0. \quad (6)$$

Let $x'(t)$ be a trajectory associated with a nominal input curve $u'(t)$. If there exists an $R > 0$ such that the metric tube of radius R around $x'(t)$ is fully contained inside $\text{int } X$, i.e., for every $t \geq 0$, $\{x : d(x, x'(t)) \leq R\} \subseteq \text{int } X$, then any trajectory $x(t)$ under the feedback policy

$$u(t, x) = \pi(x) - \pi(x'(t)) + u'(t) \quad (7)$$

with initial condition $d(x(0), x'(0)) \leq R$ satisfies, for $t \geq 0$,

$$d(x(t), x'(t)) \leq e^{-ct} d(x(0), x'(0)).$$

²<https://github.com/gtfactslab/immrax>

Proof. Letting $v(t) = u'(t) - \pi(x'(t))$, and $f_\pi(t, x) = f_a(x) + g(x)\pi(x)$, the closed-loop dynamics are

$$\dot{x} = f_{cl}(t, x) = f_\pi(t, x) + \sum_{j=1}^m g_j(x)v_j(t). \quad (8)$$

Computing the asymmetric contraction matrix from Definition 2, by linearity, for $t \in [0, \infty)$, $x \in X$,

$$G_{cl}(t, x; a, c) = G_\pi(t, x; a, c) + \sum_{j=1}^m [\Theta(x)^T \partial_{g_j(x)} \Theta(x) + \Theta(x)^T \Theta(x) \frac{\partial g_j}{\partial x}(x) + a \frac{\partial g_j}{\partial x}(x)] v_j(t)$$

where $G_\pi(t, x; a, c)$ is the asymmetric contraction matrix (Definition 2) of f_π . The condition $\text{LOSS}(\Theta, \pi; X, a, b, c) \leq 0$ implies that we have $\mu_2(G_\pi(t, x)) \leq 0$ by Corollary 1. By assumption (6), the bracketed term vanishes. Thus, subadditivity of $\mu_2(\cdot)$ implies $\mu_2(G_{cl}(t, x; a, c)) \leq \mu_2(G_\pi(t, x)) + \mu_2(0) \leq 0$. Therefore, Lemma 1 implies X is a contraction region for the closed-loop system f_{cl} (8) at rate c .

To conclude, since $u(t, x'(t)) = u'(t)$, the uniqueness of solutions implies $x'(t)$ is the trajectory of f_{cl} from initial condition $x'(0)$. Because x' is a trajectory of a contracting system with a metric tube of radius R around it contained in the open and connected set $\text{int } X$, there is a minimizing geodesic between x and x' contained in $\text{int } X$ which shrinks exponentially at rate c [7, Thm 1]. $\square$

Remark 2 (Geodesic metric tube). The tube of radius R in Corollary 2 is analytically important, ensuring geodesic paths connecting the two trajectories are always contained within the contraction region. However, computationally finding such sublevel sets of the geodesic metric is difficult due to the need to search for geodesics. For simpler guarantees, one could apply the uniform bounds on the metric to obtain ℓ_2 -ball tubes sandwiching a geodesic metric tube: if $\{x : \|x - x'(t)\|_2 \leq \bar{R}\} \subseteq \text{int } X$ for $\bar{R} > 0$, then the radius $R = \sqrt{a\bar{R}}$ geodesic metric ball is also contained in $\text{int } X$, and points satisfying $\|x - x'(0)\|_2 \leq \frac{\sqrt{a}}{\sqrt{b}}\bar{R}$ live within the radius R ball at time 0. In practice, we find that controllers trained to satisfy the contraction condition within region X have good behavior when geodesics and even the trajectories themselves leave X (see Section IV). This may be due to the overestimation induced by bound propagation making the closed-loop system more robust than we can formally certify.

Remark 3 (Killing field condition). The condition (6) is the asymmetric analogue to the strong CCM condition [3, C2] which requires the actuation vector fields g_j to be Killing fields of the metric $M(x) = \Theta(x)^T \Theta(x) + aI$,

$$\partial_{g_j(x)} M(x) + M(x) \frac{\partial g_j}{\partial x}(x) + \frac{\partial g_j}{\partial x}(x)^T M(x) = 0,$$

which can be seen by a computation similar to Lemma 1.

For systems with a constant $g(x) = B$ for $B \in \mathbb{R}^{n \times m}$, we can augment the neural network with a linear layer projecting x into the orthogonal subspace to B , using a matrix $B^\perp$ satisfying $B^\perp B = 0$. For instance, when $B = [0, I]^T$, restricting the argument of Θ to the first $n - m$ components of x structurally satisfies (6).

In the general non-constant case, enforcing constraints such as this on the neural network Θ is generally challenging. With

dynamic extension, any nonlinear system can be transformed into this form using an integral control law.

Remark 4 (Control contraction metrics (CCM) and complete integrability). Following the discussion in [3, p. 3], our learned metric $M(x) = \Theta(x)^T \Theta(x)$ and policy π provide a *completely integrable* differential control law $\delta u = \frac{\partial \pi}{\partial x}(x) \delta x$, which exponentially stabilizes the variational dynamics $\delta \dot{x} = (\frac{\partial f_a}{\partial x}(x) + \sum_{j=1}^m \frac{\partial g_j}{\partial x}(x) u'_j) \delta x + g(x) \delta u$, along trajectories (x', u') . This can be seen by evaluating LMI (3) directly on the closed-loop dynamics f_{cl} (8), and simplifying the resulting expression using the Killing field condition discussed in Remark 3 to obtain the following:

$$M(x) \left(\frac{\partial f_a}{\partial x}(x) + g(x) \frac{\partial \pi}{\partial x}(x) \right) + (\star)^T M(x) + \partial_{f_a(x) + g(x)\pi(x)} M(x) + 2cM(x) \leq 0,$$

which verifies the quadratic (Finsler-)Lyapunov condition on $V(x, \delta x) = \delta x^T M(x) \delta x$. In contrast, the convex criteria from CCM [3] instead require only a *path integrable* differential controller $\delta u = K(t, x) \delta x$, where K may not be the Jacobian of an explicit function. This comes at the cost of an *implicit* control policy, which requires an online search for a geodesic path between the current state and the reference. Instead, our controller provides an *explicit* representation, where two simple forward inferences of the policy π provide the control input at minimal computational cost.

IV. EXAMPLE: CONTRACTING TRACKING CONTROL FOR 10 STATE QUADROTOR

In this section, we use Algorithm 1 to design a contracting tracking controller for the 10-state quadrotor from [22],³

$$\ddot{p}_x = -\tau \sin \theta, \quad \ddot{p}_y = \tau \cos \theta \sin \phi, \quad \ddot{p}_z = g - \tau \cos \theta \cos \phi,$$

with $x = [p_x, p_y, p_z, \dot{p}_x, \dot{p}_y, \dot{p}_z, \tau, \phi, \theta, \psi]^T$, $[p_x, p_y, p_z]^T \in \mathbb{R}^3$ in North-East-Down convention, $[\phi, \theta, \psi]^T \in \mathbb{R}^3$ the XYZ Euler-angle representation, $\tau \in \mathbb{R}$ the net (mass-normalized) thrust, and $u = [\dot{\tau}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ is the control input.

For the controller, we learn a network with 2 hidden layers of 32 neurons returning a matrix $N(x) \in \mathbb{R}^{4 \times 11}$, and set $\pi(x) = N(x) \begin{bmatrix} x \\ 1 \end{bmatrix}$ (affine parameterization). For the metric, we learn a network with 2 hidden layers of 32 neurons returning the upper triangle of the matrix $\Theta(x) \in \mathbb{R}^{10 \times 10}$.

We trained the contraction region $X = [-10, 10]^3 \times [-5, 5]^3 \times [\frac{2g}{3}, \frac{4g}{3}] \times [-\frac{\pi}{8}, \frac{\pi}{8}]^2 \times [-\frac{\pi}{2}, \frac{\pi}{2}]$ by optimizing over growing regions $\frac{n}{100}X$, where we checkpoint and iterate n when the loss hits 0 until $n = 100$. We partition each n -th region uniformly into $10^2 = 100$ partitions over the (ϕ, θ) states. We use the AdamW optimizer with the loss from Algorithm 1, with $a = 1$, $b = 50$, $c = 0.1$. In total, the training took a total of 10479 steps in 644 seconds, including 62 seconds for JIT compilation.⁴

We use the differential flatness property of quadrotor dynamics to extract a dynamically feasible pair $(x'(t), u'(t))$ based on a desired C^3 curve $[p'_x(t), p'_y(t), p'_z(t), \psi'(t)]^T$.

³The code reproducing the results is available at https://github.com/gtfactslab/neural_contraction

⁴Code was run on a desktop running Ubuntu 22.04, with AMD Ryzen 5 5600X, NVIDIA RTX 3070, and 32 GB of RAM.

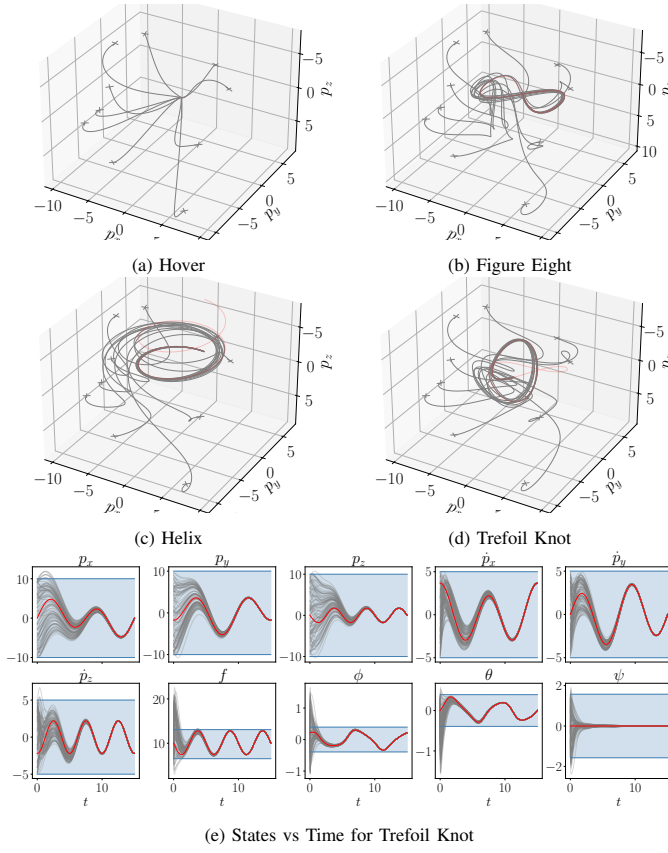


Fig. 1: Plots (a)-(d) show the (p_x, p_y, p_z) states for 10 randomly sampled initial conditions from the verified contraction region X (pictured with the $\times$) track several nominal trajectories (pictured in red) under the feedback policy $u(t, x) = \pi(x) - \pi(x'(t)) + u'(t)$ from Corollary 2. Plot (e) shows the state vs time for 100 randomly sampled initial conditions starting in X for the trefoil knot (d). The nominal trajectory is in red, with the verified contraction region X in blue. While some trajectories leave the verified region, the controller empirically behaves well, bringing them back into the region and stabilizing to the commanded trajectory.

Then, we apply the explicit feedback policy derived in Corollary 2 to track this trajectory. Compared to [4], we get a formal guarantee of contraction over the whole region X , which was intractable with sampling-based guarantees for $n = 10$, and we completely avoid the need to search for minimizing geodesics like the approach from [22].

V. CONCLUSION AND FUTURE WORK

In this letter, we presented a framework for jointly learning a neural network controller and a neural contraction metric with formal guarantees of closed-loop contraction over a region X . By leveraging a novel asymmetric contraction condition, GPU-parallelized corner checks, and end-to-end automatic differentiation, our approach avoids dependency management overconservativeness and sample complexity issues associated to Lipschitz-based guarantees. For control-affine systems, we also developed an explicit tracking controller requiring only two forward inferences of the network.

Several directions remain for future work. Incorporating forward invariance guarantees for the certified region and enforcing bounded control inputs would strengthen the practical applicability of the framework. More broadly, extending the approach to learn Finsler-Lyapunov functions or to certify

contraction with respect to non-Euclidean norms could further expand its scope to systems where Riemannian metrics are not the most natural choice.

REFERENCES

- [1] 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.
- [2] C. Liu, T. Armon, C. Lazarus, C. Strong, C. Barrett, and M. J. Kochenderfer, "Algorithms for verifying deep neural networks," *Foundations and Trends in Optimization*, vol. 4, no. 3-4, pp. 244–404, 2021.
- [3] I. R. Manchester and J.-J. E. Slotine, "Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, 2017.
- [4] D. Sun, S. Jha, and C. Fan, "Learning certified control using contraction metric," in *Conference on Robot Learning*, pp. 1519–1539, PMLR, 2021.
- [5] M. Zakwan, L. Xu, and G. Ferrari-Trecate, "Neural exponential stabilization of control-affine nonlinear systems," in *IEEE Conference on Decision and Control (CDC)*, pp. 8602–8607, 2024.
- [6] A. Davydov, "Verifying closed-loop contractivity of learning-based controllers via partitioning," in *Asilomar Conference on Signals, Systems, and Computers*, pp. 1740–1744, 2025.
- [7] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.
- [8] J. Rohn, "Positive definiteness and stability of interval matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 15, no. 1, pp. 175–184, 1994.
- [9] M. Fitzsimmons and J. Liu, "Computation and formal verification of neural network contraction metrics," *IEEE Control Systems Letters*, vol. 8, pp. 2367–2372, 2024.
- [10] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [11] F. Forni and R. Sepulchre, "A differential Lyapunov framework for contraction analysis," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 614–628, 2013.
- [12] A. Davydov, S. Jafarpour, and F. Bullo, "Non-Euclidean contraction theory for robust nonlinear stability," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6667–6681, 2022.
- [13] E. D. Sontag, "Contractive systems with inputs," in *Perspectives in Mathematical System Theory, Control, and Signal Processing: A Festschrift in Honor of Yutaka Yamamoto on the Occasion of his 60th Birthday*, pp. 217–228, Springer, 2010.
- [14] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 1st edition, ed., 2001.
- [15] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, "On the effectiveness of interval bound propagation for training verifiably robust models," *arXiv preprint arXiv:1810.12715*, 2018.
- [16] M. Althoff, G. Frehse, and A. Girard, "Set propagation techniques for reachability analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 369–395, 2021.
- [17] H.-D. Tran, D. Manzananas Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson, "Star-based reachability analysis of deep neural networks," in *International symposium on formal methods*, pp. 670–686, Springer, 2019.
- [18] X. Chen, *Reachability Analysis of Non-Linear Hybrid Systems Using Taylor models*. PhD Thesis, Fachgruppe Informatik, RWTH Aachen University, 2015.
- [19] K. Xu, Z. Shi, H. Zhang, Y. Wang, K.-W. Chang, M. Huang, B. Kaillkhura, X. Lin, and C.-J. Hsieh, "Automatic perturbation analysis for scalable certified robustness and beyond," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1129–1141, 2020.
- [20] A. Harapanahalli, S. Jafarpour, and S. Coogan, "immrax: A parallelizable and differentiable toolbox for interval analysis and mixed monotone reachability in JAX," *IFAC-PapersOnLine*, vol. 58, no. 11, pp. 75–80, 2024.
- [21] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018.
- [22] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust feedback motion planning via contraction theory," *The International Journal of Robotics Research*, vol. 42, no. 9, pp. 655–688, 2023.