Learning Stability Certificates from Data

Nicholas M. Boffi
Harvard University

Stephen Tu
Google Brain Robotics

Nikolai Matni
University of Pennsylvania

Jean-Jacques E. Slotine
MIT

Vikas Sindhwani
Google Brain Robotics

Abstract: Many existing tools in nonlinear control theory for establishing stability or safety of a dynamical system can be distilled to the construction of a *certificate function* that guarantees a desired property. However, algorithms for synthesizing certificate functions typically require a closed-form analytical expression of the underlying dynamics, which rules out their use on many modern robotic platforms. To circumvent this issue, we develop algorithms for learning certificate functions only from trajectory data. We establish bounds on the generalization error – the probability that a certificate will not certify a new, unseen trajectory – when learning from trajectories, and we convert such generalization error bounds into global stability guarantees. We demonstrate empirically that certificates for complex dynamics can be efficiently learned, and that the learned certificates can be used for downstream tasks such as adaptive control.

Keywords: Generalization bounds, Lyapunov functions, contraction metrics

1 Introduction

A fundamental barrier to widespread deployment of reinforcement learning policies on real robots is the lack of formal safety and stability guarantees. While much research has focused on how to train control policies for complex systems, considerably less emphasis has been placed on verifying stability for the resulting closed-loop system. Without any a-priori guarantees, practitioners will be hesitant to deploy learned solutions in the real world regardless of performance in simulation.

Many powerful tools have been developed in nonlinear control theory to address the safety and stability of systems with known dynamics. The most well-known technique is the construction of a Lyapunov function [1, 2] to demonstrate asymptotic stability of a system with respect to an equilibrium point. Similarly, barrier functions [3–5] are used to show set-invariance, which has been widely used in safety-critical applications to prove that a system does not exit a desired safe set. Contraction analysis [6] provides an alternative view of stability, applicable to many problems in nonlinear control and robotics, by considering the convergence of trajectories towards each other rather than to an equilibrium point. The unifying theme among these tools is the construction of a *certificate function* (the Lyapunov/barrier function or contraction metric) that proves a given desirable property for the system of interest. These certificates have strong converse results [7–9], which imply the existence of a certificate function if the desired property does hold, and can also be used for controller synthesis [4, 10, 11].

The main obstacle for producing certificate functions in modern robotics and reinforcement learning is that existing synthesis and verification tools such as sum-of-squares (SOS) optimization [12] or SMT solvers [13] typically assume the dynamics can be written down analytically in closed form. Furthermore, the functions of interest are often constrained to lie in restrictive classes such as polynomial basis functions of fixed degree. This presents a serious hurdle in modern robotics, where (a) sophisticated physics simulators are widely used to model complex environments and (b) control policies are often represented with complex deep neural networks. Finally, both SOS optimization and formal verification tools are computationally intensive, thus limiting their applicability.

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

To avoid these limitations, recent approaches have proposed to treat certificate synthesis as a machine learning problem, and train powerful function approximators such as deep neural networks and reproducing kernel Hilbert space (RKHS) predictors on trajectory data collected from a dynamical system [14–19]. The general strategy is to enforce the desired certificate condition (e.g. the Lie derivative of a function V should be negative) along collected samples. Empirically, this has been shown to be quite effective, and the learned certificate often generalizes well outside of the training data. However, a deeper theoretical understanding of when and why this approach works is missing.

Contributions. Consider Figure 1, where a contraction metric, which certifies pairwise convergence of trajectories, is learned from rollouts of a damped Van der Pol oscillator. Regions of the state space for which the learned metric is *not* contracting are shown as a function of the number of trajectories n. While the size of the violating regions appears to shrink as n increases, Figure 1 raises many questions. How much data does one need to collect so that the violating regions cover at most a prescribed fraction of the relevant state space? Is the learning consistent, i.e. do the regions vanish as $n \to \infty$?

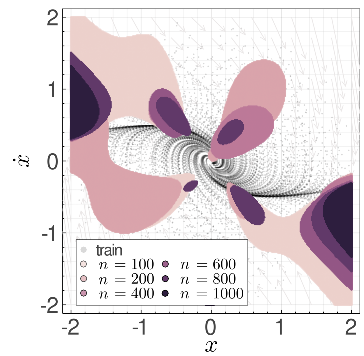

Figure 1: Violation sets for contraction metric learning on the damped Van der Pol oscillator.

In this paper, we show that learning is indeed consistent. To this end, we compute upper bounds on the volume of the violating regions which tend to zero as n grows. We do this in two steps. First, we formulate a general optimization framework that encompasses learning many existing certificate functions, and use statistical learning theory to prove a fast $\tilde{O}(k/n)$ rate on the generalization error – the probability the learned certificate will not certify a new, unseen trajectory – where k is the effective number of parameters of the function class for the certificate. We then translate bounds on the generalization error into non-probabilistic bounds on the volume of the violating regions. We conclude with experiments, which show that certificates can be efficiently learned from trajectories, and that the learned certificates can perform downstream tasks such as adaptive control against unknown disturbances. Full proofs and more experimental details can be found in the extended version of the paper [20].

2 Related Work

Prior research generally focuses on learning certificates for a fixed system from trajectories, or on using certificate conditions as regularizers when learning models for control.

Learning Lyapunov functions from data. Giesl et al. [21] propose to learn a Lyapunov function from noisy trajectories using a specific reproducing kernel. Their algorithm first fits a dynamics model from data, and then uses interpolation to construct a Lyapunov function from the learned model. The authors prove L_∞ convergence results on the Lie derivative of the constructed Lyapunov function compared to the ground truth, with rates depending on a dense cover of the state space.

Our work circumvents this two-step identification procedure by directly analyzing the generalization error of a Lyapunov function learned by enforcing derivative conditions along the training data. Many other authors have proposed similar approaches. Kenanian et al. [22] show how to estimate the joint spectral radius of a switched linear system by learning a common quadratic Lyapunov function directly from data. Their analysis heavily exploits properties of linear systems. Chen et al. [23] study how to learn a quadratic Lyapunov function for piecewise affine systems in feedback with a neural network controller. Richards et al. [14] use a sum-of-squares neural network representation to learn the largest region of attraction of a nonlinear system. Manek and Kolter [15] jointly train a neural network model and Lyapunov function. Neither Richards et al. [14] nor Manek and Kolter [15] provide formal guarantees that the learned Lyapunov function will generalize to new trajectories. Both Chang et al. [24] and Ravanbakhsh and Sankaranarayanan [25] propose to use ideas from formal verification to falsify the validity of a learned candidate Lyapunov function. A significant limitation is the requirement of access to the true dynamics.

In many of these works, the Lie derivative constraint that defines a Lyapunov function is relaxed to a soft constraint, so that first-order gradient methods can be used for optimization. We note that our generalization analysis can be modified to handle soft constraints in a straightforward manner.

Learning barrier functions from data. Barrier functions are relaxations of Lyapunov functions that demonstrate invariance of a subset of the state space. Recently, many authors have proposed to use and learn barrier functions from data for safety-critical applications. Taylor et al. [16] assume a control barrier function (CBF) is valid for both a nominal and unknown system model, and use the CBF to guide safe learning of the unknown system dynamics. More closely related to our work, Robey et al. [17] learn a CBF for a known nonlinear dynamical system from expert demonstrations, and use Lipschitz arguments to extend the validity of the CBF beyond the training data. Jin et al. [18] propose to jointly learn a Lyapunov, barrier, and a policy function from data. They also prove validity of the learned certificates using Lipschitz arguments.

Learning contracting vector fields and contraction metrics from data. The literature on learning contraction metrics [6] from data is more sparse. In an imitation learning context, Sindhwani et al. [26] propose to learn a vector field from demonstrations that satisfies contraction in the identity metric. The authors parameterize the vector field as a vector-valued reproducing kernel. Khadir et al. [27] also learn a vector field from demonstrations by using sum-of-squares to enforce contraction. They argue by smoothness considerations that the learned vector field actually contracts in a tube around the demonstration trajectories. We note that in both these works, the metric is held fixed and is assumed to be known. Singh et al. [19] jointly learn a model and a control contraction metric [28, 29] from data, and show empirically that using contraction as a regularizer in model learning can lead to better sample efficiency when learning to control. We leave studying the generalization properties of jointly learning an explicit model and a contraction metric to future work.

Statistical bounds in optimization and control. Our generalization bounds are similar in spirit to those provided for random convex programs (RCPs) [30, 31]. Random convex programming is concerned with approximating solutions to convex programs with an infinite number of constraints. Such infinitely-constrained problems are approximated by drawing n i.i.d. samples from a distribution ν over the constraint parameters and enforcing constraints on samples. One can then show that the probability that a new sample from ν violates the constraint for the approximate solution scales as $O(d/n)$ where d is the number of decision variables. Our results can be viewed as generalizing these bounds beyond convex programs, though our constants are less sharp. In our experiments, we use the RCP bound for numerically computing generalization bounds when the problem is convex.

3 Learning Certificates Framework

3.1 Problem Statement

We assume the underlying dynamical system is given by a continuous-time autonomous system of the form $\dot{x} = f(x)$, where f is continuous, unknown, and the state $x \in \mathbb{R}^p$ is fully observed. Let $X \subseteq \mathbb{R}^p$ be a compact set and let $T \subseteq \mathbb{R}_+$ be the maximal interval starting at zero for which a unique solution $\varphi_t(\xi)$ exists for all initial conditions $\xi \in X$ and $t \in T$. We assume access to sample trajectories generated from random initial conditions. Specifically, let $\mathcal{D}$ denote a distribution over X, and let $\xi_1, ..., \xi_n$ be n i.i.d. samples from $\mathcal{D}$. We are given access to the n trajectories $\{\varphi_t(\xi_i)\}_{i=1,...,n, t\in T}$. For simplicity of exposition, we assume that we can exactly differentiate the trajectories $\varphi_t(\xi)$ with respect to time. In our experiments, we compute $\dot{x}$ numerically.

Let $\mathcal{V}$ be a space of continuously differentiable functions $V : \mathbb{R}^p \mapsto \mathbb{R}^q$. Let $h : \mathbb{R}^p \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^{q\times p} \mapsto \mathbb{R}$ be a fixed and known continuous function. Our goal is to choose a $V \in \mathcal{V}$ such that

$$h\left(\varphi_t(\xi), \dot{\varphi}_t(\xi), V(\varphi_t(\xi)), \frac{\partial V}{\partial x}(\varphi_t(\xi))\right) \le 0 \;\; \forall \xi \in X,\ t \in T\,. \tag{3.1}$$

As we describe below, through suitable choices of the function h, equation (3.1) can be used to enforce various defining conditions for certificates such as Lyapunov functions and contraction metrics. We note that our framework can be modified to allow for more derivatives of V, including higher order derivatives and also time derivatives for handling time-varying dynamics.

We study the following optimization problem for searching for a solution to (3.1):

$$\mathrm{find}_{V\in\mathcal{V}} \text{ s.t. } h\left(\varphi_t(\xi_i), \dot{\varphi}_t(\xi_i), V(\varphi_t(\xi_i)), \frac{\partial V}{\partial x}(\varphi_t(\xi_i))\right) \le -\gamma\,,\ \ i = 1, ..., n,\ t \in T\,. \tag{3.2}$$

Here, $\gamma > 0$ is a positive margin value which will allow us to generalize the behavior of V on h outside of the sampled data. In practice, we often solve (3.2) with a cost term on V such as its norm. Let $\hat{V}_n \in \mathcal{V}$ denote a solution to (3.2), assuming one exists. We quantify the *generalization* of $\hat{V}_n$ by the probability of violation over trajectories starting from $\xi \sim \mathcal{D}$:

$$\mathsf{err}(\hat{V}_n) := \mathbb{P}_{\xi\sim\mathcal{D}}\left\{\max_{t\in T} h\left(\varphi_t(\xi), \dot{\varphi}_t(\xi), V(\varphi_t(\xi)), \frac{\partial V}{\partial x}(\varphi_t(\xi))\right) > 0\right\} . \tag{3.3}$$

In Section 4, we prove $O(k \cdot \mathrm{polylog}(n)/n)$ decay rates for $\mathsf{err}(\hat{V}_n)$ for various parametric and non-parametric function classes $\mathcal{V}$, where k denotes the effective number of parameters of the class $\mathcal{V}$. In Section 5, we show how $\mathsf{err}(\hat{V}_n) \le \varepsilon$ bounds translate into global, non-probabilistic results. Before we state our main results, we instantiate our framework for two key certificate functions.

3.1.1 Lyapunov stability analysis

Let zero be an equilibrium point for $\dot{x} = f(x)$. Let $D \subseteq \mathbb{R}^p$ be an open set containing the origin. A Lyapunov function $V : \mathbb{R}^p \mapsto \mathbb{R}$ is a locally positive definite function such that $V(0) = 0$, $V(x) > 0$ for $x \in D \setminus \{0\}$, and $\langle \nabla V(x), f(x)\rangle < 0$ for $x \in D \setminus \{0\}$. It is well known (see e.g. Slotine and Li [2]) that the existence of such a Lyapunov function V proves the local asymptotic stability of the origin. Our framework can be used to learn a Lyapunov function from stable trajectories by taking $h(x, \dot{x}, V(x), \nabla V(x)) = \langle \nabla V(x), \dot{x}\rangle + \alpha(V(x))$. Here, $\alpha : \mathbb{R} \mapsto \mathbb{R}$ is a class $\mathcal{K}$ function, i.e. a continuous, strictly increasing function satisfying $\alpha(0) = 0$.

3.1.2 Contraction metrics

A system is said to be contracting in a region D with rate α if there exists a uniformly positive definite Riemannian metric $M(x)$ such that $\frac{\partial f}{\partial x}(x)^{\mathsf{T}} M(x) + M(x)\frac{\partial f}{\partial x}(x) + \dot{M}(x) \preceq -2\alpha M(x)$ for $x \in D$ [6]. Given knowledge of $\frac{\partial f}{\partial x}$, this condition fits into our framework by taking $h\left(x, \dot{x}, M(x), \frac{\partial f}{\partial x}(x)\right) = \lambda_{\max}\left(\frac{\partial f}{\partial x}(x)^{\mathsf{T}} M(x) + M(x)\frac{\partial f}{\partial x}(x) + \dot{M}(x) + 2\alpha M(x)\right)$.

Without knowledge of $\frac{\partial f}{\partial x}$, it is not immediately clear how to evaluate h from trajectories. Instead, we leverage results from Forni and Sepulchre [32], who reformulate contraction in terms of Lyapunov theory. Consider a candidate differential Lyapunov function $V(x, \delta x) = \delta x^{\mathsf{T}} M(x)\delta x$ for the *prolongated system* $\begin{bmatrix}\dot{x}\\ \delta\dot{x}\end{bmatrix} = \begin{bmatrix} f(x) \\ \frac{\partial f}{\partial x}(x)\delta x\end{bmatrix}$ defined on the tangent bundle $\mathcal{T}D = \cup_{x\in D}\{x\} \times \mathcal{T}_x D \simeq D \times \mathbb{R}^p$. The contraction condition is equivalent to:

$$\langle \nabla_x V(x, \delta x), f(x)\rangle + \langle \nabla_{\delta x} V(x, \delta x), \frac{\partial f}{\partial x}(x)\delta x\rangle \le -\alpha V(x, \delta x) \;\; \forall x \in D \, , \delta x \in \mathbb{R}^p \, . \tag{3.4}$$

We can enforce (3.4) by directly sampling trajectories on $\mathcal{T}D$, by exploiting that the *variational dynamics* obeyed by $\delta x(t)$ is identical to the local linearization of f around $x(t)$. Specifically, we sample pairs of initial conditions $x_0^{(1)}$ and $x_0^{(2)} = x_0^{(1)} + \delta x_0$ for some small perturbation δx_0. Numerical differentiation of $x^{(1)}(t)$ and $\delta x(t) = x^{(1)}(t) - x^{(2)}(t)$ provides access to $\dot{x}^{(1)} = f(x^{(1)})$ and $\delta\dot{x}(t) = \frac{\partial f}{\partial x}\delta x(t)$, which then allows us to evaluate (3.4) along system trajectories.

4 Generalization Error Results

We first define the notion of stability we will assume. Recall that X is the set containing sample initial conditions, and T is the interval over which our trajectories evolve.

Assumption 4.1 (Stability in the sense of Lyapunov)**.** *We assume there exists a compact set $S \subseteq \mathbb{R}^p$ such that $\varphi_t(\xi) \in S$ for all $\xi \in X$, $t \in T$. Let the constant $B_S := \sup_{x\in S}\|x\|$.*

Note that contraction implies Assumption 4.1, so that contracting systems are also covered in this setting. Next, we make some regularity assumptions on the function class $\mathcal{V}$.

Assumption 4.2 (Uniform boundedness of $\mathcal{V}$)**.** *We assume there exist finite constants B_V, $B_{\nabla V}$ such that* $\sup_{V\in\mathcal{V}}\sup_{x\in S}\|V(x)\| \le B_V$ *and* $\sup_{V\in\mathcal{V}}\sup_{x\in S}\left\|\frac{\partial V}{\partial x}(x)\right\| \le B_{\nabla V}$.

Given Assumptions 4.1–4.2, we define B_h (resp. L_h) to be an upper bound on $|h(x, f(x), V, \frac{\partial V}{\partial x})|$ (resp. the Lipschitz constant of $(V, \frac{\partial V}{\partial X}) \mapsto h(x, f(x), V, \frac{\partial V}{\partial X})$) over $x \in S$, $|V| \le B_V$ and $\|\frac{\partial V}{\partial x}\| \le B_{\nabla V}$. Note that both B_h and L_h are guaranteed to be finite by our assumptions.

We now introduce, with slight abuse of notation, the shorthand $h(\xi, V)$ for $\xi \in X$, $V \in \mathcal{V}$ as $h(\xi, V) := \max_{t \in T} h\left(\varphi_t(\xi), \dot{\varphi}_t(\xi), V(\varphi_t(\xi)), \frac{\partial V}{\partial x}(\varphi_t(\xi))\right)$. The key insight to our analysis is the simple observation that any feasible solution $\hat{V}_n$ to (3.2) achieves zero empirical risk on the loss $\hat{R}_n(V) := \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{\{h(\xi_i, V) > -\gamma\}}$. In particular, since $\mathbb{P}_{\xi \sim \mathcal{D}}(h(\xi, \hat{V}_n) > 0) = \mathbb{E}_{\xi \sim \mathcal{D}} \mathbf{1}_{\{h(\xi, \hat{V}_n) > 0\}}$, we can use results from statistical learning theory which give us fast rates for zero empirical risk minimizers with margin γ. The following result is adapted from Theorem 5 of Srebro et al. [33].

Lemma 4.1. *Fix a $\delta \in (0, 1)$. Assume that Assumption 4.1 and Assumption 4.2 hold. Suppose that the optimization problem (3.2) is feasible and let $\hat{V}_n$ denote a solution. The following statement holds with probability at least $1 - \delta$ over the randomness of $\xi_1, ..., \xi_n$ drawn i.i.d. from $\mathcal{D}$:*

$$\mathbb{P}_{\xi \sim \mathcal{D}}(h(\xi, \hat{V}_n) > 0) \le K\left(\frac{\log^3 n}{\gamma^2}\mathcal{R}_n^2(\mathcal{V}) + \frac{2\log(\log(4B_h/\gamma)/\delta)}{n}\right) .$$

Here, $\mathcal{R}_n(\mathcal{V}) := \sup_{\xi_1, \ldots, \xi_n \in X} \mathbb{E}_{\varepsilon \sim \mathrm{Unif}(\{\pm 1\}^n)} \sup_{V \in \mathcal{V}} \frac{1}{n} |\sum_{i=1}^n \varepsilon_i h(\xi_i, V)|$ is the Rademacher complexity of the function class $\mathcal{V}$ and K is a universal constant.

Lemma 4.1 reduces bounding $\mathsf{err}(\hat{V}_n)$ to bounding the Rademacher complexity $\mathcal{R}_n(\mathcal{V})$. Define the norm $\|\cdot\|_{\mathcal{V}}$ on $\mathcal{V}$ as $\|V\|_{\mathcal{V}} := \sup_{x \in S} \left\| \begin{bmatrix} V(x) \\ \frac{\partial V}{\partial x}(x) \end{bmatrix} \right\|$. By Assumptions 4.1–4.2 and Dudley's entropy inequality [34], we can bound $\mathcal{R}_n(\mathcal{V})$ by the estimate $\mathcal{R}_n(\mathcal{V}) \le \frac{24 L_h}{\sqrt{n}} \int_0^\infty \sqrt{\log N(\varepsilon; \mathcal{V}, \|\cdot\|_{\mathcal{V}})}\, d\varepsilon$. Here, $N(\varepsilon; \mathcal{V}, \|\cdot\|_{\mathcal{V}})$ is the covering number of $\mathcal{V}$ at resolution ε in the $\|\cdot\|_{\mathcal{V}}$-norm. We use this strategy to obtain generalization bounds for (3.2) over various representations. For ease of exposition we assume that $q = 1$, i.e. $V : \mathbb{R}^p \mapsto \mathbb{R}$. The extension to $q > 1$ is straightforward.

4.1 Lipschitz parametric function classes

We consider the following parametric representation:

$$\mathcal{V} = \{V_\theta(\cdot) = g(x, \theta) : \theta \in \mathbb{R}^k, \|\theta\| \le B_\theta\} . \tag{4.1}$$

We assume $g : \mathbb{R}^p \times \mathbb{R}^k \mapsto \mathbb{R}$ is twice continuously differentiable, which implies that $\mathcal{V}$ satisfies Assumption 4.2. The parameterization (4.1) is very general and encompasses function classes such as neural networks with differentiable activation functions. Furthermore, Dudley's estimate combined with a volume comparison argument yields $\mathcal{R}_n(\mathcal{V})^2 \le O(k/n)$, which implies the following result.

Theorem 4.2. *Under Assumption 4.1, if problem (3.2) over the parametric function class (4.1) is feasible, then any solution $\hat{V}_n$ satisfies with probability at least $1 - \delta$ over $\xi_1, ..., \xi_n$:*

$$\mathsf{err}(\hat{V}_n) \le O(1)\left(B_\theta^2 (L_g + L_{\nabla g})^2 L_h^2 \frac{k \log^3 n}{\gamma^2 n} + \frac{\log(\log(B_h/\gamma)/\delta)}{n}\right) . \tag{4.2}$$

Here, $L_g := \sup_{x \in S, \|\theta\| \le B_\theta} \|\nabla_\theta g(x, \theta)\|$ and $L_{\nabla g} := \sup_{x \in S, \|\theta\| \le B_\theta} \|\frac{\partial^2 g}{\partial \theta \partial x}(x, \theta)\|$.

Often times (4.1) is more structured. For instance, in sum-of-squares (SOS) optimization, we have:

$$\mathcal{V} = \left\{V_Q(x) = m(x)^\mathsf{T} Q m(x) : Q \in \mathbb{R}^{d \times d}, Q = Q^\mathsf{T} \succeq 0, \|Q\|_F \le B_Q\right\} , \tag{4.3}$$

where $m : \mathbb{R}^p \mapsto \mathbb{R}^d$ is a monomial feature map. Note that (4.3) is an instance of (4.1) with $k = d(d+1)/2$. Hence Theorem 4.2 implies a bound of the form $\mathsf{err}(\hat{V}_n) \le \tilde{O}(d^2/n)$. However, we can actually use the matrix structure of (4.3) to sharpen the bound to $\mathsf{err}(\hat{V}_n) \le \tilde{O}(d/n)$ by a more careful estimate of $\mathcal{R}_n(\mathcal{V})$ using the dual Sudakov inequality.

Theorem 4.3. *Under Assumption 4.1, if problem (3.2) over the parametric linear function class (4.3) is feasible, then any solution $\hat{V}_n$ satisfies with probability at least $1 - \delta$ over $\xi_1, ..., \xi_n$:*

$$\mathsf{err}(\hat{V}_n) \le O(1)\left(B_Q^2 (B_m^2 + B_{Dm} B_m)^2 L_h^2 \frac{d \log^2 d \log^3 n}{\gamma^2 n} + \frac{\log(\log(B_h/\gamma)/\delta)}{n}\right) . \tag{4.4}$$

Here, $B_m := \sup_{x \in S} \|m(x)\|$ and $B_{Dm} := \sup_{x \in S} \|\frac{\partial m}{\partial x}(x)\|$.

4.2 Reproducing kernel Hilbert space function classes

We now consider the following non-parametric function class:

$$\mathcal{V} = \left\{ V_\alpha(\cdot) = \int_\Theta \alpha(\theta)\phi(\cdot;\theta)\, d\theta : \|V_\alpha\|_\nu := \sup_{\theta\in\Theta} \left|\frac{\alpha(\theta)}{\nu(\theta)}\right| \le B_\alpha \right\} . \tag{4.5}$$

Here, $\phi(\cdot;\theta)$ is a nonlinear function and ν is a probability distribution over Θ. This function class is a subset of the reproducing kernel Hilbert space (RKHS) defined by the kernel $k(x, y) = \int_\Theta \phi(x;\theta)\phi(y;\theta)\nu(\theta)\, d\theta$, and is dense in the RKHS as $B_\alpha \to \infty$ [35]. We further assume that $\phi(x;\theta)$ is of the form $\phi(x;\theta) = \phi(\langle x, w\rangle + b)$ with ϕ differentiable and $\theta = (w, b)$. RKHSs of this type often arise naturally. For instance, Bochner's theorem [36] states that every translation invariant kernel can be expressed in this form.

Theorem 4.4. *Suppose that $|\phi| \le 1$, ϕ is L_ϕ-Lipschitz, ϕ is differentiable, ϕ' is $L_{\phi'}$-Lipschitz, and that $B_\theta := \sup_{\theta\in\Theta}\|\theta\|$ is finite. Under Assumption 4.1, if problem (3.2) over the non-parametric class (4.5) is feasible, then any solution $\hat{V}_n$ satisfies with probability at least $1-\delta$ over $\xi_1, ..., \xi_n$:*

$$\mathsf{err}(\hat{V}_n) \le O(1)\left(B_\alpha^2(1 + B_\theta L_\phi)^2 L_h^2 \frac{\kappa \log^3 n}{\gamma^2 n} + \frac{\log(\log(1 + B_h/\gamma)/\delta)}{n} + 1/n^2 \right) ,$$

where $\kappa := \frac{B_\alpha^2 L_h^2}{\gamma^2}\left((1 + B_\theta L_\phi)^2 \log n + B_\theta^2(B_S + 1)^2(L_\phi + B_\theta L_{\phi'})^2 p\right)$.

5 Global Stability Results

In this section, we show how the bounds from Section 4 can be translated into global results for the learned certificate functions. To facilitate our analysis, we assume the dynamics is incrementally stable. Incremental stability is implied by contraction, but is stronger than Lyapunov stability. Before stating the assumption, we say that $\beta(s, t)$ is a class $\mathcal{KL}$ function if for every t the map $s \mapsto \beta(s, t)$ is a class $\mathcal{K}$ function and for every s the map $t \mapsto \beta(s, t)$ is continuous and non-increasing.

Assumption 5.1 (Incremental stability, c.f. Hanson and Raginsky [37])**.** *There exists a class $\mathcal{KL}$ function β such that for all $\xi_1, \xi_2 \in X$, $\|\varphi_t(\xi_1) - \varphi_t(\xi_2)\| \le \beta(\|\xi_1 - \xi_2\|, t)$ for all $t \in T$.*

With Assumption 5.1 in hand, we are ready to state a result regarding learned Lyapunov functions. For what follows, let $\mathbb{B}_2^p(r)$ denote the closed ℓ_2-ball in $\mathbb{R}^p$ of radius r, $\mathbb{S}^{p-1}$ denote the sphere in $\mathbb{R}^p$, and $\mu_{\mathsf{Leb}}(\cdot)$ denote the Lebesgue measure on $\mathbb{R}^p$.

Theorem 5.1. *Suppose the system satisfies Assumption 5.1, and suppose the set X is full-dimensional and compact. Define the set $S := \cup_{t\in T}\varphi_t(X)$. Let $V : S \mapsto \mathbb{R}$ be a twice-differentiable positive definite function satisfying $V(x) \ge \mu\|x\|^2$ for all $x \in S$. Define the violation set X_b as:*

$$X_b := \left\{ \xi \in X : \max_{t\in T}\langle \nabla V(\varphi_t(\xi)), f(\varphi_t(\xi))\rangle > \lambda V(\varphi_t(\xi)) \right\} . \tag{5.1}$$

Let ν denote the uniform probability measure on X and suppose that $\nu(X_b) \le \varepsilon$. Define the function $q(x) := \langle \nabla V(x), f(x)\rangle$, and denote the constants $B_{\nabla q} := \sup_{x\in S}\|\nabla q(x)\|$, $B_{\nabla V} := \sup_{x\in S}\|\nabla V(x)\|$. Let $r(\varepsilon) := \left(\frac{\varepsilon \mu_{\mathsf{Leb}}(X)}{\mu_{\mathsf{Leb}}(\mathbb{B}_2^p(1))}\right)^{1/p}$. Then for all $\eta \in (0, 1)$:

$$\langle \nabla V(x), f(x)\rangle \le -(1-\eta)\lambda V(x) \;\; \forall x \in \tilde{S} \setminus \mathbb{B}_2^p\left(0, \sqrt{\frac{\beta(r(\varepsilon), 0)}{\eta\mu}(B_{\nabla V} + \lambda^{-1}B_{\nabla q})}\right) . \tag{5.2}$$

Here, $\tilde{S} := \cup_{t\in T}\varphi_t(\tilde{X})$ with $\tilde{X} := \{\xi \in X : \mathbb{B}_2^p(\xi, r(\varepsilon)) \subset X\}$. Furthermore, for every $\xi \in X$, let $u_\xi(t)$ denote the solution to the differential equation:

$$\dot{u}_\xi = -\lambda u_\xi + (B_{\nabla q} + \lambda B_{\nabla V})\beta(r(\varepsilon), t) , \;\; u_\xi(0) = V(\xi) . \tag{5.3}$$

Then for every $\xi \in \tilde{X}$ and $t \in T$, the inequality $V(\varphi_t(\xi)) \le u_\xi(t)$ holds.

Theorem 5.1 states that the learned Lyapunov function V satisfies the Lie derivative decrease condition on all of $\tilde{S}$ except for a ball of radius $r_b \le O(\sqrt{\beta(r(\varepsilon), 0)})$ around the origin. Since $r(\varepsilon) \to 0$

as $\varepsilon \to 0$, Theorem 5.1 shows that the quality of our Lyapunov function increases as the measure of the violation set X_b decreases. Furthermore, we can apply the bounds in Section 4 to obtain an upper bound on the radius r_b of the ball as a function of the number of sample trajectories. For example, Theorem 4.2 states that $\nu(X_b) \leq O(k/n)$ if n random samples are drawn uniformly from X. For simplicity assume $X = \mathbb{B}_2^p(1)$ and $\beta(r, 0) \leq O(r)$, which implies $r_b \leq O((k/n)^{1/2p})$. Setting $r_b \leq \zeta$ and solving for n, we find $n \geq \Omega(k \cdot \zeta^{-2p})$.

Equation (5.3) yields bounds of the form $V(\varphi_t(\xi)) \leq V(\xi)e^{-\lambda t} + O(r(\varepsilon)h_\beta(t))$, where h_β depends on the specific form of β. For example if $\beta(s,t) \leq Me^{-\alpha t}s$ for some $\alpha > \lambda$, then $h_\beta(t) = e^{-\lambda t}$. On the other hand, if we have the slower rate $\beta(s,t) \leq Ms/(t+1)$, then $h_\beta(t) = t^2 e^{-\lambda t}$. We note that Theorem 5.1 is conceptually similar to the results from Liu et al. [38], but incremental stability assumption dramatically simplifies the proof and enables us to make the constants explicit.

We now state a similar result to Theorem 5.1 for metric learning. Let $\psi_t(\cdot)$ denote the induced flow on the prolongated system $g(x, \delta x) = (f(x), \frac{\partial f}{\partial x}(x)\delta x)$ and $\theta_t(\delta\xi; \xi)$ denote the second element of $\psi_t(\xi, \delta\xi)$. Further let $\zeta_p(r)$ be the Haar measure of a spherical cap in $\mathbb{S}^{p-1}$ with arc length r.

Theorem 5.2. *Fix an $\eta \in (0, 1)$. Suppose that $X \subseteq \mathbb{R}^p$ is full-dimensional and $p \geq 2$. Let $\dot{x} = f(x)$ be contracting in the metric $M_\star(x)$ with rate $\lambda > 0$. Assume that $mI \preceq M_\star(x) \preceq LI$. Let $V(x, \delta x) : \mathcal{T}S \mapsto \mathbb{R}$ be of the form $V(x, \delta x) = \delta x^\mathsf{T} M(x)\delta x$ for some positive definite matrix function $M(x)$ satisfying $M(x) \succeq \mu I$. Define the violation set Z_b as:*

$$Z_b := \left\{ (\xi, \delta\xi) \in X \times \mathbb{S}^{p-1} : \max_{t \in T} \langle \nabla V(\psi_t(\xi, \delta\xi)), g(\psi_t(\xi, \delta\xi)) \rangle > \lambda V(\psi_t(\xi, \delta\xi)) \right\} . \tag{5.4}$$

Suppose that $\nu(Z_b) \leq \varepsilon$, where ν is the uniform probability measure on $X \times \mathbb{S}^{p-1}$. Define $r(\varepsilon) := \sup\left\{ r > 0 : r^p \zeta_p(r) \leq \frac{\varepsilon \mu_{\mathsf{Leb}}(X)}{\mu_{\mathsf{Leb}}(\mathbb{B}_2^p(1))} \right\}$, and let $r_b := \sqrt{r(\varepsilon) B_H (B_{\nabla q} + \lambda B_{\nabla V})(L/m)^{3/2}/(\eta\lambda\mu)}$, where $B_H := \sup_{x \in S} \left\| \frac{\partial^2 f}{\partial x^2} \right\|$. Define the sets $\tilde{X}_t(r_b) := \{\xi \in \tilde{X} : \inf_{\delta\xi \in \mathbb{S}^{p-1}} \|\theta_t(\delta\xi; \xi)\| \geq r_b\}$ for $t \in T$, with $\tilde{X} := \{\xi \in X : \mathbb{B}_2^p(\xi, r(\varepsilon)) \subset X\}$. Then the system will be contracting in the metric $M(x)$ at the rate $(1-\eta)\lambda$ for every $x \in \tilde{S}(r_b) := \cup_{t \in T} \varphi_t(\tilde{X}_t(r_b))$.

6 Learning Certificates in Practice

We empirically study the generalization behavior of both learning Lyapunov functions and contraction metrics from trajectory data. We consider Lyapunov functions parameterized by $V(x) = x^\mathsf{T}(L(x)L(x)^\mathsf{T} + I)x$, where $L(x)$ is the (reshaped) value of a fully connected neural network with $\tanh$ activations of size $p \times h \times h \times p \cdot (2p)$, where p is the state-dimension of x and h is the hidden width. For metric learning, we study a convex formulation via SOS programming. Each matrix element $M_{ij}(x) = \langle w_{ij}, \phi(x) \rangle$ is given by a polynomial where w_{ij} are the learned weights and $\phi(x)$ is a feature map of monomials in the state vector. In our experiments, we numerically estimate the generalization error of a learned certificate using a test set. We compute an upper confidence bound (UCB) of the estimate using the Chernoff inequality with $\delta = 0.01$, as described by Langford [39].

Damped pendulum. We learn a Lyapunov function for the damped pendulum from 1000 training trajectories. Figure 2 (UL) shows the level sets of a typical learned Lyapunov function, where we also numerically rollout a dense set of trajectories starting from $\{x \in \mathbb{R}^2 : V_\theta(x) = 10\}$ to check set invariance. In Figure 2 (UR), we add a disturbance $\langle a, \kappa\phi(t) \rangle$ to the dynamics where $a \in \mathbb{R}^{10}$ is unknown and $\phi(t)$ are random sinusoids. We use an adaptive control law [2] based on the learned Lyapunov function to regulate $x \to 0$. We vary $\kappa \in \{1, 6, 10\}$ to study the robustness of the adaptation. Figure 2 (UR) shows that the learned Lyapunov function is able to provide enough information to robustly regulate the state even as the disturbance κ increases by a factor of 10, whereas the system without adaptation is driven far from the origin.

Stable standing for quadrupeds. We learn a discrete-time Lyapunov function for a quadruped robot [40] as it recovers from external forcing. We apply a random impulse force in the (x, y) plane at time $t = 0$ to the Minitaur quadruped environment in PyBullet [41], and use a hand-tuned PD controller to return the minitaur to a standing position. We train a discrete-time Lyapunov function in order to handle the discontinuities in the trajectories introduced by contact forces.

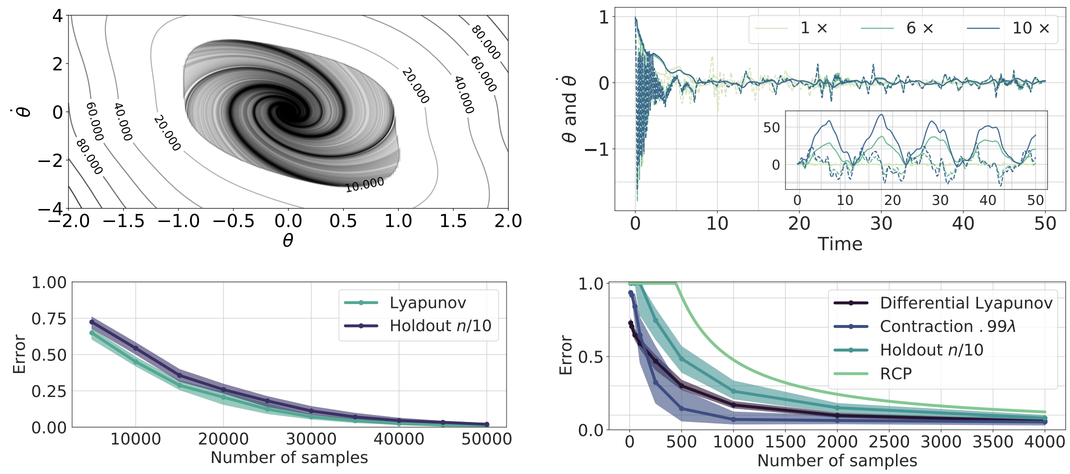

Figure 2: **(UL)** The level sets of a learned Lyapunov function on trajectories from a damped pendulum. Trajectories are initialized along the 10 level set and rolled out to demonstrate set invariance. **(UR)** Adaptive control of a damped pendulum driven by a random sinusoidal input. θ is shown in solid and $\dot{\theta}$ is shown dashed. Color indicates strength of the input. Main figure shows performance using adaptation via the learned Lyapunov function, while inset shows performance without adaptation. **(LL)** Generalization error of a learned Lyapunov function collected from trajectories of a standing minitaur impacted with a random impulse force and stabilized with a hand-tuned PD controller. **(LR)** Generalization error of a learned metric on a 6D gradient flow.

Figure 2 (LL) shows the result of this experiment. For the **Lyapunov** curve, the resulting model trained on n trajectories is then validated using a 10000 trajectory test set. The generalization error is the ratio of trajectories which violate the desired decrease condition *for any step* $k \in \{1, ..., 199\}$. We run 30 trials and plot the 10/50/90-th percentile of the generalization UCB. With $n = 50000$, the median generalization UCB is 1.11%. Since in practice a separate test set may not be available, we also compare to splitting the available training data into an actual training set of size $0.9n$ and a validation set of size $0.1n$. The model is trained on the actual training set, and a generalization UCB is calculated from the validation set. We run 30 trials of this setup and plot the 10/50/90-th percentile in the **Holdout** curve. After $n = 50000$, the median generalization UCB is 1.99%.

6-dimensional gradient system. Gradient flow has recently been explored in the context of Riemannian motion policies for robotics [42, 43], and converges for nonconvex losses with contracting dynamics [44]. We learn a metric for gradient flow on the nonconvex loss $\mathcal{L}(x) = \|x\|^2 + \sum_{i \neq j} x_i^2 x_j^2$ for $x \in \mathbb{R}^6$. Figure 2 (LR) shows the generalization error curves for the differential Lyapunov constraints. Because the SOS program for metric learning is convex, we can apply generalization bounds from randomized convex programming (RCP) [31]. We also plot the probability that the learned metric is a true contraction metric with rate $.99\lambda$ on the test set (probability with rate λ is low) and a generalization UCB obtained using a validation set. For each curve, we plot the 10/50/90-th percentile of the generalization UCB. As the number of samples increases, the error probability for differential Lyapunov constraints decreases, and the learned metric becomes a true metric with reduced rate with high probability. With $n = 4000$, the median generalization UCBs are 5.85%, 8.02%, and 5.35% for differential Lyapunov on the test set, differential Lyapunov on the validation set, and contraction with rate $.99\lambda$, respectively.

7 Conclusion

Our work shows that certificate functions can be efficiently learned from data, and raises many interesting questions for future work. Extending the results to handle both noisy state observations and process noise in the dynamics would allow for learning certificates in uncertain environments. Another interesting question is to establish bounds for joint learning of both the unknown dynamics and a certificate, which has shown to be effective in practice [15, 19]. Finally, lower bounds on the learning certificate problem would highlight the amount of conservatism introduced in our results.

References

[1] A. M. Lyapunov. *The general problem of the stability of motion (in Russian)*. PhD thesis, University of Kharkov, 1892.

[2] J.-J. Slotine and W. Li. *Applied Nonlinear Control*. 1991.

[3] F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[4] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference*, 2019.

[5] B. T. Lopez, J. E. Slotine, and J. P. How. Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, 5(3):1031–1036, 2021.

[6] W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[7] C. M. Kellett. Classical converse theorems in lyapunov's second method. *Discrete & Continuous Dynamical Systems - B*, 20(8):2333–2360, 2015.

[8] P. Giesl. Converse theorems on contraction metrics for an equilibrium. *Journal of Mathematical Analysis and Applications*, 424(2):1380–1403, 2015.

[9] L. Chen and J. Slotine. Notes on metrics in contraction analysis. *MIT Nonlinear Systems Laboratory Report*, (NSL-121101), Nov. 2012.

[10] E. D. Sontag. A 'universal' construction of artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13(2):117–123, 1989.

[11] M. Krstić and P. V. Kokotović. Control lyapunov functions for adaptive nonlinear stabilization. *Systems & Control Letters*, 26(1):17–23, 1995.

[12] A. A. Ahmadi, G. Hall, A. Papachristodoulou, J. Saunderson, and Y. Zheng. Improving efficiency and scalability of sum of squares optimization: Recent advances and limitations. In *2017 IEEE 56th Annual Conference on Decision and Control*, 2017.

[13] S. Gao, S. Kong, and E. M. Clarke. dreal: An smt solver for nonlinear theories over the reals. In *Automated Deduction – CADE-24*, 2013.

[14] S. M. Richards, F. Berkenkamp, and A. Krause. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning*, 2018.

[15] G. Manek and J. Z. Kolter. Learning stable deep dynamics models. In *Neural Information Processing Systems*, 2019.

[16] A. Taylor, A. Singletary, Y. Yue, and A. Ames. Learning for safety-critical control with control barrier functions. *arXiv:1912.10099*, 2019.

[17] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni. Learning control barrier functions from expert demonstrations. *arXiv:2004.03315*, 2020.

[18] W. Jin, Z. Wang, Z. Yang, and S. Mou. Neural certificates for safe control policies. *arXiv:2006.08465*, 2020.

[19] S. Singh, S. M. Richards, J.-J. E. Slotine, V. Sindhwani, and M. Pavone. Learning stabilizable nonlinear dynamics with contraction-based regularization. *arXiv:1907.13122*, 2019.

[20] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhwani. Learning stability certificates from data. *arXiv:2008.05952*, 2020.

[21] P. Giesl, B. Hamzi, M. Rasmussen, and K. Webster. Approximation of lyapunov functions from noisy data. *Journal of Computational Dynamics*, 7(1):57–81, 2020.

[22] J. Kenanian, A. Balkan, R. M. Jungers, and P. Tabuada. Data driven stability analysis of black-box switched linear systems. *Automatica*, 109:108533, 2019.

[23] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado. Learning lyapunov functions for piecewise affine systems with neural network controllers. *arXiv:2008.06546*, 2020.

[24] Y.-C. Chang, N. Roohi, and S. Gao. Neural lyapunov control. In *Neural Information Processing Systems*, 2019.

[25] H. Ravanbakhsh and S. Sankaranarayanan. Learning control lyapunov functions from counterexamples and demonstrations. *Autonomous Robots*, 43:275–307, 2019.

[26] V. Sindhwani, S. Tu, and S. M. Khansari-Zadeh. Learning contracting vector fields for stable imitation learning. *arXiv:1804.04878*, 2018.

[27] B. E. Khadir, J. Varley, and V. Sindhwani. Teleoperator imitation with continuous-time safety. In *Robotics: Science and Systems*, 2019.

[28] I. R. Manchester and J. E. Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 62(6):3046–3053, 2017.

[29] S. Singh, A. Majumdar, J. Slotine, and M. Pavone. Robust online motion planning via contraction theory and convex optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[30] M. C. Campi and S. Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

[31] G. C. Calafiore. Random convex programs. *SIAM J. Optim*, 20(6):3427–3464, 2010.

[32] F. Forni and R. Sepulchre. A differential lyapunov framework for contraction analysis. *IEEE Transactions on Automatic Control*, 59(3):614–628, 2013.

[33] N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low-noise and fast rates. In *Neural Information Processing Systems*, 2010.

[34] M. J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. 2019.

[35] A. Rahimi and B. Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, 2008.

[36] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Neural Information Processing Systems*, 2007.

[37] J. Hanson and M. Raginsky. Universal simulation of dynamical systems by recurrent neural nets. In *Learning for Dynamics and Control*, 2020.

[38] S. Liu, D. Liberzon, and V. Zharnitsky. Almost lyapunov functions for nonlinear systems. *Automatica*, 113:108758, 2020.

[39] J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, 2005.

[40] G. Kenneally, A. De, and D. E. Koditschek. Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*, 1(2):900–907, 2016.

[41] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2020.

[42] M. A. Rana, A. Li, D. Fox, B. Boots, F. Ramos, and N. Ratliff. Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems. *arXiv:2005.13143*, 2020.

[43] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox. Riemannian motion policies. *arXiv:1801.02854*, 2018.

[44] P. M. Wensing and J.-J. E. Slotine. Beyond convexity – contraction and global convergence of gradient descent. *PLoS One*, 15(8):e0236661, 2020.