Automatica 185 (2026) 112798

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Data-driven certificate synthesis ☆

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

[a] Department of Engineering Science, University of Oxford, United Kingdom
[b] Department of Computer Science, University of Oxford, United Kingdom

ARTICLE INFO

Article history:
Received 8 February 2025
Received in revised form 8 August 2025
Accepted 19 November 2025
Available online 12 January 2026

Keywords:
Verification of dynamical systems
Safety
Reachability
Statistical learning
Scenario approach

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 system trajectories as data, we aim at learning a specific function termed *certificate* for each property we wish to verify. We seek to minimize a loss function, 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.

1. Introduction

Dynamical systems offer a rich class of models for describing the behavior of diverse, complex systems (Hirsch, Smale, & Devaney, 2003). It is often of importance that these systems meet certain properties, for example, stability, safety or reachability (Edwards, Peruffo, & Abate, 2024; Jagtap, Soudjani, & Zamani, 2021; Nejati, Lavaei, Jagtap, Soudjani, & Zamani, 2023; Prajna, Jadbabaie, & Pappas, 2007). Verifying the satisfaction of these properties, also termed as specifications, is a challenging, but important, problem.

One research direction involves discretizing the state space (Abate, Giacobbe, Roy, & Schnitzer, 2025; Badings et al., 2023; Rickard, Badings, Romao, & Abate, 2023), in order to construct a finite model with guarantees generated using probabilistic or statistical model checkers (Badings et al., 2022; Rickard, Abate, & Margellos, 2024). Discretizing the state space however tends to be computationally expensive, even for low-dimensional systems.

An alternative approach to verify properties of dynamical systems that does not require discretizing the state space, is through the use of *certificates* (Abate et al., 2025; Ames et al., 2019; Prajna & Jadbabaie, 2004). The goal is to determine a function over the system's state space that 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 (Lyapunov, 1994). Here we consider constructing reachability, safety, and reach-while-avoid (RWA) certificates for discrete time systems. Overviews of techniques for certificate learning can be found in Abate et al. (2025), Dawson, Gao, and Fan (2023); Table 1 summarizes the related literature, which we discuss below.

One approach to certificate synthesis considers verifying the behavior of systems assuming that a model of the underlying dynamical system is known. Restricting the class of models to polynomial functions, certificates can be obtained by solving a convex sum-of-squares problem (Papachristodoulou & Prajna, 2002). More generally, synthesis approaches leveraging SAT-modulo-theories can alternatively be leveraged (Ahmed et al., 2020; Dawson et al., 2023). Availability of a model allows for co-design of

☆ This work was supported by the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Marcello Farina under the direction of Editor Florian Dorfler.

* Corresponding author.
E-mail addresses: luke.rickard@eng.ox.ac.uk (L. Rickard), alessandro.abate@cs.ox.ac.uk (A. Abate), kostas.margellos@eng.ox.ac.uk (K. Margellos).

https://doi.org/10.1016/j.automatica.2025.112798

Table 1
Classification of certificate synthesis approaches.
* Theorem 1, and Algorithms 1 & 2, follow a non-convex scenario approach methodology, that does not require knowledge of the Lipschitz constant of the dynamics, and offer probabilistic verification bounds that do not necessarily scale exponentially in the state space dimension as with Nejati et al. (2023), Salamati, Lavaei, Soudjani, and Zamani (2024).

Model-Based Synthesis & Guarantees	Data-Driven Synthesis & Model-Based Guarantees	Data-Driven Synthesis & Guarantees
Sum of Squares Programming (Papachristodoulou & Prajna, 2002)	Counter-Example Guided Inductive Synthesis (Abate, Ahmed, Edwards, Giacobbe, & Peruffo, 2021; Chang, Roohi, & Gao, 2019; Dai, Landry, Pavone, & Tedrake, 2020; Edwards et al., 2024)	Neural Network Techniques (Anand & Zamani, 2023; Sun, Jha, & Fan, 2020)
MPC + Reachability analysis (Abate, Giacobbe, & Roy, 2024; Ren, Lu, Lv, Zhang, & Xue, 2023)	Neural Hamilton–Jacobi Reachability Analysis (Solanki et al., 2025; Yang, Hu, Wei, Li, & Liu, 2024)	Convex Scenario Optimization (Nejati et al., 2023; Salamati et al., 2024)
SAT-modulo-theory synthesis (Ahmed, Peruffo, & Abate, 2020)	Neural Certificates for Safety (Jin, Wang, Yang, & Mou, 2020)	Theorem 1, Algorithms 1 & 2*

a controller meeting certain specifications, such as safety and reachability (Abate et al., 2024; Ren et al., 2023), using tools based on Model Predictive Control (MPC), or reachability analysis.

The inclusion of the model's knowledge in the synthesis procedure in practice limits the complexity of the models that may be studied. To alleviate this requirement, data-driven techniques, such as counter-example guided inductive synthesis (CEGIS) (Abate et al., 2021; Chang et al., 2019; Dai et al., 2020; Edwards et al., 2024) are able to synthesize certificates for general nonlinear systems. This is achieved via the use of neural networks as certificate templates, allowing for the approximation of any function within a certain function space (Hornik, Stinchcombe, & White, 1989). Neural networks have also been explored as a tool in Solanki et al. (2025), Yang et al. (2024) for guaranteeing reachability, and in Jin et al. (2020) for safety analysis.

Such approaches involve data-driven synthesis, however, they still require a model of the system when it comes to providing guarantees on the synthesized certificates. Obtaining a model of the system is in general difficult, as it requires domain-specific knowledge. To alleviate these issues, in this work we follow a data-driven route that is model-free as far both the certificate synthesis and guarantee process is concerned. One way to achieve this involves using one set of samples to synthesize a certificate, and a separate set for validation (Sun et al., 2020). An alternative approach which does not require different samples for validation and is hence also the one most closely related to our formulation, involves probabilistic property satisfaction using a convex design and results on scenario optimization (Nejati et al., 2023; Salamati et al., 2024), and extensions to neural networks (Anand & Zamani, 2023; Sun et al., 2020). However, these developments rely on the system dynamics being Lipschitz continuous and for the Lipschitz constant to be known (or a bound on this to be available). Moreover, the probabilistic guarantees provided exhibit an exponential growth with respect to the system dimension. Both issues are not present within our proposed approach.

In this work, we follow a scenario approach paradigm as in Nejati et al. (2023), Salamati et al. (2024), however, we exploit some different statistical learning theoretic developments in scenario optimization. This allows us to remove the requirements for convexity and knowledge of the Lipschitz constant of the dynamics, and establish probabilistic verification bounds that do not necessarily scale exponentially on the state dimension, but their complexity rather depends on the complexity of the underlying property verification task.

In particular, we use any parameterized function approximator as certificate template, and learn these parameters using a finite number of system trajectories treated as samples. We formulate the certificate synthesis problem as a (possibly) non-convex optimization program, that involves minimizing an appropriately designed loss function, whose minimum value implies that a given property is satisfied. To minimize that loss function we also design a subgradient descent style procedure (Boyd & Vandenberghe, 2014). We accompany the synthesized certificate with *probably approximately correct* (PAC) guarantees on its validity, and hence on the probability of satisfying the underlying property, when it comes to a new system trajectory. It is to be noted that such a procedure does not require using a separate data-set for validation. To establish such PAC guarantees, we make use of bounds on the change of a quantity termed *compression set* (namely, a subset of the data which would return the same result as the entire set) (Campi & Garatti, 2023; Floyd & Warmuth, 1995; Margellos, Prandini, & Lygeros, 2015), through recent advancements of the so-called *scenario approach* (Campi & Garatti, 2008, 2018a, 2018b; Campi, Garatti, & Ramponi, 2018; Garatti & Campi, 2022). In particular, we are inspired by the novel theoretical *pick-to-learn framework* (Paccagnan, Campi, & Garatti, 2023), which provides a meta-algorithm for calculating a compression set with favorable properties. Here we extend the scope of the pick-to-learn framework by providing a constructive instance of the general framework to compute the cardinality of compression sets for non-convex optimization.

Our main contributions can be summarized as follows:

1. We develop a novel methodology for the synthesis of certificates to verify a wide class of properties, namely, reachability, safety and reach-while-avoid specifications, of discrete time dynamical systems. Our results complement the ones in Badings et al. (2022) 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. Capitalizing on developments on scenario optimization using the notion of compression, we accompany the constructed certificates with probabilistic guarantees on their generalization properties, namely, on how likely it is that the certificate remains valid when it comes to a new system trajectory. We contrast our approach with Nejati et al. (2023) 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 *compression set*, which is instrumental in obtaining meaningful probabilistic guarantees. This results in *a posteriori* bounds which, however, scale favorably with respect to the system dimension. This process is novel per se and provides a constructive approach for the general compression set calculation in Paccagnan et al. (2023), opening the road for its use in general non-convex optimization problems.

Notation. We use $\{\xi_k\}_{k=0}^{K}$ to denote a sequence indexed by $k \in \{0, 1, \ldots, K\}$. $V \models \psi$ 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 statements on the behavior of a dynamical system. Hence, we begin by defining a dynamical system, before considering the certificates and properties they verify.

2.1. Discrete time dynamical systems

We consider a bounded state space $X \subset \mathbb{R}^n$, 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 all 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 $f : X \to \mathbb{R}^n$, and assume it to permit unique solutions, but make no further 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. 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. 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 predefined control law structure, such as Model Predictive Control (Garcia, Prett, & Morari, 1989).

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 (for instance, 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 in practice, 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).

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 $V : \mathbb{R}^n \to \mathbb{R}$, let ψ^s and $\psi^{\Delta}(\xi)$ be conditions such that, if

$$\left[\exists V : \left(V \models \psi^s \wedge (\forall \xi \in \Xi) V \models \psi^{\Delta}(\xi)\right)\right] \implies \phi(\xi), \forall \xi \in \Xi,$$

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 trajectory-independent conditions in ψ^s, as well as the trajectory-dependent 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 associated certificates (and certificate conditions) that meet the format of Definition 1. We assume that V is continuous, so that when considering the supremum/infimum of V over a bounded set, 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 ∂X_G denotes its boundary. If, for all $\xi \in \Xi$,*

$$\phi_{\text{reach}}(\xi) := \exists k \in \{0, \ldots, T\} : x(k) \in X_G, \tag{2}$$

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. 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

$$V(x) \leq 0, \ \forall x \in X_I, \tag{3}$$

$$V(x) \geq -\delta, \ \forall x \in \partial X_G, \tag{4}$$

$$V(x) > -\delta, \ \forall x \in X \setminus X_G, \tag{5}$$

$$V(x) > 0, \ \forall x \in \mathbb{R}^N \setminus X, \tag{6}$$

$$V(x(k+1)) - V(x(k)) \tag{7}$$
$$< -\frac{1}{T}\left(\sup_{x \in X_I} V(x) + \delta\right), \ k = 0, \ldots, k_G - 1,$$

where $k_G := \min\{k \in \{0, \ldots, T\} : V(x(k)) \leq -\delta\}$, 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) (to ensure we do not leave the domain, where (7) may not hold), 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). Conditions (4)–(5) 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 V is decreasing along system trajectories till the first time the goal set is reached (by the definition of the time instance k_G). If T is allowed to tend 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 (Edwards et al., 2024). To gain some intuition on (7), see that if $k_G = T$, its recursive application leads to

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

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 Fig. 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.

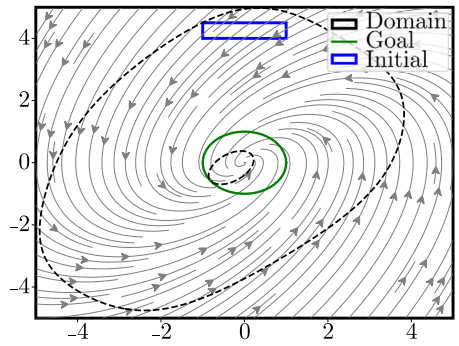

Fig. 1. Pictorial illustration of the level sets associated with the reach certificate for the system in (35).

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

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

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 $\xi \in \Xi$,*

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

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 that a system exhibits the safety property is equivalent to checking that all trajectories emanating from the initial set avoid the unsafe set for all time instances, until horizon T. The safety property may be constructed for unbounded X.

We now define relevant sufficient conditions for a certificate to verify this property, namely,

$$V(x) \leq 0, \forall x \in X_I, \tag{10}$$

$$V(x) > 0, \forall x \in X_U, \tag{11}$$

$$V(x(k+1)) - V(x(k)) \tag{12}$$
$$< \frac{1}{T}\left(\inf_{x \in X_U} V(x) - \sup_{x \in X_I} V(x)\right), k = 0, \dots, T-1.$$

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,

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

where the inequality holds 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. Since we consider finite horizon properties, this increase allows us to be less conservative compared with a simple negativity condition, which would be recovered if we allow T to tend to infinity.

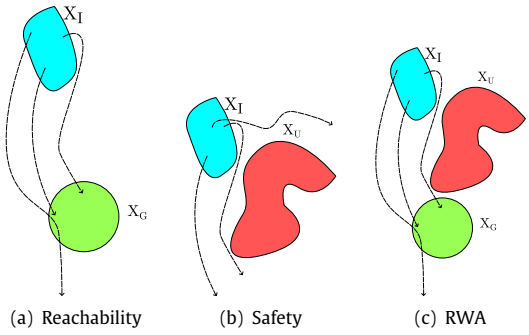

Fig. 2. Pictorial illustration of (a) reachability, (b) safety, and (c) RWA properties, respectively. Black lines illustrate sample trajectories that satisfy the associated properties.

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*

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

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 $\xi \in \Xi$,*

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

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.

The RWA property is derived from the reachability and safety properties, thus the conditions ψ^s_{RWA}, are the conjunction of ψ^s_{reach} and ψ^s_{safe}, and $\psi^{\Delta}_{\text{RWA}}(\xi)$ is given by the conjunction of $\psi^{\Delta}_{\text{reach}}(\xi)$ with the following requirement:

$$V(x(k+1)) - V(x(k)) \tag{15}$$
$$< \frac{1}{T}\left(\inf_{x \in X_U} V(x) + \delta\right), k = k_G, \dots, T-1,$$

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

$$V \models \psi^s_{\text{RWA}} \wedge (\forall \xi \in \Xi) V \models \psi^{\Delta}_{\text{RWA}}(\xi). \tag{16}$$

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

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

To obtain our sample set, we consider N initial conditions, sampled from $\mathbb{P}$, 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 mild 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.

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, or "risk", $\epsilon \in (0, 1)$ such that

$$\mathbb{P}^N \Big\{ \{\xi^i\}_{i=1}^N \in \Xi^N : \mathbb{P}\{\xi \in \Xi : \neg\phi(\xi)\} \leq \epsilon \Big\} \geq 1 - \beta. \tag{17}$$

We achieve this by considering a bound on the probability of a new trajectory satisfying our certificate conditions. 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, and hence the property holds (at least) with the same probability.

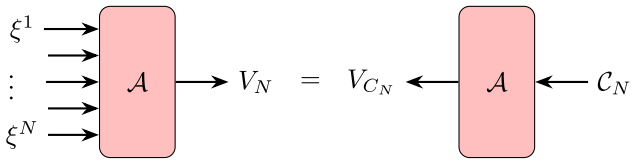

Fig. 3. Pictorial illustration of the compression set notion of Definition 2.

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 that exhibits certain properties 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.

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 $V_{\mathcal{C}_N} = \mathcal{A}(\mathcal{C}_N)$. We say that $\mathcal{C}_N$ is a compression of $\{\xi^i\}_{i=1}^N$ for algorithm $\mathcal{A}$, if

$$V_{\mathcal{C}_N} = \mathcal{A}(\mathcal{C}_N) = \mathcal{A}(\{\xi^i\}_{i=1}^N) = V_N. \tag{18}$$

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.

Fig. 3 illustrates Definition 2 pictorially. It should be noted that compression set cardinalities may be bounded *a priori* (Campi & Garatti, 2008), that is, without knowledge of the sample-set, or obtained *a posteriori*, and hence depending on the given set $\{\xi^i\}_{i=1}^N$ (Campi & Garatti, 2018b). We could take a compression set as the entire sample set, resulting in a trivial property violation 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 (Campi et al., 2018). Therefore, we introduce the subscript N in our notation for $\mathcal{C}_N$ (set) and C_N (corr. cardinality), respectively.

The properties this algorithm $\mathcal{A}$ must satisfy are as follows (adapted from Campi & Garatti, 2023).

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 $\{\xi^i\}_{i=1}^N$, 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).

If these are satisfied we may use the theorem below to provide probabilistic guarantees on property satisfaction.

Theorem 1 (*Probabilistic Guarantees*)**.** *Consider any algorithm* $\mathcal{A}$ *satisfying Assumption 2 such that* $V_N = \mathcal{A}(\{\xi^i\}_{i=1}^N) \models \bigwedge_{i=0}^N \psi^\Delta(\xi^i) \wedge$

ψ^s, with trajectories $\{\xi^i\}_{i=1}^N$ generated in an i.i.d. manner from a distribution satisfying Assumption 1. Fix $\beta \in (0,1)$, and for $k < N$, let $\varepsilon(k,\beta,N)$ be the (unique) solution to the polynomial equation in the interval $[k/N, 1]$

$$\frac{\beta}{2N}\sum_{m=k}^{N-1}\frac{\binom{m}{k}}{\binom{N}{k}}(1-\varepsilon)^{m-N} + \frac{\beta}{6N}\sum_{m=N+1}^{4N}\frac{\binom{m}{k}}{\binom{N}{k}}(1-\varepsilon)^{m-N} = 1, \tag{19}$$

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

$$\mathbb{P}^N\Big\{\{\xi^i\}_{i=1}^N \in \Xi^N : \mathbb{P}\{\xi \in \Xi : \neg\phi(\xi)\} \le \varepsilon(C_N,\beta,N)\Big\} \ge 1-\beta. \tag{20}$$

Proof. Fix $\beta \in (0,1)$, and for each $\{\xi^i\}_{i=1}^N$ 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.

We first demonstrate that if the certificate conditions are not satisfied on a new sample, then there will be a change in the compression set when the algorithm is fed all samples plus the new violating sample, as follows

$$\begin{aligned}\{\xi \in \Xi : V_N \not\models \psi^s \wedge \psi^\Delta(\xi)\} &\subseteq \{\xi \in \Xi : V_N \neq \mathcal{A}(\{\xi\}_{i=1}^N \cup \{\xi\})\}\\ &= \{\xi \in \Xi : \mathcal{A}(\mathcal{C}_N) \neq \mathcal{A}(\mathcal{C}_N^+)\}\\ &\subseteq \{\xi \in \Xi : \mathcal{C}_N \neq \mathcal{C}_N^+\},\end{aligned} \tag{21}$$

where $\mathcal{C}_N^+$ denotes a compression set for algorithm $\mathcal{A}$ when fed with $\{\xi\}_{i=1}^N \cup \{\xi\}$. 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., $\mathcal{A}(\{\xi^i\}_{i=1}^N \cup \{\xi\})$ (the output of our algorithm when fed with one more sample) is different from V_N. The opposite statement does not always hold, as having a different certificate does not necessarily mean the old one violates an existing condition for a new $\xi \in \Xi$. The equality holds as $V_N = \mathcal{A}(\mathcal{C}_N)$, and $\mathcal{A}(\{\xi\}_{i=1}^N \cup \{\xi\}) = A(\mathcal{C}_N^+)$, 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 $\mathcal{C}_N^+ \neq \mathcal{C}_N$. 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 $\mathcal{A}(\mathcal{C}_N) = \mathcal{A}(\mathcal{C}_N^+)$.

This derivation establishes the fact that the probability of V_N violating the property when it comes to a new ξ, is bounded by the probability that the compression set changes, i.e., we have that

$$\mathbb{P}\{\xi \in \Xi : V_N \not\models \psi^s \wedge \psi^\Delta(\xi)\} \le \mathbb{P}\{\xi \in \Xi : \mathcal{C}_N \neq \mathcal{C}_N^+\}. \tag{22}$$

We can now make use of Campi and Garatti (2023, 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 $\varepsilon(C_N,\beta,N)$, i.e.,

$$\mathbb{P}\{\xi \in \Xi : \mathcal{C}_N^+ \neq \mathcal{C}_N\} \le \varepsilon(C_N,\beta,N), \tag{23}$$

where the expression of $\varepsilon(k,\beta,N)$ for different values of k is given in (19). By (22) and (23), we have that

$$\mathbb{P}^N\Big\{\{\xi^i\}_{i=1}^N \in \Xi^N : \mathbb{P}\{\xi \in \Xi : V_N \not\models \psi^s \wedge \psi^\Delta(\xi)\} \le \varepsilon(C_N,\beta,N)\Big\} \ge 1-\beta.$$

By the implication in Definition 1, (20) follows, thus concluding the proof. □

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. 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 (Campi & Garatti, 2008; Margellos et al., 2015), 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). This is since in a non-convex setting, samples that give rise to inactive constraints may still belong to a compression set, as they may affect the optimal parameter implicitly.
3. An alternative procedure is to use sampled trajectories and to 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 Badings et al. (2022); 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 mechanisms to synthesize a certificate from sampled trajectories, thus offering a constructive approach for algorithm $\mathcal{A}$ in Theorem 1.

We treat a certificate as an appropriately parameterized "template" (e.g., neural network), and denote the parameter vector θ. We then have that our certificate V_N depends on θ, which is a vector we seek to identify to instantiate our certificate. For the results of this section, we simply write V_θ and drop the dependency on N to ease notation.

4.1. Certificate and compression set computation

We provide an algorithm that seeks to determine an optimal certificate parameterization $\theta^\star$, resulting in a certificate $V_{\theta^\star}$. To this end, for a $\xi \in \Xi$ and parameter vector θ, let

$$L(\theta,\xi) = l^\Delta(\theta,\xi) + l^s(\theta), \tag{24}$$

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 the quantity $\max_{\xi\in\mathcal{D}} L(\theta,\xi)$ 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. The motivating idea is to perform a subgradient descent step where one is allowed to follow an incorrect gradient as long as it points in the right direction. We

Algorithm 1 Certificate Synthesis and Compression Set Computation.

1: **function** $\mathcal{A}(\theta, \mathcal{D})$
2: Set $k \leftarrow 0$ ▷ Initialize iteration index
3: Set $\mathcal{C} \leftarrow \varnothing$ ▷ Initialize 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: **repeat**
9: $k \leftarrow k + 1$ ▷ Update iteration index
10: $\mathcal{M} \leftarrow \{\tilde{\xi} \in \mathcal{D} : L(\theta, \tilde{\xi}) \geq \max_{\xi \in \mathcal{C}} L(\theta, \xi)\}$ ▷ Find samples with loss greater than compression set loss
11: $\overline{g}_{\mathcal{M}} \leftarrow \{\nabla_\theta L(\theta, \tilde{\xi})\}_{\tilde{\xi} \in \mathcal{M}}$ ▷ Subgradients of loss function for $\tilde{\xi} \in \mathcal{M}$
12: $\overline{\xi}_{\mathcal{C}} \in \operatorname{argmax}_{\tilde{\xi} \in \mathcal{C}} L(\theta, \tilde{\xi})$ ▷ Find a sample with maximum loss from $\mathcal{C}$
13: $\overline{g}_{\mathcal{C}} \leftarrow \nabla_\theta L(\theta, \overline{\xi}_{\mathcal{C}})$ ▷ Approximate subgradient of loss function for $\tilde{\xi} = \overline{\xi}_{\mathcal{C}}$
14: **if** $\exists \overline{g} \in \overline{g}_{\mathcal{M}} : \langle \overline{g}, \overline{g}_{\mathcal{C}} \rangle \leq 0 \wedge \overline{g} \neq 0$ **then** ▷ If there is a misaligned subgradient (take the maximum if multiple)
15: $\theta \leftarrow \theta - \alpha \overline{g}$ ▷ Step in the direction of misaligned subgradient
16: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\overline{\xi}\}$ ▷ Update compression set with sample corresponding to $\overline{g}$
17: **else**
18: $\theta \leftarrow \theta - \alpha \overline{g}_{\mathcal{C}}$ ▷ Step in the direction of approximate subgradient
19: $L_k \leftarrow \min\left\{L_{k-1}, \max_{\xi \in \mathcal{C}} L(\theta, \xi)\right\}$ ▷ Update "running" loss value
20: **until** $|L_k - L_{k-1}| \leq \eta$ ▷ Iterate until tolerance is met
21: **return** $\theta, \mathcal{C}_N = \mathcal{C} \cup \operatorname{argmax}_{\xi \in \mathcal{D}} L(\theta, \xi)$

explain the main steps of Algorithm 1 with the aid of Fig. 4, where each sample gives rise to a concave triangular constraint.

Algorithm 1 takes as input 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 by means of a subgradient descent regime for the sample-independent loss until this loss is non-positive, which serves as a form of warm starting. Then, we follow the subgradient associated with the worst case sample and add it to the compression set $\mathcal{C}$ (step 15–16, point M_1 in Fig. 4). When iterates get to point a like M_2, the subgradient step becomes inexact, as for the same parameter there exists a sample resulting in a higher loss (see asterisk). Such a sample is in $\mathcal{M}$, step 10 of Algorithm 1. However, the algorithm does not "jump" to that point, as the inner-product condition in step 14 of the algorithm is not yet satisfied. Graphically, this is since the M_2 and the red asterisk are on a side of the respective constraint with the same slope. As such the algorithm performs inexact subgradient descent steps up to point M_3; this is the first instance where the condition in step 14 is satisfied (i.e., there exists another constraint with opposite slope[1]) and hence the algorithm "jumps" to a point with higher loss and subgradient of opposite sign. This procedure is then repeated as shown in the figure, with the red line indicating the iterates' path. The "jumps" serve as an exploration step to investigate the non-convex landscape, while their number (plus two for initial and final worst case sample) corresponds to the cardinality of the returned compression set. We iterate till the loss value meets a given tolerance η (see steps 20 and 19). It is to be understood that if $\mathcal{C}$ is empty (as per initialization) steps 12–13 are not performed.

Overall, Algorithm 1 can be viewed 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 follows a subgradient descent scheme with "jumps" that (i) allows minimizing a (possibly) non-convex loss function, and (ii) the mechanism that triggers the "jumps" provides the means to compute a compression set. Such a mechanism serves as an efficient alternative to existing methodologies, as we construct it iteratively. At the same time

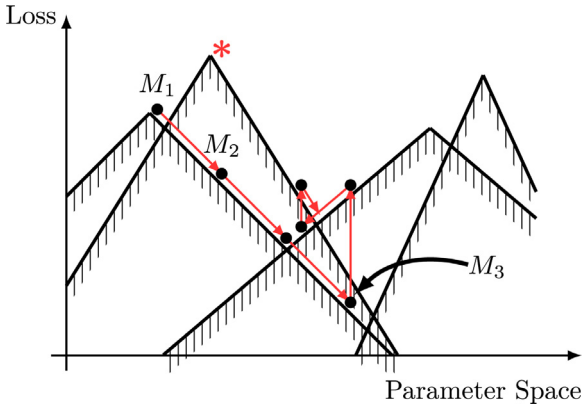

Fig. 4. Graphical illustration of Algorithm 1.

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 added sample has a loss higher than that of the compression samples (see step 10), and is also informative in the sense of having a misaligned subgradient that allows for exploration (see step 14). It should be highlighted that the underpinning idea of constructing the compression set by incrementing it by one sample at a time is inspired by the so called pick-to-learn paradigm proposed in Paccagnan et al. (2023). That methodology is general and does not involve a gradient-descent scheme equipped with our proposed logic. This design is thus novel and serves as a constructive instance of the general methodology of Paccagnan et al. (2023).

The main features of Algorithm 1 are summarized in the proposition below, while its proof can be found in Appendix A.

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 $\mathcal{C}_N$.

[1] This constraint with opposite slope may be any constraint with loss greater than the loss evaluated on the compression set, not just the maximum one.

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.

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, if the loss is driven to zero (the assumed minimum value), then all certificate conditions are met and hence the probabilistic guarantees obtained refer to guarantees on the probability of satisfaction of the underlying property. It should also be noted that in the numerical simulations presented below, we equipped the subgradient descent scheme with a momentum term thus constructing a deterministic version (as the step size is deterministic) of the so called Adam algorithm (Kingma & Ba, 2015) to boost performance.

4.2. Discarding mechanism

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 (Campi & Garatti, 2011; Romao, Papachristodoulou, & Margellos, 2023). To this end, consider Algorithm 2. At each iteration of this algorithm, the compression set returned by Algorithm 1 (step 5) is discarded from $\mathcal{D}$, and added to a record of the set of removed samples $\mathcal{R}$ (steps 6–7). We repeat the process till the worst case loss $\max_{\xi\in\mathcal{D}} L(\theta,\xi) \geq 0$ becomes zero (its minimum value). This implies that Algorithm 1 is invoked each time with fewer samples as its input, while the set $\mathcal{R}$ progressively increases. The set of samples that are removed across the algorithm's iterations is denoted by $\mathcal{R}_N$, and forms a compression set for 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.

Algorithm 2 Compression Set Update with Discarding.

1: Fix $\{\xi^i\}_{i=1}^N$
2: Set $\mathcal{C} \leftarrow \varnothing$ ▷ Initialize compression set
3: Set $\mathcal{D} \leftarrow \{\xi^i\}_{i=1}^N$ ▷ Initialize "running" samples
4: **while** $\max_{\xi\in\mathcal{D}} L(\theta,\xi) > 0$ **do**
5: $\theta, \mathcal{C} \leftarrow \mathcal{A}(\theta, \mathcal{D})$ ▷ Call Algorithm 1
6: $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{C}$ ▷ Discard compression set $\mathcal{C}$ from $\mathcal{D}$
7: $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{C}$ ▷ Store discarded samples
8: **return** θ, $\mathcal{R}_N = \mathcal{R}$

This algorithm can be thought of as an add-on to Algorithm 1 and in general to the procedure of Paccagnan et al. (2023) as it offers the means to trade the size of the compression set to performance. Unlike Algorithm 1 and Paccagnan et al. (2023, Algorithm 1) that iteratively increase the samples used for learning, Algorithm 2 gradually decreases the number of samples used as input to $\mathcal{A}$ across iterations.

Proposition 5 (*Algorithm 2 Properties*). *Consider Assumption 1, Assumption 3 and Algorithm 2 with $\mathcal{D} = \{\xi_i\}_{i=1}^N$ and a fixed (sample independent) initialization for the parameter θ. We then have:*

1. Algorithm 2 converges to a minimum loss value of zero, returning a parameter vector $\theta^\star$ and a set $\mathcal{R}_N$.
2. The set $\mathcal{R}_N$ with cardinality $R_N = |\mathcal{R}_N|$ forms a compression set for Algorithm 2.
3. Algorithm 2 satisfies Assumption 2.

The proof of this proposition can be found in Appendix A. Since it establishes that Algorithm 2 satisfies Assumption 2, we have that Algorithm 2 enjoys the guarantees of Theorem 1 with $\mathcal{R}_N$ in place of $\mathcal{C}_N$.

The cardinality of the compression set does not necessarily increase with the state space dimension, but is rather dependent on the complexity of the problem. For example, a problem where some trajectories approach or even enter the unsafe set presents a more challenging synthesis problem than one where trajectories all move in the opposite direction to the unsafe set, thus we expect the former to have a larger compression set even if the problem is smaller in dimension. This claim is supported numerically by the results of Section 6.

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 (Clarke, 1990).

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

$$l^s(V_\theta) := \int_{X\setminus X_G} \max\{0, -\delta - V_\theta(x)\}\,\mathrm{d}x + \int_{X_I} \max\{0, V_\theta(x)\}\,\mathrm{d}x + \int_{\mathbb{R}^N\setminus X} \max\{0, -V_\theta(x)\}\,\mathrm{d}x. \tag{25}$$

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. For a sufficiently expressive function approximator, 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; these samples are considered here to be fixed and they are not related with the ones used to provide probabilistic guarantees. 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. Our loss function takes then the form:

$$\hat{l}^s(V_\theta) := \frac{1}{|\mathcal{X}_{\overline{G}}|}\sum_{x\in\mathcal{X}_{\overline{G}}} \max\{0, -\delta - V_\theta(x)\} + \frac{1}{|\mathcal{X}_I|}\sum_{x\in\mathcal{X}_I} \max\{0, V_\theta(x)\} + \frac{1}{|\mathcal{X}_\partial|}\sum_{x\in\mathcal{X}_\partial} \max\{0, -V_\theta(x)\}. \tag{26}$$

We define l^Δ by

$$l^\Delta(V_\theta,\xi) := \max\left\{0, \max_{k=0,\dots,k_G-1}\Big(V_\theta(x(k+1)) - V_\theta(x(k))\Big) - \frac{1}{T}\Big(\sup_{x\in\mathcal{X}_I} V_\theta(x) + \delta\Big)\right\}. \tag{27}$$

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

$$l^{s}(V_\theta) + \max_{i=1,\dots,N} l^{\Delta}(V_\theta, \xi^i) = 0, \tag{28}$$

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

$$V_\theta \models \psi^{s}_{\text{reach}} \wedge (i = 1, \dots, N) V_\theta \models \psi^{\Delta}_{\text{reach}}(\xi^i). \tag{29}$$

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. However, if one is not interested in the construction of a certificate and only in such guarantees, then Theorem 2 in Badings et al. (2022) provides an alternative.

Proposition 6 (*Theorem 2 in Badings et al., 2022*)**.** *Fix $\beta \in (0, 1)$, and for $r = 0, \dots, N-1$, determine $\varepsilon(r, \beta, N)$ such that*

$$\sum_{k=0}^{r} \binom{N}{k} \varepsilon^k (1-\varepsilon)^{N-k} = \frac{\beta}{N}, \tag{30}$$

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

$$\mathbb{P}^N \Big\{ \{\xi^i\}_{i=1}^{N} \in \Xi^N : \mathbb{P}\{\xi \in \Xi : \neg\phi(\xi)\} \le \varepsilon(R_N, \beta, N) \Big\} \ge 1 - \beta. \tag{31}$$

This is an *a posteriori* result, as R_N can be determined only once the samples are observed. In this case, we have a compression set which is the set of all 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 (30).

We remark that one could obtain different bounds through alternative statistical techniques, such as Hoeffding's inequality (Hoeffding, 1963) or Chernoff's bound (Chernoff, 1952). Since these bounds are of different nature, we do not pursue that avenue further here.

We compare the risk levels ε computed by each approach on a benchmark example in (35) under a *safety* specification; general conclusions are case dependent, as both bounds are *a posteriori*. For a fixed β, Fig. 5 shows the resulting risk levels for varying N across 5 independently sampled sets of trajectories. The difference between the orange curve and the blue one can be interpreted as the price related to certificate generation, as per Theorem 1. For sufficiently large N, this price is marginal. As the specification is deterministically safe, no discarding is performed for Proposition 6, resulting in 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 Nejati et al. (2023)

The results in Nejati et al. (2023) constitute the closest to our work. As no results on reachability and RWA problems were provided in Nejati et al. (2023), 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 Nejati et al. (2023) are structurally different and of complementary nature to our work: next, we review the main result in Nejati et al. (2023), adapted to our notation.

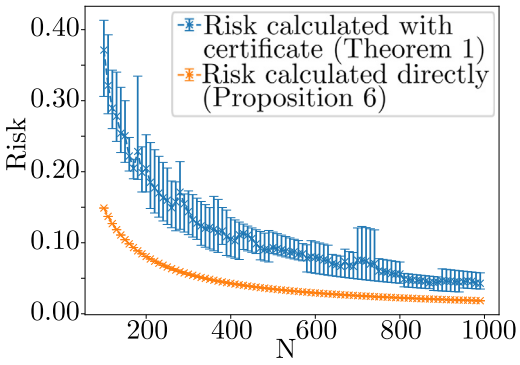

Fig. 5. Comparison of the bounds in Theorem 1 and Proposition 6 for direct property evaluation. Median values across the 5 runs are shown with a cross, and ranges are indicated by error bars.

Theorem 2 (*Theorem 5.3 in Nejati et al., 2023*)**.** *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 (24) is Lipschitz continuous with constant $\mathcal{L}$. Consider then the problem*

$$\begin{aligned} \eta_N^\star \in \underset{d=(\gamma,\lambda,c,\theta),\eta\in\mathbb{R}}{\arg\min} \quad & \eta \\ \text{st.} \quad & V_\theta(x) - \gamma \le \eta, \ \forall x \in X_I \\ & V_\theta(x) - \lambda \ge -\eta, \ \forall x \in X_U \\ & \gamma + cT - \lambda - \mu \le \eta, c \ge 0, \\ & V_\theta(f(x_i)) - V_\theta(x_i) - c \le \eta, \ i = 1, \dots, N, \end{aligned} \tag{32}$$

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

$$\kappa(\delta) \le \mathbb{P}\{\mathbb{B}_\delta(x)\}, \forall \delta \in \mathbb{R}_{\ge 0}, \forall x \in X, \tag{33}$$

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 (30), with $r = d$ and by replacing the right hand-side with β. If $\eta_N^\star \le \mathcal{L}\kappa^{-1}\epsilon(|d|, \beta, N)$, we have that

$$\mathbb{P}^N \Big\{ \{\xi^i\}_{i=1}^{N} \in \Xi^N : \ \phi_{\text{safe}}(\xi), \ \forall \xi \in \Xi \Big\} \ge 1 - \beta. \tag{34}$$

The following remarks are in order.

1. The result in Nejati et al. (2023), capitalizing on the developments of Mohajerin Esfahani, Sutter, and Lygeros (2015), is *a priori*, as opposed to the *a posteriori* assessments of our analysis that are in turn based on Campi and Garatti (2023). Moreover, Nejati et al. (2023) 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 (33) 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 Mohajerin Esfahani et al. (2015)), while it depends linearly on the dimension of the decision space $|d|$ (see dependence of ε below (33)). This dependence in the results of Nejati et al. (2023) raises computational challenges to obtain useful bounds: we demonstrate this numerically in Section 6 employing one of the examples considered in Nejati et al. (2023). On the contrary, Theorem 1 depends only on the cardinality of the compression set.
2. The result in Nejati et al. (2023) 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.

Table 2
Probabilistic guarantees for the system in (35). Standard deviations are shown in parentheses 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. 6	Empirical Property Risk $\hat{\varepsilon}$	Direct Bound Computation Time (s)
Reach Certificate (Proposition 1)	0.026 (0.004)	0 (0)	16597 (9157)	0.018 (0)	0 (0)	2.5 (0.3)
Safety Certificate (Proposition 2)	0.052 (0.017)	0 (0)	7924 (505)	0.018 (0)	0 (0)	7 (1)
RWA Certificate (Proposition 3)	0.037 (0.021)	0 (0)	20120 (15783)	0.018 (0)	0 (0)	7 (1)

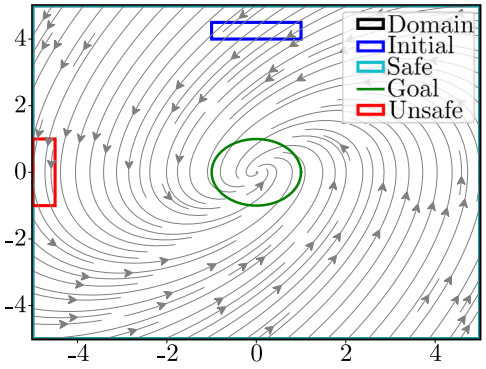

Fig. 6. Phase plane plot for the dynamical system of (35). The different sets shown are related to the sets that appear in the definitions of the reachability, safety and RWA property. For each case, only the relevant sets are considered.

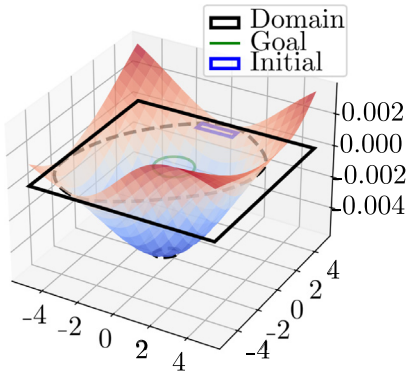

Fig. 7. Surface plot of the reachability certificate.

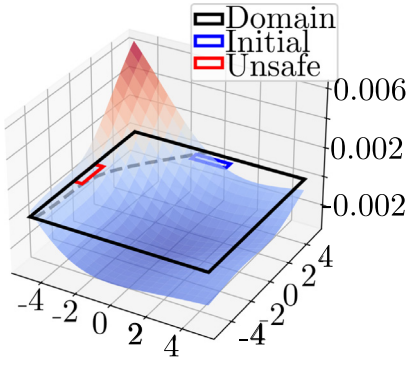

Fig. 8. Surface plot of the safety/barrier certificate.

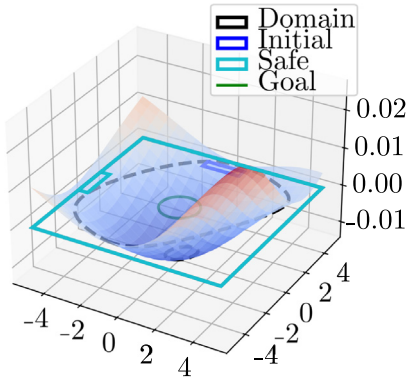

Fig. 9. Surface plot of the RWA certificate.

3. The results of Nejati et al. (2023) can be extended to continuous-time dynamical systems, which is also possible for our results but outside the scope of this article: we refer to Rickard, Abate, and Margellos (2025) for extensions to such cases.

6. Numerical results

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.[2]

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,

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) - \frac{T_d}{2} x_2(k) \\ x_2(k) + \frac{T_d}{2}(x_1(k) - x_2(k)) \end{bmatrix}, \tag{35}$$

where $T_d = 0.1$ and we use time horizon $T = 100$ steps. We used a neural network with 2 hidden layers, and 5 neurons per layer, with sigmoid activation functions thus leading to a parameter vector of size 51. The phase plane plot for these dynamics is in Fig. 6 alongside different sets related to the definition of reachability, safety and RWA properties are shown.

Surface plots of the reachability, barrier and RWA certificate are shown in Fig. 7, Figs. 8 and 9, respectively. The zero and

[2] The codebase is available at https://github.com/lukearcus/fossil_scenario

$-\delta$-sublevel sets of these certificates are highlighted with dashed black lines. With reference to Fig. 7 notice that the zero-sublevel set includes both the initial and the goal set, and no states outside the domain as expected. Similarly, in Fig. 8 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 2 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 2 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, or the underlying property. The number violating the certificate conditions (empirical certificate risk) is shown in the second column of Table 2, and the number violating the property (empirical property risk) is shown in the fifth column. Note that, as expected, the empirical values are lower than the theoretical bounds.

The fourth column of Table 2 provides the risk bound ε that would be obtained for direct property violation statements (however, without allowing for certificate construction) as per Proposition 6, 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 2 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

$$\begin{aligned} x_i(k+1) &= x_i(k) + 0.1x_{i+1}(k), \ \ i = 1 \dots 7, \\ x_8(k+1) &= x_8(k) - 0.1(576x_1(k) + 2400x_2(k) \\ &\quad + 4180x_3(k) + 3980x_4(k) + 2273x_5(k) \\ &\quad + 800x_6(k) + 170x_7(k) + 20x_8(k)). \end{aligned} \tag{36}$$

We define $X = [-2.2, 2.2]^8, X_I = [0.9, 1.1]^8, X_U = [-2.2, -1.8]^8$ and use a neural network with 2 hidden layers, and 10 neurons per layer, with sigmoid activation functions thus leading to a parameter vector of size 211. Once again, the entire of the initial set can be shown to be safe, 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 0.273 s, with a standard deviation of 0.018 s.

This certificate is computed much faster than those in Table 2, which 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 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 from it.

Due to the higher-dimensional state space, this certificate is not illustrated pictorially. It is accompanied by a probabilistic certificate $\varepsilon = 0.019$ (standard deviation 0.001) computed by means of Theorem 1. Using Proposition 6, we find a guarantee of 0.018 (standard deviation 0), after 2.26 s (standard deviation 0.05 s).

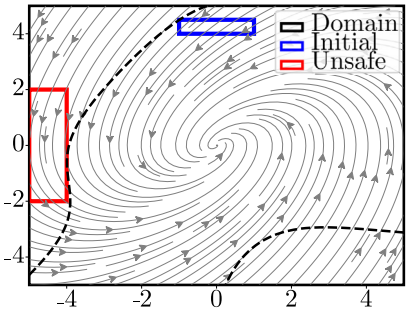

Fig. 10. Phase plane plot, initial and unsafe set for of partially unsafe system.

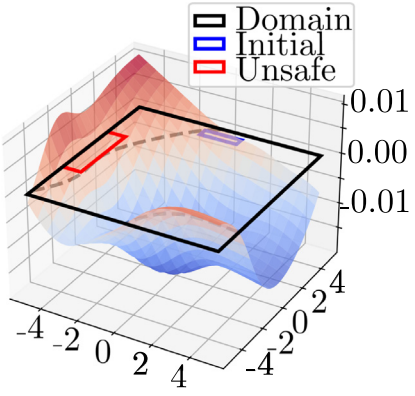

Fig. 11. Surface plot of the safety/barrier certificate for the partially unsafe system of Fig. 10.

6.3. Partially unsafe systems

We now consider the problem of safety certificate construction for the system in (35) with an enlarged unsafe region (see Fig. 10). We employ the same neural network as in Section 6.1. 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 (Edwards et al., 2024), or stochastic dynamics (Prajna, Jadbabaie, & Pappas, 2004), 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 Figs. 10 and 11. Fig. 11 provides a surface of the constructed certificate, and demonstrates that it separates the initial and the unsafe set. The computation time was 17971 s (standard deviation 1414 s).

For this certificate, we obtained a theoretical risk bound $\varepsilon = 0.388$ (standard deviation 0.035) by means of Theorem 1, and an empirical property risk of $\hat{\varepsilon} = 0.011$ (standard deviation 0.002). Proposition 6 gives risk bound 0.042 (standard deviation 0.004) after 3.2 s (standard deviation 0.1 s).

6.4. Comparison with Nejati et al. (2023)

We extend the comparison of our work with that in Nejati et al. (2023), which has been reviewed in Section 5. To this end, we construct a safety certificate for a two-dimensional DC Motor as considered in Nejati et al. (2023), using a neural network with 1 hidden layer, and 5 neurons per layer, with sigmoid activation functions thus leading to a parameter vector of size 21. We

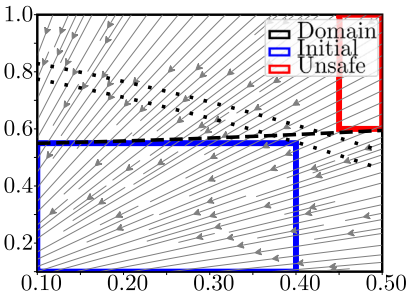

Fig. 12. Comparison with Nejati et al. (2023). The zero-level set of the safety certificate of our approach is dashed; level sets that separate the initial and unsafe sets (i.e. γ- and λ- level sets) from Nejati et al. (2023) are dotted.

first replicate the methodology of Nejati et al. (2023), using the Lipschitz constants they provide.

The methodology of Nejati et al. (2023) required 257149 samples and 307 s (standard deviation 44 s) of computation time to compute a barrier certificate with confidence at least equal to 0.99. Using 1000 samples and 1.4 s of computation time (standard deviation 0.1 s), we obtained $\varepsilon = 0.01$ (standard deviation 0), i.e. we can bound safety with a risk of 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. Fig. 12 illustrates a phase plane plot and the initial and unsafe sets for this problem. The dotted lines correspond to the sublevel sets constructed in Nejati et al. (2023) (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, a discretized version of a model taken from Edwards et al. (2024), with the required Lipschitz constants estimated using the technique in Wood and Zhang (1996).

$$\begin{aligned} x_1(k+1) &= x_1(k) + 0.1\left(\frac{x_1(k)x_2(k)}{5} - \frac{x_3(k)x_4(k)}{2}\right), \\ x_2(k+1) &= x_2(k) + 0.1\cos(x_4(k)), \\ x_3(k+1) &= x_3(k) + 0.001\sqrt{|x_1(k)|}, \\ x_4(k+1) &= x_4(k) + 0.1\left(-x_1(k) - x_2(k)^2 + \sin(x_4(k))\right). \end{aligned} \tag{37}$$

Due to the reasons outlined in Section 5, the approach of Nejati et al. (2023) 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 set", which is instrumental in obtaining scalable probabilistic guarantees. Our numerical experiments demonstrate the efficacy of our methods on a number of examples, involving comparison with related methodologies in the literature.

Appendix 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 $k_G = \min\{k \in \{0,\dots,T\} : V(x(k)) \leq -\delta\}$. 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,

$$\begin{aligned} V(x(k)) &< V(x(0)) - \frac{k}{T}\Big(\sup_{x\in X_I} V(x) + \delta\Big) \\ &\leq \frac{T-k}{T}\sup_{x\in X_I} V(x) - \frac{k}{T}\delta \leq -\frac{k}{T}\delta \leq 0, \end{aligned} \tag{A.2}$$

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 $V(x(T)) \leq -\delta$, 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,

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

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

$$\begin{aligned} V(x(k)) &< V(x(0)) + \frac{k}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big) \\ &\leq \frac{T-k}{T}\sup_{x\in X_I} V(x) + \frac{k}{T}\inf_{x\in X_U} V(x) \\ &\leq \frac{k}{T}\inf_{x\in X_U} V(x) \leq \inf_{x\in X_U} V(x). \end{aligned} \tag{A.4}$$

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 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

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

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 Certificate

Since we must satisfy ψ_{reach}, we can conclude that, following Proposition 1, state trajectories emanating from X_I will reach the goal set X_G in at most T time steps.

By (11) we have that $V(x) > 0$, for all $x \in_U$ while by (3) 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, (7) implies that for all $k = 0, \ldots, k_G - 1$,

$$-\frac{1}{T}\Big(\sup_{x\in X_I} V(x) + \delta\Big) < \frac{1}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big). \tag{A.6}$$

Therefore,

$$V(x(k+1)) - V(x(k)) < \frac{1}{T}\Big(\inf_{x\in X_U} V(x) - \sup_{x\in X_I} V(x)\Big), k = 0, \ldots, k_G - 1. \tag{A.7}$$

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 (15), we have that

$$V(x(k+1)) - V(x(k)) < \frac{1}{T}\Big(\inf_{x\in X_U} V(x) + \delta\Big), k = k_G, \ldots, T-1. \tag{A.8}$$

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 that trajectories reach this sublevel set, and hence remain safe for $k = k_G, \ldots, T$.

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, 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 $\min_{\xi\in\mathcal{D}} L(\theta, \xi)$ 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 $\mathcal{D} = \{\xi_i\}_{i=1}^N$. 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 $\mathcal{D}$. Notice that exactly the same iterates will be generated, as $\mathcal{C}_N$ contains all samples that have a misaligned subgradient and value greater than the loss evaluated on the running compression set. 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 10 of Algorithm 1, or its subgradient is such that the quantity in step 14 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 10 of Algorithm 1, and has a subgradient such that the quantity in step 14 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 10 then its duplicate will already be in the compression set, and hence the exact and approximate subgradients in steps 11 and 13 would be identical. As such, the quantity in step 14 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

$$\mathcal{C} \subset \bigcap_{j=N+1}^{\bar{N}} \Big(\{\xi^i\}_{i=1}^N \cup \{\xi^j\}\Big) = \{\xi^i\}_{i=1}^N, \tag{A.9}$$

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 $\mathcal{D} = \{\xi^i\}_{i=1}^N \cup \{\xi\}$. 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 10 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 $\mathcal{D} = \{\xi^i\}_{i=1}^{N+\bar{N}}$, 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 10, then the algorithm would proceed to step 18, 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.9) this then implies that the returned set should be $\mathcal{C}$ up to a measure zero set. □

A.3. Proof of Proposition 5

1. At every iteration, Algorithm 1, is called with fewer samples, and initialized on the optimal parameter set from the previous iteration. Hence, the loss value is a non-increasing sequence. If all samples are removed, the loss is zero, since we optimize only the sample-independent loss. Hence, the sequence converges to zero (in the worst-case upon removing all samples).
2. Consider Algorithm 2 with $\mathcal{D} = \{\xi_i\}_{i=1}^{N}$. Denote by $\mathcal{C}_i$ the set returned at step 5 of Algorithm 2, and recall that $\mathcal{C}_i \subseteq \mathcal{D}$ is the compression set returned by Algorithm 1 when this is invoked at that part of the process. Notice then that the set $\mathcal{R}_N$ returned by Algorithm 2 can be expressed as $\mathcal{R}_N = \bigcup_i \mathcal{C}_i$.

We need to show that $\mathcal{R}_N$ is a compression set in the sense of Definition 2 with $\mathcal{A}$ being Algorithm 2 with $\mathcal{D} = \{\xi_i\}_{i=1}^{N}$. To see this, we "re-run" Algorithm 2 from the same initial choice of the parameter vector θ but with $\mathcal{R}_N$ in place of $\mathcal{D}$. At the first iteration, the set returned in step 5 is $\mathcal{C}_1$ (and the parameter returned would be the same with the one that would be obtained if all samples were employed) as this is a compression set for Algorithm 1 invoked at that step with $\mathcal{D} = \mathcal{R}_{\mathcal{N}}$. As such, in step 6 and 7 we would, respectively, have that $\mathcal{D} = \mathcal{R}_N \setminus \mathcal{C}_1$, and $\mathcal{R} = \mathcal{R}_N$ since $\mathcal{C}_1$ is already in $\mathcal{R}_N$. Proceeding analogously, we have that Algorithm 2 terminates with the set $\mathcal{R}$ remaining intact to $\mathcal{R}_N$ and $\mathcal{D}$ being empty, and $\mathcal{R} = \mathcal{R}_N$. This establishes that $\mathcal{R}_N$ is a compression set for Algorithm 2.
3. Since Algorithm 1 satisfies Assumption 2, and we simply call this algorithm repeatedly, then Algorithm 2, also inherits these properties and satisfies Assumption 2.

References

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

Abate, Alessandro, Giacobbe, Mirco, & Roy, Diptarko (2024). Stochastic omega-regular verification and control with supermartingales. In Arie Gurfinkel, & Vijay Ganesh (Eds.), *Computer aided verification* (pp. 395–419).

Abate, Alessandro, Giacobbe, Mirco, Roy, Diptarko, & Schnitzer, Yannik (2025). Model checking and strategy synthesis with abstractions and certificates. In Nils Jansen, Sebastian Junges, Benjamin Lucien Kaminski, Christoph Matheja, Thomas Noll, Tim Quatmann, Mariëlle Stoelinga, & Matthias Volk (Eds.), *Principles of verification: cycling the probabilistic landscape, part II* (pp. 360–391).

Ahmed, D., Peruffo, A., & Abate, A. (2020). Automated and sound synthesis of Lyapunov functions with SMT solvers. In *Proceedings of TACAS, LNCS 12078* (pp. 97–114).

Ames, Aaron D., Coogan, Samuel, Egerstedt, Magnus, Notomista, Gennaro, Sreenath, Koushil, & Tabuada, Paulo (2019). Control barrier functions: Theory and applications. In *ECC* (pp. 3420–3431). IEEE.

Anand, Mahathi, & Zamani, Majid (2023). Formally verified neural network control barrier certificates for unknown systems. *IFAC-PapersOnLine*, *56*(2), 2431–2436.

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

Badings, Thom S., Romao, Licio, Abate, Alessandro, Parker, David, Poonawala, Hasan A., Stoelinga, Mariëlle, et al. (2023). Robust control for dynamical systems with non-Gaussian noise via formal abstractions. *Journal of Artificial Intelligence Research*, *76*, 341–391.

Boyd, Stephen P., & Vandenberghe, Lieven (2014). *Convex Optimization*. Cambridge University Press.

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

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

Campi, Marco, & Garatti, Simone (2018a). *SIAM Series on optimization, Introduction to the scenario approach*.

Campi, Marco C., & Garatti, Simone (2018b). Wait-and-judge scenario optimization. *Mathematical Programming*, *167*(1), 155–189.

Campi, Marco C., & Garatti, Simone (2023). Compression, generalization and learning. *Journal of Machine Learning Research*, *24*, 339:1–339:74.

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

Chang, Ya-Chien, Roohi, Nima, & Gao, Sicun (2019). Neural Lyapunov control. In *NeurIPS* (pp. 3240–3249).

Chernoff, Herman (1952). 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.

Clarke, Frank H. (1990). *Optimization and nonsmooth analysis*. Society for Industrial and Applied Mathematics.

Dai, Hongkai, Landry, Benoit, Pavone, Marco, & Tedrake, Russ (2020). Counter-example guided synthesis of neural network Lyapunov functions for piecewise linear systems. In *CDC* (pp. 1274–1281). IEEE.

Dawson, Charles, Gao, Sicun, & Fan, Chuchu (2023). Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control. *IEEE Trans. Robotics*, *39*(3), 1749–1767.

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

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

Garatti, Simone, & Campi, Marco C. (2022). Risk and complexity in scenario optimization. *Mathematical Programming*, *191*(1), 243–279.

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

Hirsch, Morris W., Smale, Stephen, & Devaney, Robert L. (2003). Differential equations, dynamical systems, and an introduction to chaos.

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

Hornik, Kurt, Stinchcombe, Maxwell B., & White, Halbert (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

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

Jin, Wanxin, Wang, Zhaoran, Yang, Zhuoran, & Mou, Shaoshuai (2020). Neural certificates for safe control policies. CoRR abs/2006.08465.

Kingma, Diederik P., & Ba, Jimmy (2015). Adam: A method for stochastic optimization. In *ICLR (poster)*.

Lyapunov, Alexander Mikhailovich (1994). The general problem of the stability of motion.

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

Mohajerin Esfahani, Peyman, Sutter, Tobias, & Lygeros, John (2015). 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.

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

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

Papachristodoulou, Antonis, & Prajna, Stephen (2002). On the construction of Lyapunov functions using the sum of squares decomposition. In *CDC* (pp. 3482–3487). IEEE.

Prajna, Stephen, & Jadbabaie, Ali (2004). Safety verification of hybrid systems using barrier certificates. In *Lecture Notes in Computer Science: Vol. 2993, HSCC* (pp. 477–492). Springer.

Prajna, Stephen, Jadbabaie, Ali, & Pappas, George J. (2004). Stochastic safety verification using barrier certificates. In *CDC* (pp. 929–934). IEEE.

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

Ren, Dejin, Lu, Wanli, Lv, Jidong, Zhang, Lijun, & Xue, Bai (2023). Model predictive control with reach-avoid analysis. In *IJCAI* (pp. 5437–5445). ijcai.org.

Rickard, Luke, Abate, Alessandro, & Margellos, Kostas (2024). Learning robust policies for uncertain parametric Markov decision processes. In *Proceedings of Machine Learning Research: Vol. 242, l4DC* (pp. 876–889). PMLR.

Rickard, Luke, Abate, Alessandro, & Margellos, Kostas (2025). Continuous-time data-driven barrier certificate synthesis. CoRR abs/2503.13392.

Rickard, Luke, Badings, Thom S., Romao, Licio, & Abate, Alessandro (2023). Formal controller synthesis for Markov jump linear systems with uncertain dynamics. In *Lecture Notes in Computer Science*: *Vol. 14287*, *QEST* (pp. 10–29). Springer.

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

Salamati, Ali, Lavaei, Abolfazl, Soudjani, Sadegh, & Zamani, Majid (2024). Data-driven verification and synthesis of stochastic systems via barrier certificates. *Autom.*, *159*, Article 111323.

Solanki, Prashant, Vertovec, Nikolaus, Schnitzer, Yannik, Beers, Jasper Van, de Visser, Coen, & Abate, Alessandro (2025). Certified approximate reachability (CARe): Formal error bounds on deep learning of reachable sets.

Sun, Dawei, Jha, Susmit, & Fan, Chuchu (2020). Learning certified control using contraction metric. In *Proceedings of Machine Learning Research*: *Vol. 155*, *CoRL* (pp. 1519–1539). PMLR.

Wood, Graham R., & Zhang, B. P. (1996). Estimation of the Lipschitz constant of a function. *8*, (1), (pp. 91–103).

Yang, Yujie, Hu, Hanjiang, Wei, Tianhao, Li, Shengbo Eben, & Liu, Changliu (2024). Scalable synthesis of formally verified neural value function for hamilton-Jacobi reachability analysis. CoRR abs/2407.20532.

Luke Rickard (IEEE Graduate Student Member) received a Masters in Engineering from the University of Oxford in 2021, and is currently working towards the D.Phil. in Engineering Science at the University of Oxford. He is currently working as a Research Associate at the University of East London, and Imperial College London.

Alessandro Abate (S'02-M'08-SM'19-F'24) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, Berkeley, CA, USA. He is Professor of Verification and Control with the Department of Computer Science, University of Oxford, Oxford, UK.

Kostas Margellos (IEEE Senior Member) received the Diploma in electrical engineering from the University of Patras, Greece, in 2008, and the Ph.D. in control engineering from ETH Zurich, Switzerland, in 2012. He spent 2013, 2014 and 2015 as a postdoctoral researcher at ETH Zurich, UC Berkeley and Politecnico di Milano, respectively. In 2016 he joined the Control Group, Department of Engineering Science, University of Oxford, where he is currently an Associate Professor. He is also a Fellow in AI & Machine Learning at Reuben College and a Lecturer at Worcester College. He is currently serving as Associate Editor in Automatica and in the IEEE Control Systems Letters, and is part of the Conference Editorial Board of the IEEE Control Systems Society and EUCA. His research interests include optimization and control of complex uncertain systems, with applications to energy and transportation networks.