Data-Driven Neural Certificate Synthesis

Luke Rickard [a] Alessandro Abate [b] Kostas Margellos [a]

[a] *Department of Engineering Science, University of Oxford*

[b] *Department of Computer Science, University of Oxford*

Abstract

We investigate the problem of verifying different properties of discrete time dynamical systems, namely, reachability, safety and reach-while-avoid. To achieve this, we adopt a data driven perspective and using past systems' trajectories as data, we aim at learning a specific function termed *certificate* for each property we wish to verify. The certificate construction problem is treated as a safety informed neural network training process, where we use a neural network to learn the parameterization of each certificate, while the loss function we seek to minimize is designed to encompass conditions on the certificate to be learned that encode the satisfaction of the associated property. Besides learning a certificate, we quantify probabilistically its generalization properties, namely, how likely it is for a certificate to be valid (and hence for the associated property to be satisfied) when it comes to a new system trajectory not included in the training data set. We view this problem under the realm of probably approximately correct (PAC) learning under the notion of compression, and use recent advancements of the so-called scenario approach to obtain scalable generalization bounds on the learned certificates. To achieve this, we design a novel algorithm that minimizes the loss function and hence constructs a certificate, and at the same time determines a quantity termed compression, which is instrumental in obtaining meaningful probabilistic guarantees. This process is novel per se and provides a constructive mechanism for compression set calculation, thus opening the road for its use to more general non-convex optimization problems. We verify the efficacy of our methodology on several numerical case studies, and compare it (both theoretically and numerically) with closely related results on data-driven property verification.

Key words: Verification of Dynamical Systems, Safety, Reachability, Statistical Learning, Scenario Approach.

1 Introduction

Dynamical systems offer a rich class of models for describing the behavior of systems [20]. It is often of importance that these systems meet certain properties, for example, stability, safety or reachability[15,22,26,32]. Verifying the satisfaction of these properties, also termed as specifications, is a challenging, but important, problem.

A common approach to verify properties of dynamical systems is through the use of *certificates* [2,30]. The goal is to determine a function over the system's state space which exhibits certain properties. A well-investigated example of such certificate is that of a Lyapunov function, used to verify that dynamics satisfy some stability property [23]. Here we consider constructing reachability, safety, and reach-while-avoid (RWA) certificates for discrete time systems. In general, constructing such certificates is a case dependent task and requires domain specific expertise. Restricting the class of certificates to polynomial functions, they can be obtained by solving a convex sum-of-squares problem [28,29].

In this work, rather than imposing assumptions on the class of certificate functions, we view the problem under a data driven lens. To this end, we use a *neural network* as a certificate template, which allows handling a general class of dynamical systems. Neural networks are well studied function approximators, with a broad array of applications, from image recognition [37] to reinforcement learning [6]. We employ neural networks with no assumptions on their structure to generate certificate parameterizations, and develop an algorithm for training these networks based on past system trajectories that play the role of data. We consider a safety informed neural network training process, employing a subgradient descent procedure to tackle the underlying non-convex optimization problem [5], and minimizing a loss function encompassing conditions on the certificate to be learned that encode the satisfaction of the associated property.

Email address: `rickard@robots.ox.ac.uk` (Luke Rickard).

This work was supported by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1.

Besides learning a certificate, we quantify probabilistically its generalization properties, namely, how likely it is for a certificate to be valid (and hence for the associated property to be satisfied) when it comes to a new system trajectory not included in the training data set. We view this problem under the realm of probably approximately correct (PAC) learning under the notion of compression [11,17,24], and use recent advancements of the so-called scenario approach [7,8,10,12,18] to obtain scalable generalization bounds on the learned certificates.

There has been a considerable amount of work towards data driven verification. One literature stream considers a direct property verification approach with no certificate construction. One such approach is to consider discretizing the state space [4,34], in order to construct a finite model which may be verified using statistical model checkers, or through statistical learning theoretic results [3,33]. Such techniques frequently suffer from the so-called "curse of dimensionality" which prohibits their application to systems of higher dimension. Therefore, abstraction-free techniques that involve certificate synthesis constitute an alternative approach. To this end, work has been conducted on synthesizing provably correct neural certificates [1,15], making use of SAT-modulo-theory solvers to verify that the synthesized networks meet the requirements. Alternatively, the techniques in [26,36] are the most closely related ones to our work, and consider synthesizing a neural certificate in a data-driven manner, using samples from the state space. However, the associated analysis accompanies the constructed certificates probabilistic guarantees of different nature, and is complementary to our approach. In particular, unlike our work, the generalization guarantees in [26] scale linearly in the dimension of the decision space (certificate parameterization) and exponentially in the uncertainty space (where samples are drawn from). This dependency hampers the application of these techniques to systems of higher dimensions.

Our main contributions can be summarized as follows:

(1) We develop a novel methodology for the synthesis of neural certificates to verify a wide class of properties, namely, reachability, safety and reach-while-avoid properties, of discrete-time dynamical systems. We accompany the constructed certificates with probabilistic guarantees on their generalization properties, namely, on how likely it is that the certificate fails to be valid for a new trajectory of our system. Our results complement the ones in [3] which are concerned with direct property verification and do not construct certificates. Our framework constitutes a first step towards control synthesis exploiting the constructed certificates.
(2) Our probabilistic guarantees are based on recent advancements in the so-called scenario approach theory, and are based on the notion of compression. This results in *a posteriori* bounds which, however, scale favorably with respect to the system dimension. We offer statements that are different, in some sense complementary to [26], while overcoming the scalability challenges of the bounds therein. We contrast our approach with [26] and discuss the relative merits of each, both theoretically (Section 5) and numerically (Section 6).
(3) As a byproduct of our certificate construction algorithm, we provide a novel mechanism to compute the quantity termed *compression*, which is instrumental in obtaining meaningful probabilistic guarantees. This process is novel per se and provides a constructive approach for the general compression set calculation in [27], opening the road for its use in general non-convex optimization problems.

The rest of the paper unfolds as follows: Section 2 introduces the different certificates under consideration, while Section 3 provides our probabilistic certificate guarantees. Section 4 provides a data driven algorithm that enjoys such guarantees, while Section 5 compares our work with the most closely results in the literature. Section 6 provides a numerical investigation, while Section 7 concludes the manuscript and provides some directions for future work.

Notation. We use $\{\xi_k\}_{k=0}^K$ to denote a sequence indexed by $k \in \{0, 1, \dots, K\}$. $V \models \psi$ defines condition satisfaction i.e., it evaluates to true if the quantity V on the left satisfies the condition ψ on the right, e.g., $x = 1 \models x > 0$ evaluates to true and $x = -1 \models x > 0$ evaluates to false. Using $\not\models$ represents the logical inverse of this (i.e., condition dissatisfaction). By $(\forall \xi \in \Xi) V \models \psi(\xi)$ we mean that some quantity V satisfies a condition ψ which, in turn, depends on some parameter ξ, for all $\xi \in \Xi$.

2 Certificates

We consider a family of certificates that allow us to make (probabilistic) statements on the behavior of a dynamical system, namely, how likely it is that it satisfies certain properties. To this end, we begin by defining a dynamical system, before considering the various certificates and the properties they verify.

2.1 Discrete-Time Dynamical Systems

We consider a bounded state space $X \subseteq \mathbb{R}^n$, and a dynamical system whose evolution starts at an initial state $x(0) \in X_I$, where $X_I \subseteq X$ denotes the set of possible initial conditions. From an initial state, we can uncover a finite trajectory, i.e., a sequence of states $\xi = \{x(k)\}_{k=0}^T$, where $T \in \mathbb{N}_+$, by following the dynamics

$$x(k+1) = f(x(k)). \tag{1}$$

We define $f\colon X \to \mathbb{R}^n$, but make no assumptions on its properties. The set of all possible trajectories $\Xi \subseteq X_I \times X^T$ is then the set of all trajectories starting from the initial set X_I. Note that this set-up considers only deterministic systems, but our methods are applicable to systems with stochastic dynamics - we discuss this in further detail in Section 3.1. We do not consider controller synthesis here, but recognize that our general form of dynamical system allows for verifying systems with controllers "in the loop": for instance, our techniques allow us to verify the behavior of a system with a predifined control law structure, such as Model Predictive Control [19].

In Section 3, we discuss using a finite set of trajectories in order to provide generalization guarantees for future trajectories. Our techniques only require a finite number of samples, and are *theoretically* not restricted on the properties of such samples (i.e. we may have a finite number of samples each with an infinitely long time horizon). However, we discuss in Section 4 how one can synthesize a certificate, and our algorithms are required to store, and perform some calculations, on these trajectories (which is not *practically* possible for T taken to infinity, or continuous time trajectories).

We are interested in verifying whether the behavior of a dynamical system satisfies certain properties. We use $\phi(\xi)$ to refer to a property of interest (defined concretely in the sequel), which is evaluated on a trajectory $\xi \in \Xi$. Specifically, we will define conditions ψ^s, that will have to be satisfied over some sub-domains of the state space, and $\psi^{\Delta}(\xi)$ that will define conditions that will have to be satisfied only at specific points along a trajectory $\xi \in \Xi$. The separate notations ψ^s and ψ^{Δ} are used to distinguish between trajectory-independent and -dependent conditions, respectively.

In order to verify the satisfaction of a property ϕ, we consider the problem of finding a *certificate* as follows.

Definition 1 (Property Verification & Certificates) *Given a property $\phi(\xi)$, and a function $V\colon \mathbb{R}^n \to \mathbb{R}$, let ψ^s and $\psi^{\Delta}(\xi)$ be conditions such that, if*

$$\exists V \colon V \models \psi^s \wedge (\forall \xi \in \Xi) V \models \psi^{\Delta}(\xi) \implies \phi(\xi), \forall \xi \in \Xi,$$

then the property ϕ is verified for all $\xi \in \Xi$. We then say that such a function V is a certificate *for the property encoded by ϕ.*

In words, the implication of Definition 1 is that if a certificate V satisfies the conditions in ψ^s, as well as the conditions in $\psi^{\Delta}(\xi)$, for all $\xi \in \Xi$, then the property $\phi(\xi)$ is satisfied for all trajectories $\xi \in \Xi$.

2.2 Certificates

We now provide a concrete definition for a number of these properties, and the associated certificates (and certificate conditions) that meet the format of Definition 1. We fix a time horizon $T < \infty$. We assume that V is continuous, so that when considering the supremum/infimum of V over X (already assumed to be bounded) or over some of its subsets, this is well-defined.

Property 1 (Reachability) *Consider* (1), *and let $X_G, X_I \subset X$ denote a goal and initial set, respectively. Assume further that X_G is compact and denote by ∂X_G its boundary. If, for all $\xi \in \Xi$,*

$$\begin{aligned} \phi_{\text{reach}}(\xi) :=& \exists k \in \{0, \dots, T\} \colon x(k) \in X_G, \\ & \wedge\ \forall j \in \{0, \dots, k\} \colon x(j) \in X \end{aligned} \tag{2}$$

holds, then we say that ϕ_{reach} encodes a reachability property. Ξ denotes the set of trajectories consistent with (1) *and with initial states contained within X_I.*

By the definition of ϕ_{reach} it follows that verifying that a system exhibits the reachability property is equivalent to verifying that all trajectories generated from the initial set enter the goal within at most T time steps, and stay within the domain X till then. In order to verify this property, we consider a certificate that must satisfy a number of conditions. These conditions are summarized next. Fix $\delta > -\inf_{x \in X_I} V(x) \geq 0$. We then have

$$V(x) \leq 0,\ \forall x \in X_I, \tag{3}$$
$$V(x) \geq -\delta,\ \forall x \in \partial X_G, \tag{4}$$
$$V(x) > -\delta,\ \forall x \in X \setminus X_G, \tag{5}$$
$$V(x) > 0,\ \forall x \in \mathbb{R}^N \setminus X, \tag{6}$$
$$\begin{aligned} & V(x(k+1)) - V(x(k)) \\ & < -\frac{1}{T}\Big(\sup_{x \in X_I} V(x) + \delta\Big),\ k = 0, \dots, k_G - 1, \end{aligned} \tag{7}$$

where $k_G := \min\{k \in \{0, \dots, T\} \colon V(x(k)) \leq -\delta\}$, or $k_G = T$, if there is no such k. Conditions (4)-(6) allow characterizing different parts of the state space by means of specific level sets of V. In particular, we require V to be non-positive within the initial set X_I (3) and positive outside the domain (6), while V should be no more negative than a pre-specified level $-\delta < 0$ in the rest of the domain X (5), and the sublevel set V less than $-\delta$ should be contained within the goal set X_G (4).

In the case that T tends to infinity (i.e. an infinite time horizon), the difference condition in (7) is reduced to a negativity requirement, as is standard in the literature [15]. Due to our finite time horizon, we require conditions (4)-(5) to provide a bound on the value of our function which we must reach within the time horizon.

The condition in (7) is a decrease condition (its right-hand side is negative due to the choice of δ), that implies

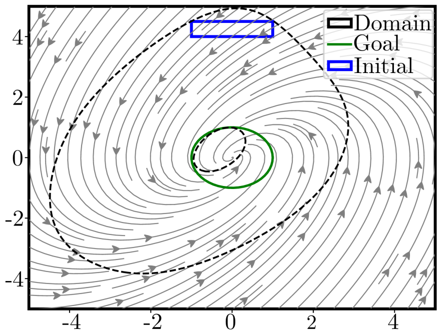


Fig. 1. Pictorial illustration of the level sets associated with the reach certificate for the system in (42).

V is decreasing along system trajectories till the first time the goal set is reached (by the definition of the time instance k_G). To gain some intuition on (7), see that if $k_G = T$, its recursive application leads to

$$V(x(T)) < V(x(0)) - T\frac{1}{T}\Big(\sup_{x\in X_I} V(x) + \delta\Big) \leq -\delta, \quad (8)$$

where the inequality holds since $V(x(0)) \leq \sup_{x\in X_I} V(x)$. Therefore, if the system starts within X_I, then it reaches the goal set (see (4)) in at most T steps.

A graphical representation of these conditions is provided in Figure 1. The inner sublevel set (with dashed line) is the set obtained when the certificate value is less than $-\delta$, whilst the outer one is the set obtained when the certificate is less than 0. The decrease condition then means that we never leave the larger sublevel set and must instead converge to the smaller sublevel set.

Now introduce ψ^s_{reach} to encode conditions (3)-(6), while $\psi^\Delta_{\text{reach}}(\xi)$ captures (7). Notice that the latter depends on ξ as it is enforced on consecutive states $x(k)$ and $x(k+1)$ along a trajectory.

With this in place, we can now define our first certificate.

Proposition 1 (Reachability Certificate) *A function $V : \mathbb{R}^n \to \mathbb{R}$ is a reachability certificate if*

$$V \models \psi^s_{\text{reach}} \wedge (\forall \xi \in \Xi) V \models \psi^\Delta_{\text{reach}}(\xi). \quad (9)$$

The proof is based on (8); provided formally in Appendix A. In words, Proposition 1 implies that a function V is a reachability certificate if it satisfies (3)-(6), and (7) for all trajectories generated by our dynamics.

We now consider a safety property, which is in some sense dual to reachability.

Property 2 (Safety) *Consider* (1)*, and let $X_I, X_U \subset X$ with $X_I \cap X_U = \varnothing$ denote an initial and an unsafe set, respectively. If for all $\xi \in \Xi$,*

$$\phi_{\text{safe}}(\xi) := \forall k \in \{0, \ldots, T\}, x(k) \notin X_U,$$

holds, then we say that ϕ_{safe} encodes a safety property. Ξ denotes the set of trajectories consistent with (1) *and with initial state contained within X_I.*

By the definition of ϕ_{safe}, it follows that verifying a system exhibits the safety property is equivalent to verifying all trajectories emanating from the initial set avoid the unsafe set for all time instances, until horizon T.

We now define the relevant criteria necessary for a certificate to verify this property.

$$V(x) \leq 0, \forall x \in X_I, \quad (10)$$
$$V(x) > 0, \forall x \in X_U, \quad (11)$$
$$V(x(k+1)) - V(x(k)) \quad (12)$$
$$< \frac{1}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big), \; k = 0, \ldots, T-1.$$

Notice that even if $\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x) > 0$, i.e., in the case where the last condition encodes an increase of V along the system trajectories, the system still avoids entering the unsafe set. In particular,

$$V(x(T)) < V(x(0)) + \Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big)$$
$$\leq \inf_{x\in X_U} V(x), \quad (13)$$

where the inequality is since $V(x(0)) \leq \sup_{x\in X_I} V(x)$. Therefore, by (11), the resulting inequality implies that even if the system starts at the least negative state within X_I, it will still remain safe.

We denote by ψ^s_{safe} the conjunction of (10) and (11), and by $\psi^\Delta_{\text{safe}}(\xi)$ the property in (12). We then have the following safety/barrier certificate

Proposition 2 (Safety/Barrier Certificate) *A function $V : \mathbb{R}^n \to \mathbb{R}$ is a safety/barrier certificate if*

$$V \models \psi^s_{\text{safe}} \wedge (\forall \xi \in \Xi) V \models \psi^\Delta_{\text{safe}}(\xi). \quad (14)$$

The proof can be found in Appendix A. Combining reachability and safety leads to richer properties. One of these is defined next.

Property 3 (Reach-While-Avoid (RWA)) *Consider* (1)*, and let $X_I, X_U, X_G \subset X$ with $(X_I \cup X_G) \cap X_U = \varnothing$ denote an initial set, an unsafe set, and a goal set, respectively. Assume further that X_G is compact and denote by ∂X_G its boundary. If for all $\xi \in \Xi$,*

$$\phi_{\text{RWA}}(\xi) := \forall k \in \{0, \ldots, T\}, x(k) \notin X_U \cup X^c$$
$$\wedge \; \exists k \in \{0, \ldots, T\}, x(k) \in X_G,$$

holds, then we say that ϕ_{RWA} encodes a RWA property. Ξ denotes the set of trajectories consistent with (1) *and with initial state contained within X_I.*

By the definition of ϕ_{RWA}, it follows that verifying that a system exhibits the RWA property is equivalent to verifying that all trajectories emanating from the initial set X_I avoid entering the unsafe set X_U (and the set complement of the domain X), and also eventually enter the goal set X_G.

Fix $\delta > 0$ such that $\delta > -\inf_{x\in X_I} V(x)$. The conditions necessary to verify this property are as follows:

$$V(x) \le 0,\ \forall x \in X_I, \tag{15}$$
$$V(x) > 0,\ \forall x \in X_U, \tag{16}$$
$$V(x) \ge -\delta,\ \forall x \in \partial X_G, \tag{17}$$
$$V(x) > -\delta,\ \forall x \in X \setminus X_G, \tag{18}$$
$$V(x(k+1)) - V(x(k)) \tag{19}$$
$$< -\frac{1}{T}\left(\sup_{x\in X_I} V(x) + \delta\right),\ k = 0,\ldots,k_G - 1,$$
$$V(x(k+1)) - V(x(k)) \tag{20}$$
$$< \frac{1}{T}\left(\inf_{x\in X_U} V(x) + \delta\right),\ k = k_G,\ldots,T - 1,$$

where recall that k_G denotes the first time the system trajectory will "hit" the $(-\delta)$-level set of V, which is associated with the goal set. We use ψ^s_{RWA} to denote the conjunction of (15)-(18), and $\psi^\Delta_{\text{RWA}}(\xi)$ for (19) and (20).

These conditions ensure that our initial and unsafe sets (including outside the domain) are separated by a zero-level set of the function V, and that there is a minimum inside the goal set. The difference conditions then ensure that we decrease from the initial set (and hence reach the goal set), and afterward do not increase so much that we enter the unsafe set.

Proposition 3 (RWA Certificate) *A function $V\colon \mathbb{R}^n \to \mathbb{R}$ is a RWA certificate if*

$$V \models \psi^s_{\text{RWA}} \wedge (\forall \xi \in \Xi) V \models \psi^\Delta_{\text{RWA}}(\xi). \tag{21}$$

The proof can be found in Appendix A. We provide a graphical representation of the properties in Figure 2.

To synthesize one of these deterministic certificates, we require complete knowledge of the behavior f of the dynamical system, to allow us to reason about the space of trajectories Ξ. This may be impractical, and we therefore consider learning a certificate in a data-driven manner.

3 Data-Driven Certificates

In some cases, one may have access to the underlying dynamics f, in which case it is possible to directly find a certificate that satisfies the relevant criteria ψ^s, ψ^Δ. However, in many real-world scenarios, access to the true dynamics requires a complete model of the physics of the system, and may not always be possible. Instead, we take a data-driven approach to synthesize a certificate based only on available trajectories/signatures of that system.

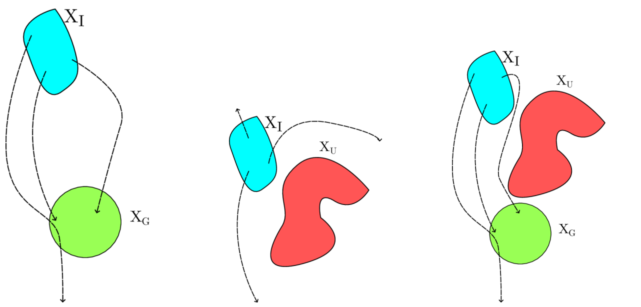
We denote by $(X_I, \mathcal{F}, \mathbb{P})$ a probability space, where $\mathcal{F}$ is a σ-algebra and $\mathbb{P}\colon \mathcal{F} \to [0,1]$ is a probability measure on our set of initial states X_I. Then, the initial state of our system is randomly distributed according to $\mathbb{P}$.

To obtain our sample set, we consider N initial conditions, according to probability distribution $\mathbb{P}$, namely $\{x^i(0)\}_{i=1}^N \sim \mathbb{P}^N$, where we assume that all samples are independent and identically distributed (i.i.d.). Initializing the dynamics from each of these initial states, we unravel a set of trajectories $\{\xi^i\}_{i=1}^N$. Since there is no stochasticity in the dynamics, we can equivalently say that trajectories (generated from the random initial conditions) are distributed according to the same probabilistic law; hence, with a slight abuse of notation, we write $\xi \sim \mathbb{P}$. In the case of a stochastic dynamical system, the vector field would depend on some additional disturbance vector; our subsequent analysis will remain valid with $\mathbb{P}$ being replaced by the probability distribution that captures both the randomness of the initial state and the distribution of the disturbance. We impose the following assumption.

Assumption 1 (Non-concentrated Mass) *Assume that $\mathbb{P}\{\xi\} = 0$, for any $\xi \in \Xi$.*

3.1 Problem Statement

Since we are now dealing with a sample-based problem, we will be constructing probabilistic certificates and hence probabilistic guarantees on the satisfaction of a given property. We will present our results for a generic property $\phi \in \{\phi_{\text{reach}}, \phi_{\text{safe}}, \phi_{\text{RWA}}\}$ and associated certificate conditions ψ^s, ψ^Δ.



(a) Reachability (b) Safety (c) RWA

Fig. 2. Pictorial illustration of (a) reachability, (b) safety, and (c) RWA properties, respectively. Black lines illustrate sample trajectories that satisfy the associated properties.

Denote by V_N a certificate of property ϕ, we introduce the subscript N to emphasize that this certificate is constructed on the basis of sampled trajectories $\{\xi^i\}_{i=1}^N$.

Problem 1 (Probabilistic Property Guarantee) *Consider N sampled trajectories, and fix a confidence level $\beta \in (0,1)$. We seek a property violation level $\epsilon \in (0,1)$ such that*

$$\begin{aligned}&\mathbb{P}^N\big\{\{\xi^i\}_{i=1}^N \in \Xi^N :\\ &\qquad \mathbb{P}\{\xi \in \Xi \colon V_N \not\models \psi^s \wedge \psi^\Delta(\xi)\} \leq \epsilon\big\} \geq 1-\beta. \end{aligned} \tag{22}$$

Addressing this problem allows us to provide guarantees even if part of the initial set does not satisfy our specification. Our statement is in the realm of probably approximately correct (PAC) learning: the probability of sampling a new trajectory $\xi \sim \mathbb{P}$ failing to satisfy our certificate condition is itself a random quantity depending on the samples $\{\xi^i\}_{i=1}^N$, and encompasses the generalization properties of a learned certificate V_N. It is thus distributed according to the joint probability measure $\mathbb{P}^N$, hence our results hold with some confidence $(1-\beta)$.

Providing a solution to Problem 1 is equivalent to determining an $\epsilon \in (0,1)$, such that with confidence at least $1-\beta$, the probability that V_N does not satisfy the condition $\psi^s \wedge \psi^\Delta(\xi)$ for another sampled trajectory $\xi \in \Xi$ is at most equal to that ϵ. As such, with a certain confidence, a certificate V_N *trained* on the basis of N sampled trajectories, will remain a valid certificate with probability at least $1-\epsilon$. Therefore, we can argue that V_N is a *probabilistic* certificate.

3.2 Probabilistic Guarantees

We now provide a solution to Problem 1. To this end, we refer to a mapping $\mathcal{A}$ such that $V_N = \mathcal{A}(\{\xi^i\}_{i=1}^N)$ as an algorithm that, based on N samples, returns a certificate V_N. Our main result will apply to a generic algorithm as long this exhibits certain properties that will be outlined as assumptions below. In Section 4 we provide a specific synthesis procedure through which $\mathcal{A}$ (and hence the certificate V_N) can be constructed, and show that this algorithm satisfies the considered properties.

The following definition constitutes the backbone of our analysis. The notion introduced below appears with different terms in the literature; we adopt the terminology introduced in [11,24] (adapted to our purposes) to align with the statistical learning literature.

Definition 2 (Compression Set) *Fix any $\{\xi^i\}_{i=1}^N$, and let $\mathcal{C}_N \subseteq \{\xi^i\}_{i=1}^N$ be a subset of the samples with cardinality $C_N = |\mathcal{C}_N| \leq N$. Define $V_{C_N} = \mathcal{A}(\mathcal{C}_N)$. We say that $\mathcal{C}_N$ is a compression of $\{\xi^i\}_{i=1}^N$ for algorithm $\mathcal{A}$, if*

$$V_{C_N} = \mathcal{A}(\mathcal{C}_N) = \mathcal{A}(\{\xi^i\}_{i=1}^N) = V_N. \tag{23}$$

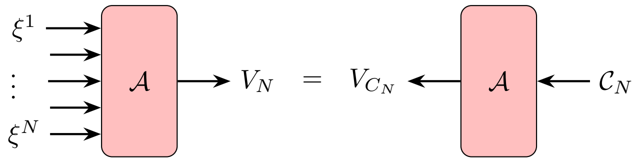


Fig. 3. Pictorial illustration of the compression set notion of Definition 2.

Notice the slight abuse of notation, as the argument of $\mathcal{A}$ might be a set of different cardinality; in the following, its domain will always be clear from the context.

Figure 3 illustrates Definition 2 pictorially. It should be noted that compression set cardinalities may be bounded *a priori* [8], that is, without knowledge of the sample-set, or obtained *a posteriori*, and hence depending on the given set $\{\xi^i\}_{i=1}^N$ [10]. Trivially, we could take a compression set as the entire sample set, resulting in a trivial risk upper bound of 1. However, it is of benefit to determine a compression set with small (ideally minimal) cardinality, as the smaller C_N is, the smaller risk we can guarantee. In this paper we are particularly interested in a posteriori results, since we solve a non-convex problem we cannot in general provide a non-trivial bound to the cardinality of the compression set a priori [12]. Therefore, we introduce the subscript N in our notation for $\mathcal{C}_N$ (set) and C_N (corr. cardinality), respectively.

The theorem below provides probabilistic guarantees that are valid irrespective of the cardinality of the underlying compression set. However, the quality of these bounds depends significantly on this cardinality, resulting in progressively sharper probabilistic bounds as the compression set cardinality decreases. In Algorithm 2 we propose a mechanism to obtain non-trivial compression sets, while avoiding computationally expensive procedures. This can be thought of as a by-product of the certificate synthesis procedure of Section 4, and constitutes per se a novel contribution of this work.

For the guarantees we introduce in the sequel to hold, that the algorithm $\mathcal{A}$ must satisfy the following properties (adapted from [11]; note that in [11] a non-concentrated mass property is also imposed, which here appears separately as Assumption 1). We later demonstrate that our proposed algorithm satisfies these.

Assumption 2 (Properties of $\mathcal{A}$) *Assume that algorithm $\mathcal{A}$ exhibits the following properties:*

(1) Preference: *For any pair of multisets $\mathcal{C}_1$ and $\mathcal{C}_2$ of elements of $\{\xi\}_{i=1}^N$, with $\mathcal{C}_1 \subseteq \mathcal{C}_2$, if $\mathcal{C}_1$ does not constitute a compression set of $\mathcal{C}_2$ for algorithm $\mathcal{A}$, then $\mathcal{C}_1$ will not constitute a compression set of $\mathcal{C}_2 \cup \{\xi\}$ for any $\xi \in \Xi$.*
(2) Non-associativity: *Let $\{\xi_i\}_{i=1}^{N+\bar{N}}$ for some $\bar{N} \geq 1$. If $\mathcal{C}$ constitutes a compression set of $\{\xi_i\}_{i=1}^N \cup \{\xi\}$ for all $\xi \in \{\xi_i\}_{i=N+1}^{N+\bar{N}}$ for algorithm $\mathcal{A}$, then $\mathcal{C}$ constitutes a compression set of $\{\xi_i\}_{i=1}^{N+\bar{N}}$ (up to a measure-zero set).*

We are now able to state the main result of this section.

Theorem 1 (Probabilistic Guarantees) *Consider any algorithm $\mathcal{A}$ such that $V_N = \mathcal{A}(\{\xi^i\}_{i=1}^N)$, and suppose it satisfies Assumption 2. Fix $\beta \in (0,1)$, and for $k < N$, let let $\varepsilon(k,\beta,N)$ be the (unique) solution to the polynomial equation in the interval $[k/N, 1]$*

$$\frac{\beta}{2N}\sum_{m=k}^{N-1}\frac{\binom{m}{k}}{\binom{N}{k}}(1-\varepsilon)^{m-N} + \frac{\beta}{6N}\sum_{m=N+1}^{4N}\frac{\binom{m}{k}}{\binom{N}{k}}(1-\varepsilon)^{m-N} = 1, \tag{24}$$

while for $k = N$ let $\varepsilon(N,\beta,N) = 1$. We then have that

$$\mathbb{P}^N\big\{\{\xi_i\}_{i=1}^N \in \Xi^N : \mathbb{P}\{\xi \in \Xi\colon V_N \not\models \psi^s \wedge \psi^\Delta(\xi))\} \le \varepsilon(\mathcal{C}_N,\beta,N)\big\} \ge 1-\beta. \tag{25}$$

Proof Fix $\beta \in (0,1)$, and for each $\{\xi\}_{i=1}^N$ let $\mathcal{C}_N$ be a compression set for algorithm $\mathcal{A}$. Moreover, note that letting $V_N = \mathcal{A}(\{\xi^i\}_{i=1}^N)$ we construct a mapping from samples $\{\xi\}_{i=1}^N$ to a decision, namely, V_N, while we impose as an assumption that this mapping satisfies the conditions of Assumption 2.

This framework directly fits into the setting of [11, Theorem 7], which implies that with confidence at least $1-\beta$, the probability that for a new $\xi \in \Xi$ the compression set changes, is at most $\epsilon(\mathcal{C}_N,\beta,N)$, i.e.,

$$\mathbb{P}\{\xi \in \Xi\colon \mathcal{C}_N^+ \neq \mathcal{C}_N\} \le \varepsilon(\mathcal{C}_N,\beta,N), \tag{26}$$

where $\mathcal{C}_N^+$ denotes a compression set for algorithm $\mathcal{A}$ when fed with $\{\xi\}_{i=1}^N \cup \{\xi\}$. However, we then have that

$$\begin{aligned}\{\xi \in \Xi\colon V_N \not\models \psi^s \wedge \psi^\Delta(\xi))\} &\subseteq \{\xi \in \Xi\colon V_N \neq \mathcal{A}(\{\xi\}_{i=1}^N \cup \{\xi\})\}\\ &= \{\xi \in \Xi\colon \mathcal{A}(\mathcal{C}_N) \neq \mathcal{A}(\mathcal{C}_N^+)\}\\ &\subseteq \{\xi \in \Xi\colon \mathcal{C}_N^+ \neq \mathcal{C}_N\},\end{aligned} \tag{27}$$

where the first inclusion is since for any $\xi \in \Xi$ for which V_N no longer satisfies the certificate condition ($\psi^s \wedge \psi^\Delta(\xi)$), we must have that the certificate changes, i.e., $\mathcal{A}(\{\xi\}_{i=1}^N \cup \{\xi\})$ (the output/certificate of our algorithm when fed with one more sample) is different from V_N. Notice that the opposite statement does not always hold, having a different certificate does not necessarily mean the old one violates an existing condition for a new $\xi \in \Xi$.

The following equality holds as $V_N = \mathcal{A}(\mathcal{C}_N)$, $\mathcal{A}(\{\xi\}_{i=1}^N \cup \{\xi\}) = A(\mathcal{C}_N^+)$, by definition of a compression set.

Finally, the last inclusion stands since any $\xi \in \Xi$ for which $\mathcal{A}(\mathcal{C}_N) \neq \mathcal{A}(\mathcal{C}_N^+)$, should be such that $C_N^+ \neq C_N$. The opposite direction does not always hold, as if $\mathcal{C}_N^+ \supset \mathcal{C}_N$ then we get another compression set of higher cardinality, and hence we may still have $\mathcal{A}(\mathcal{C}_N) = \mathcal{A}(\mathcal{C}_N^+)$. By (26) and (27), (25) follows, concluding the proof. □

By the implication in Definition 1, and under the same conditions as those in Theorem 1, we have the following corollary, which relates correctness of the certificate to bounds on the property under study.

Corollary 1 (Bounds on Property Satisfaction) *Suppose* (25) *is satisfied for the calculated risk $\epsilon(C_N,\beta,N)$, we can guarantee that*

$$\mathbb{P}^N\big\{\{\xi_i\}_{i=1}^N \in \Xi^N : \mathbb{P}\{\xi \in \Xi\colon \neg\phi(\xi)\} \le \epsilon(C_N,\beta,N)\big\} \ge 1-\beta. \tag{28}$$

That is, if there exists a certificate V_N that satisfies (25) up to a level ϵ, then the property $\phi(\xi)$ is satisfied up to the same level (with some confidence). This confidence refers to the selection of sampled trajectories we observed.

The following remarks are in order.

(1) Notice that Theorem (1) involves evaluating $\varepsilon(k,\beta,N)$ at $k = C_N$, i.e., at the cardinality of the compression set. As such, with confidence at least $1-\beta$ with respect to the choice of the trajectories $\{\xi_i\}_{i=1}^N$, the probability that V_N is not a valid certificate when it comes to a new trajectory ξ, is at most $\varepsilon(C_N,\beta,N)$. Due to the dependency of ε on the samples (via C_N), the proposed probabilistic bound is *a posteriori* as it is adapted to the samples we "see". As a result, this is often less conservative compared to *a priori* counterparts.
(2) For cases where algorithm $\mathcal{A}$ takes the form of an optimization program that is convex with respect to the parameter vector, determining non-trivial bounds on the cardinality of compression sets is possible [8,24], as this is related to the notion of support constraints in convex analysis. However, determining compression sets of low cardinality (necessary for small risk bounds) becomes a non-trivial task if $\mathcal{A}$ involves a non-convex optimization program and/or is iterative (as Algorithm 1). In such cases, samples that give rise to inactive constraints may still belong to a compression set, as they may affect the optimal parameter implicitly. A direct way of determining the minimal compression set is to resolve the problem with every subset of the samples [10,12]. Computationally, however, this would be an intense procedure, often prohibitive due to its combinatorial nature [16].
(3) An alternative procedure could be to use sampled trajectories and check directly whether a property is satisfied for them (by checking the property definition, rather than using the associated certificate's conditions). This is a valid alternative but has the drawback of not

providing a certificate V_N, but simply provides an answer as far as the property satisfaction is concerned. This direction is pursued in [3]; we review this result and compare with our approach in Section 5.1. Note that having a certificate is interesting per se, and opens the road for control synthesis; which we aim to pursue in future work.

4 Certificate Synthesis

In this section, we propose a mechanism to synthesize a certificate from sampled trajectories, thus offering a constructive approach for algorithm $\mathcal{A}$ in Theorem 1.

4.1 Neural Networks

In order to learn a certificate from samples, we consider a neural network, a well-studied class of function approximators that generalize well to a given task.

Definition 3 (Neural Network) *We denote a neural network by an input layer $z_0 \in \mathbb{R}^n$ (same dimension with the system state vector), a number of hidden layers $z_1 \in \mathbb{R}^{h_1}, \dots, z_k \in \mathbb{R}^{h_k}$, and an output layer $z_{k+1} \in \mathbb{R}$. Each layer, except the input, has an associated set of weights and biases $W_i \in \mathbb{R}^{h_{i-1} \times h_i}, b_i \in \mathbb{R}^{h_i}$, as well as an activation function $\sigma_i \colon \mathbb{R} \to \mathbb{R}$.*

The layers are related by the following equations,

$$z_i = \sigma_i(W_i z_{i-1} + b_i),\ i = 1, \dots, k, \tag{29}$$

$$z_{k+1} = W_{k+1} z_k + b_{k+1}, \tag{30}$$

where the activation function is applied element-wise to its argument.

Such a neural network acts as a "template" for our certificate V_N. Denote all tunable neural network parameters by a vector θ. We then have that our certificate V_N depends on θ. For the results of this section, we simply write V_θ and drop the dependency on N to ease notation.

4.2 Certificate and Compression Set Computation

We provide an algorithm that seeks to optimize the neural network parameters so that it results in a certificate $V_{\theta^\star}$. To this end, for a $\xi \in \Xi$ and parameter vector θ, let

$$L(\theta, \xi) = l^\Delta(\theta, \xi) + l^s(\theta), \tag{31}$$

represent an associated loss function consisting of a sample-dependent loss l^Δ, and a sample-independent loss l^s. Without loss of generality, we assume that we can drive the sample-independent loss to be zero (see further discussions later). We impose the next mild assumption, needed to prove termination of our algorithm.

Assumption 3 (Minimizers' Existence) *For any $\{\xi\}_{i=1}^N$, and any non-empty $\mathcal{D} \subseteq \{\xi\}_{i=1}^N$, the set of minimizers of $\max_{\xi \in \mathcal{D}} L(\theta, \xi)$, is non-empty.*

We aim at approximating a minimizer $\theta^\star$ of $\max_{\xi \in \mathcal{D}} L(\theta, \xi)$ when $\mathcal{D} = \{\xi\}_{i=1}^N$, which exists due to Assumption 3. We can then use that minimizer to construct $V_{\theta^\star}$. To achieve this, we employ Algorithm 1.

Algorithm 1 takes as inputs some initial (arbitrary) parameter vector θ and a set of samples $\mathcal{D} \subseteq \{\xi\}_{i=1}^N$. First, in steps 6–7, we optimize for the sample-independent loss until this loss is non-positive, which serves as a form of warm starting. In step 9, the sample $\overline{\xi}_D$ that achieves the worst-case (maximum) loss among the samples in $\mathcal{D}$ is identified, while in step 10 a subgradient of the worst-case loss function $L(\theta, \overline{\xi}_D)$ is computed. Steps 11–12 perform similar computations but with the set $\mathcal{C}$ maintained throughout the algorithm, in place of $\mathcal{D}$. As such, the subgradient in step 12 is termed approximate, as the samples in $\mathcal{C}$ may not achieve the worst-case loss value. It is to be understood that if $\mathcal{C}$ is empty (as per initialization) steps 11-12 are not performed.

Steps 13–18 of Algorithm 1 involve taking a descent step. If the inner product in step 13 is non-positive (i.e., if the approximate subgradient "steers" against the exact subgradient) and the exact subgradient is non-zero, then we proceed to step 15 and follow the exact subgradient (with stepsize α) to explore the new direction (this can be thought of as an exploration step); otherwise, we move to step 18 and follow the direction of the approximate subgradient which in this case would point towards a similar direction with the exact one. This logic prevents us from unnecessarily appending to $\mathcal{C}$ more samples. If the exact subgradient is followed, we add the associated sample $\overline{\xi}_D$ to the set $\mathcal{C}$ (step 16). We then iterate till the loss value meets a given tolerance η (see steps 8 and 19).

We view Algorithm 1 as a specific choice for the mapping $\mathcal{A}$ introduced in Section 3 when fed with $\mathcal{D} = \{\xi_i\}_{i=1}^N$, and some initial choice for θ. It terminates returning an updated θ, and a set $\mathcal{C}$ which forms a compression set for this algorithm. These are formalized below.

Proposition 4 (Algorithm 1 Properties) *Consider Assumption 1, Assumption 3 and Algorithm 1 with $\mathcal{D} = \{\xi_i\}_{i=1}^N$ and a fixed (sample independent) initialization for the parameter θ. We then have:*

(1) Algorithm 1 terminates, returning a parameter vector $\theta^\star$ and a set $\mathcal{C}_N$.
(2) The set $\mathcal{C}_N$ with cardinality $C_N = |\mathcal{C}_N|$ forms a compression set for Algorithm 1.
(3) Algorithm 1 satisfies Assumption 2.

The proof can be found in Appendix A.

Algorithm 1. Certificate Synthesis and Compression Set Computation

1: **function** $\mathcal{A}(\theta, \mathcal{D})$
2: Set $k \leftarrow 1$ ▷ Initialize iteration index
3: Set $\mathcal{C} \leftarrow \varnothing$ ▷ Initialise compression set
4: Fix $L_1 < L_0$ with $|L_1 - L_0| > \eta$ ▷ η is any fixed tolerance
5: **while** $l^s(\theta) > 0$ **do** ▷ While sample-independent state loss is non-zero
6: $g \leftarrow \nabla_\theta l^s(\theta)$ ▷ Gradient of loss function
7: $\theta \leftarrow \theta - \alpha g$ ▷ Step in the direction of sample-independent gradient
8: **while** $|L_k - L_{k-1}| > \eta$ **do** ▷ Iterate until tolerance is met
9: $\bar{\xi}_\mathcal{D} \in \arg\max_{\xi \in \mathcal{D}} L(\theta, \xi)$ ▷ Find a sample with maximum loss from $\mathcal{D}$
10: $\bar{g}_\mathcal{D} \leftarrow \nabla_\theta L(\theta, \bar{\xi}_\mathcal{D})$ ▷ Subgradient of loss function for $\xi = \bar{\xi}_D$
11: $\bar{\xi}_\mathcal{C} \in \arg\max_{\xi \in \mathcal{C}} L(\theta, \xi)$ ▷ Find a sample with maximum loss from $\mathcal{C}$
12: $\bar{g}_\mathcal{C} \leftarrow \nabla_\theta L(\theta, \bar{\xi}_\mathcal{C})$ ▷ Approximate subgradient of loss function for $\xi = \bar{\xi}_\mathcal{C}$
13: **if** $\langle \bar{g}_D, \bar{g}_\mathcal{C} \rangle \leq 0$ **then**
14: **if** $\bar{g}_D \neq 0$ **then**
15: $\theta \leftarrow \theta - \alpha \bar{g}_\mathcal{D}$ ▷ Step in the direction of exact subgradient
16: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\bar{\xi}_D\}$ ▷ Update compression set with $\xi = \bar{\xi}_D$
17: **else**
18: $\theta \leftarrow \theta - \alpha \bar{g}_\mathcal{C}$ ▷ Step in the direction of approximate subgradient
19: $L_k \leftarrow \min\{L_{k-1}, \max_{\xi \in \mathcal{D}} L(\theta, \xi)\}$ ▷ Update "running" loss value
20: $k \leftarrow k + 1$ ▷ Update iteration index
21: **return** $\theta, \mathcal{C}$

Proposition 4 implies that we can construct a certificate $V_N = V_{\theta^\star}$, while the algorithm that returns this certificate satisfies Assumption 2 and admits a compression set $\mathcal{C}_N$ with cardinality C_N. As such, Algorithm 1 offers a constructive mechanism to synthesize a certificate, and can be accompanied by the probabilistic guarantees of Theorem 1. Moreover, Assumptions 1 & 3 under which Algorithm 1 exhibits these properties are rather mild.

Our way of computing a compression set serves as an efficient alternative to existing methodologies, as we construct it iteratively. At the same time the constructed compression set is non-trivial as we avoid adding uninformative samples to it, and only add one sample per iteration in the worst case, however, the one that maximizes the loss (see step 12). This algorithm could be thought of as a constructive procedure for the general methodology proposed recently in [27].

Proposition 4 shows that Algorithm 1 terminates and produces parameter iterates that yield a non-increasing sequence of loss functions. As such, the algorithm moves towards the direction of the optimum, but we have no guarantees that it indeed reaches some (local) optimum. We conjecture the approximate subgradient used in our algorithm constitutes a descent direction [14], and hence if the step size is chosen appropriately the algorithm should converge to a stationary point. Current work focuses on formalizing this claim.

In some cases, the parameter returned by Algorithm 1 may result in a value of the loss function that is considered as undesirable (and as a result the constructed certificate might be far from meeting the desired conditions). To achieve a lower loss, we make use of a sample-and-discarding procedure [9,35]. To this end, consider Algorithm 2. At each iteration of this algorithm, the compression set returned by Algorithm 1 (step 5) is appended to a "running" set $\widetilde{\mathcal{C}}$ (see step 6). We then discard all elements of the compression set from $\mathcal{D}$, and repeat the process till the worst case loss $\max_{\xi \in \mathcal{D}} L(\theta, \xi) \geq 0$ is sufficiently small and ideally zero. This implies that Algorithm 1 is invoked each time with fewer samples as its input, while the set $\widetilde{\mathcal{C}}$ progressively increases. The set $\widetilde{\mathcal{C}}$ is a compression set that includes all samples that lead to a worst case loss, plus all samples that are removed along the process of Algorithm 2. However, it has higher cardinality compared to the original compression set, implying that improving the loss comes at the price of an increased risk level ε as the cardinality of the compression set increases.

4.3 Choices of Loss Function

We now provide some choices of the loss function $L(\theta, \xi) = l^\Delta(V_\theta, \xi) + l^s(V_\theta)$ so that minimizing that function we obtain a parameter vector $\theta^\star$, and hence also a certificate $V_{\theta^\star}$, which satisfies the conditions of the property under consideration, namely, reachability, safety, or RWA. Note that when calculating subgradients to these functions, which as we will see below are non-convex, we effectively have the so-called Clarke subdifferential [14].

Algorithm 2. Compression Set Update with Discarding

Fix $\{\xi^i\}_{i=1}^N$
Set $\widetilde{\mathcal{C}} \leftarrow \varnothing$ ▷ Initialize compression set
Set $\mathcal{D} \leftarrow \{\xi^i\}_{i=1}^N$ ▷ Initialise "running" samples
while $\max_{\xi \in \mathcal{D}} L(\theta, \xi) > 0$ **do**
  $\theta, \mathcal{C} \leftarrow \mathcal{A}(\theta, \mathcal{D})$ ▷ Call Algorithm 1
  $\widetilde{\mathcal{C}} \leftarrow \widetilde{\mathcal{C}} \cup \mathcal{C}$ ▷ Update $\widetilde{\mathcal{C}}$
  $\mathcal{D} \leftarrow \mathcal{D} \setminus \widetilde{\mathcal{C}}$ ▷ Discard $\widetilde{\mathcal{C}}$ from $\mathcal{D}$
return $\theta, \widetilde{\mathcal{C}}$

We provide some expressions for l^s and l^Δ for the reachability property in Property 1. For the other properties, the loss functions can be defined in an analogous manner. To this end, we define

$$l^s(V_\theta) := \int_{X \setminus X_G} \max\{0, -\delta - V_\theta(x)\}\, \mathrm{d}x \tag{32}$$
$$+ \int_{X_I} \max\{0, V_\theta(x)\}\, \mathrm{d}x + \int_{\mathbb{R}^N \setminus X} \max\{0, -V_\theta(x)\}\, \mathrm{d}x.$$

Focusing on the first of these integrals, if $V(x) > -\delta$ then $\max\{0, -\delta - V_\theta(x)\} = 0$, i.e., no loss is incurred, implying satisfaction of (4), (5). Under a similar reasoning, the other integrals account for (3) and (6), respectively. Note that, for a sufficiently expressive neural network, we can find a certificate V which satisfies the state constraints and hence has a sample-independent loss of zero.

In practice, we replace integrals with a summation over points generated deterministically within the relevant domains. These points are generated densely enough across the domain of interest, and hence offer an accurate approximation. This generation may happen through gridding the relevant domain, or sampling according to a fixed synthetic distribution. For the last term, we only enforce the positivity condition on the border of the domain X. Thus, we take a deterministically generated discrete set of points on each domain $\mathcal{X}_{\overline{G}}$ for points in the domain but outside the goal region, $\mathcal{X}_I$ from the initial set, and $\mathcal{X}_\partial$ for the border of the domain X. Since these samples do not require access to the dynamics we consider them separate to the sample-set $\{\xi_i\}_{i=1}^N$, and references to the size of the sample set only refer to the trajectory samples (since these are the "costly" samples). Our practical loss function is then of the following form:

$$\hat{l^s}(V_\theta) := \frac{1}{|\mathcal{X}_{\overline{G}}|} \sum_{x \in \mathcal{X}_{\overline{G}}} \max\{0, -\delta - V_\theta(x)\} \tag{33}$$
$$+ \frac{1}{|\mathcal{X}_I|} \sum_{x \in \mathcal{X}_I} \max\{0, V_\theta(x)\} + \frac{1}{|\mathcal{X}_\partial|} \sum_{x \in \mathcal{X}_\partial} \max\{0, -V_\theta(x)\}.$$

We define l^Δ by

$$l^\Delta(V_\theta, \xi) := \max\Big\{0, \frac{1}{T}\Big(\sup_{x \in X_I} V_\theta(x) + \delta\Big) - \max_{k=0,\dots,k_G-1} \Big(V_\theta(x(k+1)) - V_\theta(x(k))\Big)\Big\}. \tag{34}$$

The value of l^Δ encodes a loss if the condition in (7) is violated. If both l^s and l^Δ evaluate to zero for all $\{\xi\}_{i=1}^N$, then we have that

$$l^s(V_\theta) + \max_{i=1,\dots,N} l^\Delta(V_\theta, \xi^i) = 0, \tag{35}$$

which by Certificate 1 implies that the constructed certificate V_θ is such that

$$V_\theta \models \psi^s_{\text{reach}} \wedge (i = 1, \dots, N) V_\theta \models \psi^\Delta_{\text{reach}}(\xi^i). \tag{36}$$

Analogous conclusions hold for all other certificates.

5 Comparison with Related Work

5.1 Direct Property Evaluation

As is known in the case of Lyapunov stability theory, the existence of a certificate is useful per se, and allows one to translate a property to a scalar function. As discussed in Corollary 1, a byproduct of this certificate synthesis is that they provide guarantees on the probability of property violation (see (28)). However, if one is not interested in the construction of a certificate and only in such guarantees, then Theorem 2 in [3] provides an alternative. We adapt this result in the proposition below.

Proposition 5 (Theorem 2 in [3]) *Fix $\beta \in (0,1)$, and for $r = 0, \dots, N-1$, determine $\varepsilon(r, \beta, N)$ such that*

$$\sum_{k=0}^{r} \binom{N}{i} \epsilon^k (1-\epsilon)^{N-k} = \frac{\beta}{N}, \tag{37}$$

while for $r = N$ let $\varepsilon(N, \beta, N) = 1$. Denote by R_N the number of samples in $\{\xi^i\}_{i=1}^N$ for which $\phi(\xi^i)$ is violated. We then have that

$$\mathbb{P}^N\big\{\{\xi^i\}_{i=1}^N \in \Xi^N : \mathbb{P}\{\xi \in \Xi \colon \neg\phi(\xi)\} \le \varepsilon(R_N, \beta, N)\big\} \ge 1 - \beta. \tag{38}$$

This involves a direct application of the sampling-and-discarding results in [9]. It is *a posteriori* as R_N can be determined only once the samples are observed. To this end, the term β/N appears in the right-hand side of (38) to account for the fact that, depending on the samples, up to N terms could appear in the summation. In this setting, we have a compression set which is the set of all

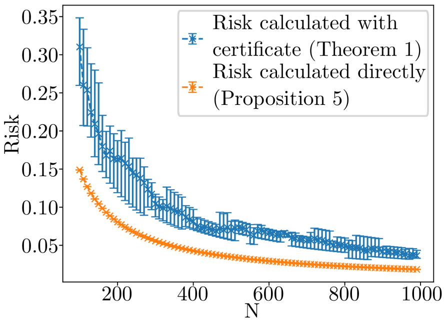


Fig. 4. Comparison of the bounds in Theorem 1 and Proposition 5 for direct property evaluation. Median values are shown with a cross, and ranges are indicated by error bars.

discarded samples, plus an additional one to support the solution after discarding. Since this additional sample is always present, we incorporate it in the formula in (37).

We remark that one could obtain different bounds through alternative statistical techniques, such as Hoeffding's inequality [21] or Chernoff's bound [13]. Since these bounds are of different nature, we do not pursue that avenue further here.

Proposition 5 offers an alternative to Theorem 1 to directly bound property violation. We compare the risk levels ε computed by each approach on our benchmark example in (42) under a safety specification; general conclusions are case dependent, as both bounds are *a posteriori*. For a fixed β, Figure 4 shows the resulting risk levels for varying N across 5 independently sampled sets of trajectories. The difference of the orange curve from the blue one can be interpreted as the price of certificate generation of Theorem 1. For sufficiently large N, this price is marginal. As the specification is deterministically safe, no discarding is performed for Proposition 5, hence a smooth curve without variability. For non-zero R_N we expect variability as R_N will be randomly distributed.

5.2 Certificate Synthesis as in [26]

The results in [26] constitute the most closely related ones with respect to our work. As no results on reachability and RWA problems were provided in [26], we limit our discussion to the safety property. As with our work, a sample-based construction is performed, where samples therein are pairs (state, next-state), as opposed to trajectories as in our work. However, the probabilistic bounds established in [26] are structurally different and of complementary nature to our work: next, we review the main result in [26], adapted to our notation.

Theorem 2 (Theorem 5.3 in [26]) *Consider (1), with initial and unsafe sets $X_I, X_U \subset X \subset \mathbb{R}^n$, respectively. Consider also N samples $\{x_i, f(x_i)\}_{i=1}^N$ from X, and assume that the loss function in (31) is Lipschitz continuous with constant $\mathcal{L}$. Consider then the problem*

$$\begin{aligned} \eta_N^\star \in &\underset{d=(\gamma,\lambda,c,\theta),\eta\in\mathbb{R}}{\arg\min} \quad \eta \\ \text{st.}\ & V_\theta(x) - \gamma \leq \eta,\ \forall x \in X_I \\ & V_\theta(x) - \lambda \geq -\eta,\ \forall x \in X_U \\ & \gamma + cT - \lambda - \mu \leq \eta,\ c \geq 0, \\ & V_\theta(f(x_i)) - V_\theta(x_i) - c \leq \eta,\ i = 1, \dots, N, \end{aligned} \tag{39}$$

where θ parameterizes V_θ, and all other decision variables are scalars leading to level sets of V_θ. Let $\kappa(\delta)$ be such that

$$\kappa(\delta) \leq \mathbb{P}\{\mathbb{B}_\delta(x)\}, \forall \delta \in \mathbb{R}_{\geq 0}, \forall x \in X, \tag{40}$$

where $\mathbb{B}_\delta(x) \subset X$ is a ball of radius δ, centered at x. Fix $\beta \in (0,1)$ and determine $\epsilon(|d|, \beta, N)$ from (37), with $r = d$ and by replacing the right hand-side with β. If $\eta_N^\star \leq \mathcal{L}\kappa^{-1}(\epsilon(|d|, \beta, N))$, we have that

$$\mathbb{P}^N\big\{\{\xi^i\}_{i=1}^N \in \Xi^N : \phi_{\text{safe}}(\xi),\ \forall \xi \in \Xi\big\} \geq 1 - \beta. \tag{41}$$

The following remarks are in order.

(1) The result in [26] is *a priori* (capitalizing on the developments of [25]), as opposed to the *a posteriori* assessments of our analysis that are in turn based on [11]. Moreover, [26] offers a guarantee that, with a certain confidence, the safety property is *always* satisfied. This is in contrast to Theorem 1 where we provide such guarantees in probability (up to a quantifiable risk level ε). However, these “always” guarantees come with potential challenges. In particular, the constraint in (40) involves the measure of a “ball” in the uncertainty space. The measure of this ball grows exponentially in the dimension of the uncertainty space (see also Remark 3.9 in [25]), while it depends linearly on the dimension of the decision space $|d|$ (see dependence of ε below (40)). This dependence in the results of [26] raises computational challenges to obtain useful bounds: we demonstrate this numerically in Section 6 employing one of the examples considered in [26]. On the contrary, Theorem 1 is independent of the dimension of these spaces and only depends on the cardinality of the compression set.
(2) The result in [26] requires inverting $\kappa(\delta)$, which may not have an analytical form in general. Moreover, it implicitly assumes some knowledge of the distribution to obtain κ, and of the Lipschitz constants of the system dynamics, which we do not require in our analysis.
(3) The results of [26] are also extended to continuous-time dynamical systems. This is also possible for our results; however, due to practical considerations, we then require knowledge of the Lipschitz constants not required in the discrete time setting. This discussion is not pursued further here.

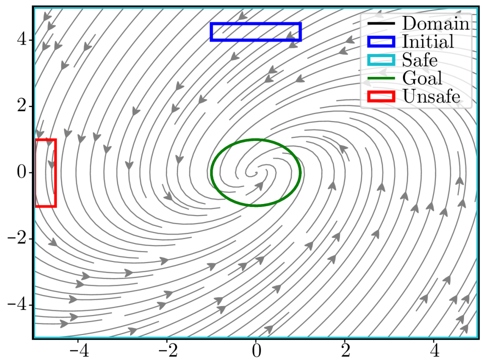


Fig. 5. Phase plane plot for the dynamical system of (42). The different sets shown are related to the sets that appear in the definitions of the reachability, safety and RWA property. For each case, only the relevant sets are considered.

6 Numerical Results

Numerical implementation was performed in Python and is available at github.com/lukearcus/fossil_scenario. Simulations were carried out on a server with 80 2.5 GHz CPUs and 125 GB of RAM. For all numerical simulations, we considered a confidence level of $\beta = 10^{-5}$, $N = 1000$ samples; our results are averaged across 5 independent repetitions, each with different multi-samples. By sample complexity, we refer to the number of *trajectory samples*, separate to the states used for the sample-independent loss since these samples can be obtained without accessing the system dynamics.

6.1 Benchmark Dynamical System

To demonstrate the efficacy of our techniques across all certificates presented, we use the following dynamical system as benchmark, with state vector $x(k) = (x_1(k), x_2(k)) \in \mathbb{R}^2$, namely,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) - \frac{T_\mathrm{d}}{2} x_2(k) \\ x_2(k) + \frac{T_\mathrm{d}}{2}(x_1(k) - x_2(k)) \end{bmatrix}, \tag{42}$$

where $T_\mathrm{d} = 0.1$ and we use time horizon $T = 100$ steps. The phase plane plot for these dynamics is in Figure 5 alongside different sets related to the definition of the reachability, safety and RWA properties are shown. For the reachability property we aim at verifying that trajectories reach the goal set X_G (circle centered at the origin), without leaving the domain X until then. The unsafe set is not relevant as far as this property is concerned For safety, we require that trajectories do not enter the unsafe region X_U. Finally, for the RWA property, we have the domain X and the unsafe set X_U.

Surface plots of the reachability, barrier and RWA certificate are shown in Figure 6, Figure 7 and Figure 8, respectively. The zero and $-\delta$-sublevel sets of these certificates are highlighted with dashed black lines. With reference to Figure 6 notice that the zero-sublevel set includes both the initial and the goal set, and no states outside the domain as expected. Similarly, in Figure 7 the zero-sublevel set of the barrier function does not pass through the unsafe set, while the zero-sublevel set of the RWA certificate does not pass through the unsafe set, and does not include states outside the domain.

The constructed certificates depend on N samples. By means of Algorithm 1 and Theorem 1, these certificates are associated with a theoretical risk bound ε (that bounds the probability that the certificate will not meet the conditions of the associated property when it comes to a new sample/trajectory). Table 1 shows this risk bound as computed via Theorem 1. We quantified empirically this property; namely, we generated additional samples and calculated the number of samples for which the computed certificate violated the associated certificate's conditions. This empirical certificate risk is denoted by $\hat{\varepsilon}$ is shown in the second column of Table 1. Note that, as expected, the empirical values are lower than the theoretical bounds.

The fourth column of Table 1 provides the risk bound ε that would be obtained for direct property violation statements (however, without allowing for certificate construction) as per Proposition 5, this always results in a risk of 0.01825 as no samples are discarded, since the system can be shown to be deterministically safe. Recall that the results in the first column of Table 1 bound (implicitly) the probability of property violation, as discussed in the second remark after the proof of Theorem 1.

6.2 Dynamical System of Higher Dimension

We now investigate a dynamical system of higher dimension with a state $x(k) \in \mathbb{R}^8$, governed by

$$\begin{aligned} x_i(k+1) &= x_i(k) + 0.1 x_{i+1}(k),\ i = 1 \dots 7, \\ x_8(k+1) &= x_8(k) - 0.1(576 x_1(k) + 2400 x_2(k) \\ &\quad + 4180 x_3(k) + 3980 x_4(k) + 2273 x_5(k) \\ &\quad + 800 x_6(k) + 170 x_7(k) + 20 x_8(k)). \end{aligned} \tag{43}$$

We define $X = [-2.2, 2.2]^8, X_I = [0.9, 1.1]^8, X_U = [-2.2, -1.8]^8$. Once again, the entire of the initial set can be shown to be safe, and so we aim to generate a guarantee as close to 0 as possible. We employ Algorithm 1 to generate a safety certificate. This required an average of 72 seconds, with a standard deviation of 4 seconds.

This certificate is computed much faster than those in Table 1, this is possible since the runtime of our algorithm is primarily constrained by how many samples need to be removed by Algorithm 2 in order to bring the loss to 0. This can be seen as a measure of how "hard" the problem is. In this example, it is likely that the sets

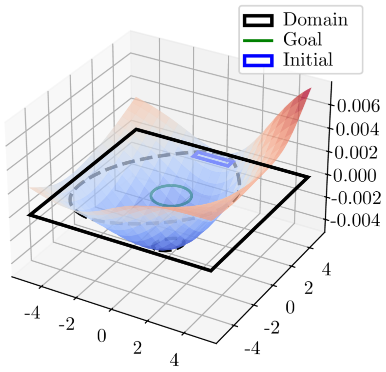


Fig. 6. Surface plot of the reachability certificate

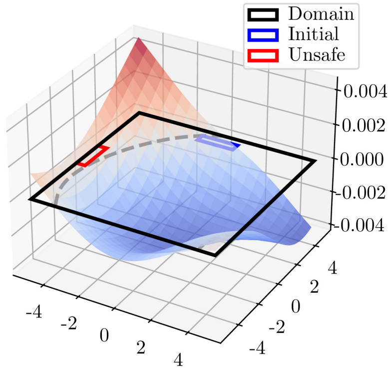


Fig. 7. Surface plot of the safety/barrier certificate.

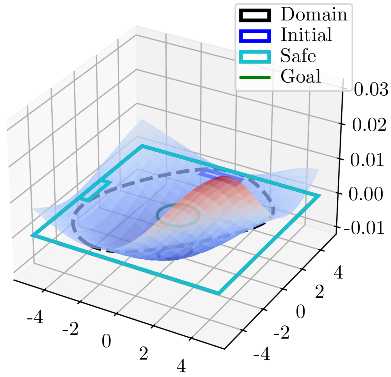


Fig. 8. Surface plot of the RWA certificate.

Table 1
Probabilistic guarantees for the system in (42). Standard deviations are shown in parantheses alongside means.

	Certificate Risk Bound ε in Theorem 1	Empirical Certificate Risk $\hat{\varepsilon}$	Algorithm 2 Computation Time (s)	Property Risk Bound ε in Prop. 5	Empirical Property Risk $\hat{\varepsilon}$	Direct Bound Computation Time (s)
Reach Certificate (Proposition 1)	0.02486 (0.00134)	0 (0)	4495 (865)	0.01825 (0)	0 (0)	10 (4)
Safety Certificate (Proposition 2)	0.03585 (0.00281)	0 (0)	1071 (141)	0.01825 (0)	0 (0)	47 (9)
RWA Certificate (Proposition 3)	0.03403 (0.00892)	0 (0)	1857 (434)	0.01825 (0)	0 (0)	29 (19)

are easy to separate whilst still maintaining the difference condition, whereas the system in the previous section required more computation since trajectories move towards the unsafe set before moving away.

Due to the higher-dimensional state space, this certificate is not illustrated pictorially. It is accompanied by a probabilistic certificate $\varepsilon = 0.02039$ (standard deviation 0) computed by means of Theorem 1. Using Proposition 5, we find a guarantee of 0.01825 (standard deviation 0), after 6 seconds (standard deviation 2s).

6.3 Partially Unsafe Systems

For the numerical experiments so far, all sampled trajectories satisfied the property of interest. We now consider the problem of safety certificate construction for the system in (42) with an enlarged unsafe region (see Figure 9). We refer to this system as partially unsafe, as some sampled trajectories enter the unsafe set. Unlike existing techniques which require either a deterministically safe system [15], or stochastic dynamics [31], we are still able to synthesize a probabilistic barrier certificate. The zero-sublevel set of the constructed safety certificate is shown by a dashed line in both Figures 9 and 10. Figure 10 provides a surface of the constructed certificate, and demonstrates that it separates the initial and the unsafe set. The computation time was 3631 seconds (standard deviation 302s).

For this certificate, we obtained a theoretical risk bound $\varepsilon = 0.27586$ (standard deviation 0.02355) by means of Theorem 1, and an empirical property risk of $\hat{\varepsilon} = 0.0144$ (standard deviation 0.0031). These guarantees are not tight; we could improve these by considering additional samples and performing the discarding procedure of Algorithm 2, however, this would lead to larger computation times. To prevent discarding too many trajectories, we only discard those in the compression set at each iteration, which is likely to be a smaller number.

Proposition 5 gives risk bound 0.04741 (standard deviation 0.00527) after 4 seconds (standard deviation 1s).

6.4 Comparison with [26]

We also compare our work with the one of [26], which has been reviewed in Section 5. To this end, we construct a safety certificate for a nonlinear, two-dimensional Jet engine model as considered in [26]. We first replicate the methodology of [26], and estimate the required Lipschitz constants using the technique in [38].

The methodology of [26] required 257149 samples and 521 seconds (standard deviation 9s) of computation time to compute a barrier certificate with confidence at least equal to 0.99. Using 1000 samples and 594 seconds of computation time (standard deviation 17s), we obtained $\varepsilon = 0.00099$ (standard deviation 0), i.e. we can bound

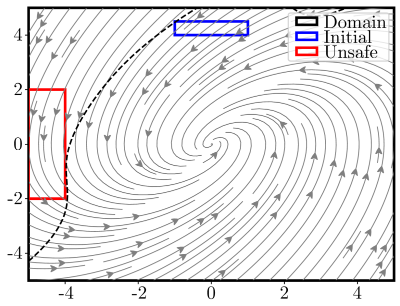


Fig. 9. Phase plane plot, initial and unsafe set for of partially unsafe system.

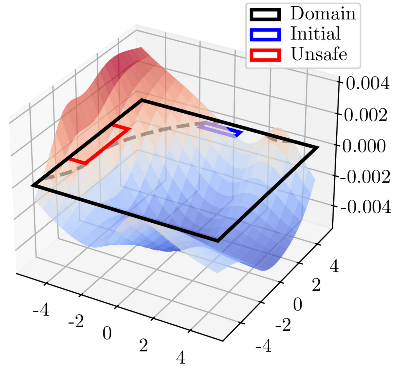


Fig. 10. Surface plot of the safety/barrier certificate for the partially unsafe system of Figure 9.

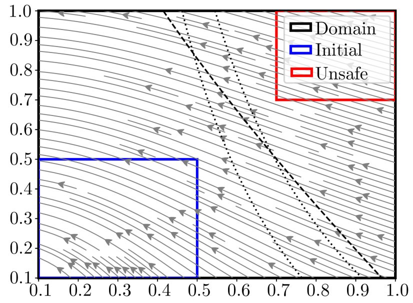


Fig. 11. Comparison with [26]. The zero-level set of the safety certificate of our approach is dashed; level sets that separate the initial and unsafe sets (i.e. γ- and λ- level sets) from [26] are dotted.

safety with a risk of 0.1%, for the same confidence. It can thus be observed that the numerical computation savings (in terms of number of samples – this might be an expensive task – and computation time) are significant. Figure 11 illustrates a phase plane plot and the initial and unsafe sets for this problem. The dotted lines correspond to the sublevel sets constructed in [26] (one lower bounding the unsafe set, the other upper bounding the initial set). The dashed line depicts the zero-sublevel set of the certificate constructed by our approach.

We also performed a comparison on the following four-dimensional system taken from [15].

$$\begin{aligned}
x_1(k+1) &= x_1(k) + \frac{x_1(k)x_2(k)}{5} - \frac{x_3(k)x_4(k)}{2}, \\
x_2(k+1) &= \cos(x_4(k)), \\
x_3(k+1) &= 0.01\sqrt{|x_1(k)|}, \\
x_4(k+1) &= -x_1(k) - x_2(k)^2 + \sin(x_4(k)).
\end{aligned} \tag{44}$$

Due to the reasons outlined in Section 5, the approach of [26] with 10^{19} samples results in a confidence of at least 10^{-30}, which is not practically useful. In contrast, with our techniques with 1000 samples we obtain a risk level of $\varepsilon = 0.02039$, with confidence at least $1 - 10^{-5}$.

7 Conclusions

We have proposed a method for synthesis of neural-network certificates, based only on a finite number of trajectories from a system, in order to verify a number of core temporally extended specifications. These certificates allow providing assertions on the satisfaction of the properties of interest. In order to synthesize a certificate, we considered a novel algorithm for solving a non-convex optimization program where the loss function we seek to minimize encodes different conditions on the certificate to be learned. As a byproduct of our algorithm, we determine a quantity termed "compression", which is instrumental in obtaining scalable probabilistic guarantees. This process is novel per se and provides a constructive mechanism for compression set calculation, thus opening the road for its use to more general non-convex optimization problems. Our numerical experiments demonstrate the efficacy of our methods on a number of examples, involving comparison with related methodologies in the literature.

Current work concentrates towards extending our analysis to continuous-time dynamical systems, as well as on considering controlled systems, thus co-designing a controller and a certificate at the same time.

References

[1] Alessandro Abate, Daniele Ahmed, Alec Edwards, Mirco Giacobbe, and Andrea Peruffo. FOSSIL: A software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks. In *HSCC*, pages 24:1–24:11. ACM, 2021.

[2] Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *ECC*, pages 3420–3431. IEEE, 2019.

[3] Thom S. Badings, Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Scenario-based verification of uncertain parametric MDPs. *Int. J. Softw. Tools Technol. Transf.*, 24(5):803–819, 2022.

[4] Thom S. Badings, Licio Romao, Alessandro Abate, David Parker, Hasan A. Poonawala, Mariëlle Stoelinga, and Nils Jansen. Robust control for dynamical systems with non-gaussian noise via formal abstractions. *Journal of Artificial Intelligence Research*, 76:341–391, 2023.

[5] Stephen P. Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2014.

[6] Mingyu Cai, Mohammadhosein Hasanbeig, Shaoping Xiao, Alessandro Abate, and Zhen Kan. Modular deep

reinforcement learning for continuous motion planning with temporal logic. *IEEE Robotics Autom. Lett.*, 6(4):7973–7980, 2021.

[7] Marco Campi and Simone Garatti. *Introduction to the Scenario Approach.* SIAM Series on Optimization, 2018.

[8] Marco C. Campi and Simone Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM J. Optim.*, 19(3):1211–1230, 2008.

[9] Marco C. Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.

[10] Marco C. Campi and Simone Garatti. Wait-and-judge scenario optimization. *Math. Program.*, 167(1):155–189, 2018.

[11] Marco C. Campi and Simone Garatti. Compression, generalization and learning. *J. Mach. Learn. Res.*, 24:339:1–339:74, 2023.

[12] Marco Claudio Campi, Simone Garatti, and Federico Alessandro Ramponi. A general scenario theory for nonconvex optimization and decision making. *IEEE Trans. Autom. Control.*, 63(12):4067–4078, 2018.

[13] Herman Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *The Annals of Mathematical Statistics*, 23(4):493 – 507, 1952.

[14] Frank H. Clarke. *Optimization and Nonsmooth Analysis.* Society for Industrial and Applied Mathematics, 1990.

[15] Alec Edwards, Andrea Peruffo, and Alessandro Abate. Fossil 2.0: Formal certificate synthesis for the verification and control of dynamical models. In *HSCC*, pages 26:1–26:10. ACM, 2024.

[16] Filiberto Fele and Kostas Margellos. Probably approximately correct nash equilibrium learning. *IEEE Trans. Autom. Control.*, 66(9):4238–4245, 2021.

[17] Sally Floyd and Manfred K. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Mach. Learn.*, 21(3):269–304, 1995.

[18] Simone Garatti and Marco C. Campi. Risk and complexity in scenario optimization. *Math. Program.*, 191(1):243–279, 2022.

[19] Carlos E. Garcia, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice - A survey. *Autom.*, 25(3):335–348, 1989.

[20] Morris W. Hirsch, Stephen Smale, and Robert L. Devaney. Differential equations, dynamical systems, and an introduction to chaos. 2003.

[21] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[22] Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. Formal synthesis of stochastic systems via control barrier certificates. *IEEE Trans. Autom. Control.*, 66(7):3097–3110, 2021.

[23] Alexander Mikhailovich Lyapunov. The general problem of the stability of motion. 1994.

[24] Kostas Margellos, Maria Prandini, and John Lygeros. On the connection between compression learning and scenario based single-stage and cascading optimization problems. *IEEE Trans. Autom. Control.*, 60(10):2716–2721, 2015.

[25] Peyman Mohajerin Esfahani, Tobias Sutter, and John Lygeros. Performance bounds for the scenario approach and an extension to a class of non-convex programs. *IEEE Transactions on Automatic Control*, 60(1):46–58, 2015.

[26] Ameneh Nejati, Abolfazl Lavaei, Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. Formal verification of unknown discrete- and continuous-time systems: A data-driven approach. *IEEE Trans. Autom. Control.*, 68(5):3011–3024, 2023.

[27] Dario Paccagnan, Marco C. Campi, and Simone Garatti. The pick-to-learn algorithm: Empowering compression for tight generalization bounds and improved post-training performance. In *NeurIPS*, 2023.

[28] Antonis Papachristodoulou, James Anderson, Giorgio Valmorbida, Stephen Prajna, Pete Seiler, and Pablo A. Parrilo. SOSTOOLS version 3.00 sum of squares optimization toolbox for MATLAB. *CoRR*, abs/1310.4716, 2013.

[29] Antonis Papachristodoulou and Stephen Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *CDC*, pages 3482–3487. IEEE, 2002.

[30] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *HSCC*, volume 2993 of *Lecture Notes in Computer Science*, pages 477–492. Springer, 2004.

[31] Stephen Prajna, Ali Jadbabaie, and George J. Pappas. Stochastic safety verification using barrier certificates. In *CDC*, pages 929–934. IEEE, 2004.

[32] Stephen Prajna, Ali Jadbabaie, and George J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Trans. Autom. Control.*, 52(8):1415–1428, 2007.

[33] Luke Rickard, Alessandro Abate, and Kostas Margellos. Learning robust policies for uncertain parametric Markov decision processes. In *L4DC*, volume 242 of *Proceedings of Machine Learning Research*, pages 876–889. PMLR, 2024.

[34] Luke Rickard, Thom S. Badings, Licio Romao, and Alessandro Abate. Formal controller synthesis for markov jump linear systems with uncertain dynamics. In *QEST*, volume 14287 of *Lecture Notes in Computer Science*, pages 10–29. Springer, 2023.

[35] Licio Romao, Antonis Papachristodoulou, and Kostas Margellos. On the exact feasibility of convex scenario programs with discarded constraints. *IEEE Trans. Autom. Control.*, 68(4):1986–2001, 2023.

[36] Ali Salamati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven verification and synthesis of stochastic systems via barrier certificates. *Autom.*, 159:111323, 2024.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[38] Graham R. Wood and B. P. Zhang. Estimation of the Lipschitz constant of a function. 8(1):91–103, 1996.

A Proofs

A.1 Certificate Proofs

A.1.1 Proof of Proposition 1 – Reachability Certificate

Fix $\delta > -\sup_{x\in X_I} V(x) \geq 0$, and recall that $k_G = \min\{k \in \{0,\dots,T\}\colon V(x(k)) \leq -\delta\}$. Consider then the difference condition in (7), namely,

$$\begin{aligned} &V(x(k+1)) - V(x(k)) \\ &< -\frac{1}{T}\Big(\sup_{x\in X_I} V(x) + \delta\Big),\ k = 0,\dots,k_G-1, \end{aligned} \tag{A.1}$$

By recursive application of this inequality $k \leq k_G$ times,

$$V(x(k)) < V(x(0)) - \frac{k}{T}\Big(\sup_{x\in X_I} V(x) + \delta\Big)$$
$$\leq \frac{T-k}{T}\sup_{x\in X_I} V(x) - \frac{k}{T}\delta \leq -\frac{k}{T}\delta \leq 0, \qquad \text{(A.2)}$$

where the second inequality is since $V(x(0)) \leq \sup_{x\in X_I} V(x)$, as $x(0) \in X_I$. The third one is since $\sup_{x\in X_I} V(x) \leq 0$ as by (3), $V(x) \leq 0$, for all $x \in X_I$, and $k \leq k_G \leq T$, while the last inequality is since $\delta > 0$.

By (A.2) we then have that for all $k \leq k_G$, $V(x(k)) < 0$, which implies that $x(k)$ does not leave X for all $k \leq k_G$ (see (5)), while by the definition of k_G, $x(k_G) \in X_G$. Notice that if $k_G = T$, then (A.2) (besides implying that $x(k) \in X$ for all $k \leq T$), also leads to $x(T) \leq -\delta$, which means that $x(T) \in X_G$ after T time steps (see (4)), which captures the latest time the goal set is reached.

Therefore, all trajectories that start within X_I reach the goal set X_G in at most T steps, without escaping X till then, thus concluding the proof. □

A.1.2 Proof of Proposition 2 – Safety Certificate

Consider the condition in (12), namely,

$$V(x(k+1)) - V(x(k))$$
$$< \frac{1}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big),\ k = 0,\dots,T-1. \qquad \text{(A.3)}$$

By recursive application of this inequality for $k \leq T$ times, we obtain

$$V(x(k)) < V(x(0)) + \frac{k}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big)$$
$$\leq \frac{T-K}{T}\sup_{x\in X_I} V(x) + \frac{k}{T}\inf_{x\in X_U} V(x)$$
$$\leq \frac{k}{T}\inf_{x\in X_U} V(x) \leq \inf_{x\in X_U} V(x). \qquad \text{(A.4)}$$

where the second inequality is is since $V(x(0)) \leq \sup_{x\in X_I} V(x)$, as $x(0) \in X_I$. The third inequality is since $\sup_{x\in X_I} V(x) \leq 0$ as by (10), $V(x) \leq 0$ for all $x \in X_I$ and $k \leq T$. The last inequality is since $\inf_{x\in X_U} V(x) \geq 0$, as by (11) $V(x) > 0$ for all $x \in X_U$, and $k \leq T$. We thus have

$$V(x(k)) < \inf_{x\in X_U} V(x),\ k = 1,\dots,T. \qquad \text{(A.5)}$$

and hence $x(k) \notin X_U, k = 0,\dots,T$ (notice that $x(0) \notin X_U$ holds since $X_I \cap X_U = \varnothing$). The latter implies that all trajectories that start in X_I avoid entering the unsafe set X_U, thus concluding the proof. □

A.1.3 Proof of Proposition 3 – RWA Certificaate

Fix $\delta > -\sup_{x\in X_I} V(x) \geq 0$, and recall that $k_G = \min\{k \in \{0,\dots,T\}\colon V(x(k)) \leq -\delta\}$. Consider then the difference condition in (19), namely,

$$V(x(k+1)) - V(x(k))$$
$$< -\frac{1}{T}\Big(\sup_{x\in X_I} V(x) + \delta\Big),\ k = 0,\dots,k_G-1, \qquad \text{(A.6)}$$

Note that this is identical to the difference condition for our reachability property, and hence following the same arguments with the proof of Proposition 1, we can infer that state trajectories emanating from X_I will reach the goal set X_G in at most T time steps.

By (18) we have that $V(x) > 0$, for all $x \in_U$ while by (15) we have that $V(x) \leq 0$, for all $x \in X_I$. Therefore, $\sup_{x\in X_I} V(x) \leq 0 \leq \inf_{x\in X_U} V(x)$. At the same time by our choice for δ we have that $\delta > -\sup_{x\in X_I} V(x)$. Combining these, we infer that $\delta > -\inf_{x\in X_U} V(x)$. Thus, (A.6) implies that for all $k = 0,\dots,k_G-1$,

$$-\frac{1}{T}\Big(\sup_{x\in X_I} V(x) + \delta\Big)$$
$$< \frac{1}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big). \qquad \text{(A.7)}$$

Therefore,

$$V(x(k+1)) - V(x(k)) \qquad \text{(A.8)}$$
$$< \frac{1}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big), k = 0,\dots,k_G-1.$$

Note that this is identical to the difference condition for our safety property, and hence following the same arguments with the proof of Proposition 2, we can infer that state trajectories emanating from X_I will never pass through the unsafe set X_U until time $k = k_G$.

Moreover, by (20), we have that

$$V(x(k+1)) - V(x(k))$$
$$< \frac{1}{T}\Big(\inf_{x\in X_U} V(x) + \delta\Big),\ k = k_G,\dots,T-1. \qquad \text{(A.9)}$$

Note that this is also a difference condition identical to that for our safety property, but with δ in place of $\sup_{x\in X_I} V(x)$ (since we know that $V(x(k_G)) \leq -\delta$ by definition of k_G). Hence, we have a safety condition for all trajectories emanating from this sublevel set. We know from (A.6) that trajectories reach this sublevel set, and hence remain safe for $k = k_G,\dots,T$

Therefore, we have shown that starting at X_I trajectories reach X_G in at most T time steps, while they never pass through X_U and do not leave the domain X, thus concluding the proof. □

A.2 Proof of Proposition 4 – Properties of Algorithm 1

(1) By construction, Algorithm 1 creates a non-increasing sequence of iterates $\{L_k\}_{k\geq 0}$ that is bounded below by the global minimum of $\max_{\xi\in D} L(\theta,\xi)$ which exists and is finite due to Assumption 3. As such, the sequence $\{L_k\}_{k\geq 0}$ is convergent, which in turn implies that Algorithm 1 terminates.

(2) We need to show that the set $\mathcal{C}_N$ is a compression set in the sense of Definition 2 with $\mathcal{A}$ being Algorithm 1 with $D = \{\xi_i\}_{i=1}^N$. To see this, we "re-run" Algorithm 1 from the same initial choice of the parameter vector θ but with $\mathcal{C}_N$ in place of D. Notice that exactly the same iterates will be generated, as $\mathcal{C}_N$ contains all samples that have led to a worst case loss across iterations (step 6). As a result, the same output will be returned, which by Definition 2 establishes that $\mathcal{C}_N$ is a compression set.

(3) We show that all properties of Assumption 2 are satisfied by Algorithm 1.

Preference: Consider a fixed (sample independent) initialization of Algorithm 1 in terms of the parameter θ. Consider also any subsets $\mathcal{C}_1, \mathcal{C}_2$ of $\{\xi^i\}_{i=1}^N$ with $\mathcal{C}_1 \subseteq \mathcal{C}_2$. Suppose that the compression set returned by Algorithm 1 when fed with $\mathcal{C}_2$ is different from $\mathcal{C}_1$. Fix any $\xi \in \Xi$ and consider the set $\mathcal{C}_2 \cup \{\xi\}$. We will show that the compression set returned by Algorithm 1 when fed with $\mathcal{C}_2 \cup \{\xi\}$ is different from $\mathcal{C}_1$ as well.

Case 1: The new sample ξ does not appear as a maximizing sample in step 9 of Algorithm 1, or its subgradient is such that the quantity in step 10 is positive. This implies that step 16 is not performed and the algorithm proceeds directly to step 18. As such, ξ is not added to the compression set returned by Algorithm 1, which remains the same with that returned when the algorithm is fed only by $\{\xi^i\}_{i=1}^N$. However, the latter is not equal to $\mathcal{C}_1$, thus establishing the claim.

Case 2: The new sample ξ appears as a maximizing sample in step 6 of Algorithm 1, and has a subgradient such that the quantity in step 10 is non-positive. As such, step 16 is performed and ξ is added to the compression returned by Algorithm 1. If $\xi \notin \mathcal{C}_1$ then the resulting compression set will be different from $\mathcal{C}_1$ as it would contain at least one element that is not $\mathcal{C}_1$, namely, ξ.

If $\xi \in \mathcal{C}_1$ then it must also be in $\mathcal{C}_2$ as $\mathcal{C}_1 \subseteq \mathcal{C}_2$. In that case ξ would appear twice in $\mathcal{C}_2 \cup \{\xi\}$, i.e., the set of samples with which Algorithm 1 is fed has ξ as a repeated sample (notice that this can happen with zero probability due to Assumption 1).

Once one of these repeated samples is added to the compression set returned by Algorithm 1, then the other will never be added. This is since when this other sample appears as a maximizing one in step 9 then its duplicate will already be in the compression set, and hence the exact and approximate subgradients in steps 10 and 12 would be identical. As such, the quantity in step 13 would be non-negative (and, by positive-definiteness of the inner product, only zero when both vectors are zero-vectors) and hence step 16 will not be performed, with the duplicate not added to the compression set. As such, one of the repeated ξ's is redundant, which implies that the compression set returned by Algorithm 1 when fed with $\mathcal{C}_2 \cup \{\xi\}$ is the same with the one that would be returned when it is fed with $\mathcal{C}_2$. However, this would imply that if $\mathcal{C}_1$ is the compression returned by Algorithm 1 when fed with of $\mathcal{C}_2 \cup \{\xi\}$, it will also be the compression set for $\mathcal{C}_2$ (as the duplicate ξ would be redundant). However, the starting hypothesis has been that $\mathcal{C}_1$ is not a compression of $\mathcal{C}_2$. As such, it is not possible for $\mathcal{C}_1$ to be a compression set of $\mathcal{C}_2 \cup \{\xi\}$ as well, establishing the claim.

Non-associativity: Consider a fixed (sample independent) initialization of Algorithm 1 in terms of the parameter θ. Let $\{\xi^i\}_{i=1}^{N+\bar{N}}$ for some $\bar{N} \geq 1$. Suppose that $\mathcal{C}$ is returned by Algorithm 1 a compression set of $\{\xi^i\}_{i=1}^N \cup \{\xi\}$, for all $\xi \in \{\xi^i\}_{i=N+1}^{N+\bar{N}}$. Therefore, up to a measure zero set we must have that

$$\mathcal{C} \subset \bigcap_{j=N+1}^{\bar{N}} \Big(\{\xi^i\}_{i=1}^N \cup \{\xi^j\} \Big) = \{\xi^i\}_{i=1}^N, \qquad \text{(A.10)}$$

where the inclusion is since $\mathcal{C}$ is assumed to be returned as a compression set by Algorithm 1 when this is fed with any set within the intersection, while the equality is since by Assumption 1 all samples in $\{\xi^i\}_{i=1}^{N+\bar{N}}$ are distinct up to a measure zero set. This implies that up to a measure zero set $\mathcal{C}$ should be a compression set returned by Algorithm 1 whenever this is fed with $\{\xi^i\}_{i=1}^N$ as any additional sample would be redundant.

Fix now any $\xi \in \{\xi^i\}_{i=N+1}^{N+\bar{N}}$, and consider Algorithm 1 with $D = \{\xi^i\}_{i=1}^N \cup \{\xi\}$. The fact that $\mathcal{C}$ is returned as a compression set for $\{\xi^i\}_{i=1}^N \cup \{\xi\}$ implies that whenever ξ is a maximizing sample in step 6 of Algorithm 1, it should give rise to a subgradient such that the quantity in step 10 of the algorithm is positive. This implies that step 18 is performed and hence ξ is not added to $\mathcal{C}$.

Considering Algorithm 1 this time with $D = \{\xi^i\}_{i=1}^{N+\bar{N}}$, i.e., fed with all samples at once, due to the aforementioned arguments, whenever a $\xi \in \{\xi^i\}_{i=N+1}^{N+\bar{N}}$ is a maximizing sample in step 6, then the algorithm would proceed to step 14, and steps 15–16 will not be executed. As such, no such ξ will be added to $\mathcal{C}$.

Hence, the compression set returned by Algorithm 1 when fed with $\{\xi^i\}_{i=1}^{N+\bar{N}}$ would be the same with the one that would be returned if the algorithm was fed with $\{\xi^i\}_{i=1}^N$. By (A.10) this then implies that the returned set should be $\mathcal{C}$ up to a measure zero set. $\square$