Unifying Qualitative and Quantitative Safety Verification of DNN-Controlled Systems

Dapeng Zhi[1], Peixin Wang[2(✉)], Si Liu[3], C.-H. Luke Ong[2], and Min Zhang[1]

[1] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
[2] Nanyang Technological University, Singapore, Singapore
peixin.wang@ntu.edu.sg
[3] ETH Zurich, Zurich, Switzerland

Abstract. The rapid advance of deep reinforcement learning techniques enables the oversight of safety-critical systems through the utilization of Deep Neural Networks (DNNs). This underscores the pressing need to promptly establish certified safety guarantees for such DNN-controlled systems. Most of the existing verification approaches rely on qualitative approaches, predominantly employing reachability analysis. However, qualitative verification proves inadequate for DNN-controlled systems as their behaviors exhibit stochastic tendencies when operating in open and adversarial environments. In this paper, we propose a novel framework for unifying both qualitative and quantitative safety verification problems of DNN-controlled systems. This is achieved by formulating the verification tasks as the synthesis of valid neural barrier certificates (NBCs). Initially, the framework seeks to establish almost-sure safety guarantees through qualitative verification. In cases where qualitative verification fails, our quantitative verification method is invoked, yielding precise lower and upper bounds on probabilistic safety across both infinite and finite time horizons. To facilitate the synthesis of NBCs, we introduce their k-inductive variants. We also devise a simulation-guided approach for training NBCs, aiming to achieve tightness in computing precise certified lower and upper bounds. We prototype our approach into a tool called and showcase its efficacy on four classic DNN-controlled systems.

Keywords: Safety verification · DNN-controlled systems · Neural barrier certificates

1 Introduction

The widespread adoption of deep reinforcement learning techniques has propelled advancements in autonomous systems, endowing them with adaptive decision-making capabilities by Deep Neural Networks (DNNs) [36]. Ensuring the safety of these DNN-controlled systems emerges as a critical concern, necessitating the

A. Gurfinkel and V. Ganesh (Eds.): CAV 2024, LNCS 14682, pp. 401–426, 2024.
https://doi.org/10.1007/978-3-031-65630-9_20

provision of certified safety guarantees. Formal methods, renowned for their rigorousness and automaticity in delivering verified safety assurances, stand as a promising means to address this concern. However, most of the existing formal verification approaches rely on qualitative approaches, predominantly employing reachability analysis [47]. Despite their significance, qualitative results fall short for DNN-controlled systems due to the constant influence of various uncertainties from different sources, such as environment noises [68], unreliable sensors [55], and even malicious attacks [67]. When qualitative verification fails, it becomes both desirable and practical to obtain quantitative guarantees, including quantified lower and upper bounds on the safety probabilities of the systems. This necessitates the use of quantitative verification engines [47].

Quantitative verification has proven its efficacy in enhancing the design and deployment across a variety of applications, including autonomous systems [33], self-adaptive systems [13], distributed communication protocols [26], and probabilistic programs [57]. These applications are commonly modeled using automata-based quantitative formalisms [25], such as Markov chains, timed automata, and hybrid automata, and undergo verification using tools such as Prism [32] and Storm [27]. Nonetheless, the quantitative verification of DNN-controlled systems is challenging due to the incorporation of intricate and almost inexplicable decision-making models by DNNs [46]. Compounding the issue, the difficulty is amplified by the continuous and infinite state space, as well as the non-linear dynamics inherent in DNN-controlled systems. First, building a faithful automata-based probabilistic model for a DNN-controlled system is challenging. This difficulty arises as one cannot predict the action a DNN might take until a specific state is provided, and exhaustively enumerating all continuous states is impractical. Second, even if such a model is constructed under certain constraints, such as bounded steps [9] and state abstractions [31], verification is susceptible to state exploration issues—a well-known problem in model checking [52]. For instance, the verification process can take up to 50 minutes for just 7 steps [9].

Leveraging barrier certificates (BCs) for verification emerges as a promising technique for formally establishing the safety of non-linear and stochastic systems [34,43]. A BC partitions the state space of the system into two parts, ensuring that all trajectories starting from a given initial set, located within one side of the BC, cannot reach a given set of states (deemed to be unsafe), located on the other side, almost surely (i.e., with probability 1) or with probability at least $p \in [0, 1)$. Once a BC is computed, it can be used to certify systems' safety properties either qualitatively or quantitatively. Recently, studies have shown that BCs can be implemented and trained in neural forms called Neural Barrier Certificates (NBCs). NBCs facilitate the synthesis of BCs and improve their expressiveness [1,37,38,58,70]. A relevant survey is delegated to [18].

In this paper, we propose a unified framework for both qualitatively and quantitatively verifying the safety of DNN-controlled systems by leveraging NBCs. The key idea is to reduce both qualitative and quantitative verification problems into a cohesive synthesis task of their respective NBCs. Specifically, we

first seek to establish almost-sure safety guarantees through qualitative verification. In cases where qualitative verification fails, our quantitative verification method is invoked, yielding precise lower and upper bounds on probabilistic safety across both infinite and finite time horizons.

We also establish relevant theoretical results. In qualitative verification, we prove that an NBC satisfying corresponding conditions serves as a qualitative safety certificate. In the quantitative counterpart, we establish that valid NBCs can be utilized to calculate certified upper and lower bounds on the probabilistic safety of systems, encompassing both infinite and finite time horizons. For infinite time horizons, as the lower bounds on probabilistic safety approach zero, indicating a decreasing trend in safety probabilities along the time horizon, we provide both linearly and exponentially decreasing lower and upper bounds on the safety probabilities over finite time horizons.

To facilitate the synthesis of valid NBCs, we further relax their constraints by defining their k-inductive variants [6]. This necessitates the conditions to be inductive for k-compositions of the transition relation within a specified bound k [11]. Consequently, synthesizing a qualified NBC becomes more manageable under these k-inductive conditions, while ensuring safety guarantees. As valid NBCs are not unique and yield different certified bounds, we devise a simulation-guided approach to train potential NBCs. This approach aims to enhance their capability to produce more precise certified bounds. Specifically, we estimate safety probabilities through simulation. The differences between the simulation results and the bounds provided by potential NBCs are incorporated into the loss function. This integration can yield more precise certified bounds after potential NBCs are successfully validated.

We prototype our approach into a tool, called UniQQ, and apply it to four classic DNN-controlled problems. The experimental results showcase the effectiveness of our unified verification approach in delivering both qualitative and quantitative safety guarantees across diverse noise scenarios. Additionally, the results underscore the efficacy of k-inductive variants in reducing verification overhead, by 25% on average, and that of our simulation-based training method in yielding tighter safety bounds, with an up to 47.5% improvement over ordinary training approaches.

Contributions. Overall, we make the following contributions.

1. We present a novel framework that unifies both qualitative and quantitative safety verification of DNN-controlled systems by reducing these verification problems into the cohesive task of synthesizing NBCs.
2. We establish relevant theoretical results, including new constraints of NBCs for both qualitative and quantitative safety verification and the associated lower and upper bounds for safety probabilities in both linear and exponential forms.
3. To accelerate training, we relax the constraints of NBCs by introducing their k-inductive variants. We also present a simulation-guided approach designed to train potential NBCs to compute safety bounds as tightly as possible.

4. We develop a prototype of our approach, showcasing its efficacy across four classic DNN-controlled systems.

All omitted proofs and supplementary experimental results can be found in the full version [71].

2 Preliminaries

Let $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ be the sets of natural numbers, integers, and real numbers, respectively.

2.1 DNN-Controlled Systems

We consider DNN-controlled systems where the control policies are implemented by deep neural networks and suppose the networks are trained for specific tasks. Formally, a DNN-controlled system is a tuple $M = (S, S_0, A, \pi, f, R)$, where $S \subseteq \mathbb{R}^n$ is the set of (possibly continuous and infinite) system states, $S_0 \subseteq S$ is the set of initial states, A is the set of actions, $\pi : S \to A$ is the trained policy implemented by a neural network, $f : S \times A \to S$ is the system dynamics, and $R : S \times A \times S \to \mathbb{R}$ is the reward function.

Trajectories. A trained DNN-controlled system $M = (S, S_0, A, \pi, f, R)$ is a decision-making system that continuously interacts with the environment. At each time step $t \in \mathbb{N}_0$, it observes a state s_t and feeds s_t into its planted NN to compute the optimal action $a_t = \pi(s_t)$ that shall be taken. Action a_t is then performed, which transits s_t into the next state $s_{t+1} = f(s_t, a_t)$ via the system dynamics f and earns a reward $r_{t+1} = R(s_t, a_t, s_{t+1})$. Given an initial state $s_0 \in S_0$, a sequence of states generated during interaction is called a *trajectory*, denoted as $\omega = \{s_t\}_{t \in \mathbb{N}_0}$. To ease the notation, we denote by ω_t the t-th element of ω, i.e., $\omega_t = s_t$, and by Ω the set of all trajectories.

State Perturbations. As DNN-controlled systems collect state information via sensors, uncertainties inevitably originate from sensor errors, equipment inaccuracy, or even adversarial attacks [66,68]. Therefore, the observed states of the systems can be perturbed and actions are computed based on the perturbed states. Formally, an observed state at time step t is $\hat{s}_t := s_t + \delta_t$ where $\delta_t \sim \mu$ is a random noise and μ is a probability distribution over $\mathbb{R}^n$. We denote by $W := \mathsf{supp}(\mu)$ the support of μ. Due to perturbation, the actual successor state is $s_{t+1} := f(s_t, \hat{a}_t)$ with $\hat{a}_t := \pi(\hat{s}_t)$ and the reward is $r_{t+1} := R(s_t, \hat{a}_t, s_{t+1})$. Note that the successor state and the reward are calculated according to the actual state and the action on the perturbed state, and this update is common [68]. We then denote a DNN-controlled system M perturbed by a noise distribution μ as $M_\mu = (S, S_0, A, \pi, f, R, \mu)$.

Assumptions. Given a DNN-controlled system $M = (S, S_0, A, \pi, f, R)$, we assume that the state space S is compact in the Euclidean topology of $\mathbb{R}^n$,

its system dynamics f and trained policy π are Lipschitz continuous. We further assume that the system has forward invariance [62], i.e., all the states fall into the state space. These assumptions are common in control theory [4,72]. For perturbation, we require that the noise distribution μ either has bounded support or is a product of independent univariate distributions.

Probability Space. Given a DNN-controlled system $M_\mu = (S, S_0, A, \pi, f, R, \mu)$, for each initial state $s_0 \in S_0$, there exists a *probability space* $(\Omega_{s_0}, \mathcal{F}_{s_0}, \mathbb{P}_{s_0})$ such that Ω_{s_0} is the set of all trajectories starting from s_0 by the environmental interaction, $\mathcal{F}_{s_0}$ is a σ-algebra over Ω_{s_0} (i.e., a collection of subsets of Ω_{s_0} that contains the empty set $\emptyset$ and is closed under complementation and countable union), and $\mathbb{P}_{s_0} : \mathcal{F}_{s_0} \to [0, 1]$ is a probability measure on $\mathcal{F}_{s_0}$. We denote the expectation operator in this probability space by $\mathbb{E}_{s_0}$.

2.2 Barrier Certificate and Its Neural Implementation

Barrier certificates (BCs) are powerful tools to certify the safety of continuous-time dynamical systems. In the following we describe the discrete-time BCs which this work is based upon. We refer readers to [42,44] for details about continuous-time BCs.

Definition 1 (Discrete-time Barrier Certificates). *Given a DNN-controlled system $M = (S, S_0, A, f, \pi, R)$ with an unsafe set $S_u \subseteq S$ such that $S_u \cap S_0 = \emptyset$. A* discrete-time barrier certificate *is a real-valued function $B : S \to \mathbb{R}$ such that for some constant $\lambda \in (0, 1]$, it holds that:*

$$B(s) \leq 0 \qquad \text{for all } s \in S_0, \tag{1}$$

$$B(s) > 0 \qquad \text{for all } s \in S_u, \tag{2}$$

$$B(f(s, \pi(s))) - B(s) + \lambda \cdot B(s) \leq 0 \qquad \text{for all } s \in S. \tag{3}$$

If there exists such a BC for the system M, then M is safe, i.e., the system cannot reach a state in the unsafe set S_u from the initial set S_0. The intuition is that: Condition (3) implies that for any $s \in S$ such that $B(s) \leq 0$, $B(f(s, \pi(s))) \leq 0$. Since Condition (1) asserts that the initial value of B is not greater than zero, any trajectory $\omega \in \Omega_{s_0}$ starting from an initial state $s_0 \in S_0$ cannot enter the unsafe set S_u, where $B(s) > 0$ (see Condition (2)), thereby ensuring the safety of the system.

Finding a BC is restricted to the expressiveness of templates. For example, even if there exists a function satisfying Condition (1) to (3), it may be not found under polynomial forms. Recent work [41,69,70] proposes a neural implementation of BCs as deep neural networks, leveraging the expressiveness of neural networks. The neural implementation of a BC is called a neural barrier certificate (NBC), which consists of training and validation. First, a learner trains a neural network (NN) to fit over a finite set of samples the conditions for a BC. After training, an NBC is then checked whether it meets the conditions. This is achieved by a verifier using SMT solvers [41,70] or other methods like

Sum-of-Squares programming [69]. If the validation result is false, a set of counterexamples can be generated for future training. This iteration is repeated until a trained candidate is validated or a given timeout is reached. This training and validation iteration is called CounterExample-Guided Inductive Synthesis (CEGIS) [2].

3 Verification Problem and Our Framework

3.1 Problem Statement

We consider the safety of DNN-controlled systems from both qualitative and quantitative perspectives. Below we fix a DNN-controlled system $M_\mu = (S, S_0, A, \pi, f, R, \mu)$ and an unsafe set $S_u \subseteq S$ such that $S_0 \cap S_u = \emptyset$ throughout the paper.

Definition 2 (Almost-Sure Safety). *The system M_μ is* almost-surely (a.s.) safe*, if a.s. no trajectories starting from any initial state $s_0 \in S_0$ enter S_u, i.e.,*

$$\forall s_0 \in S_0 . \omega \in \Omega_{s_0} \Longrightarrow \omega_t \notin S_u \ \forall t \in \mathbb{N}.$$

This almost-sure safety is a qualitative property and we call it "almost-sure" due to the stochasticity from state perturbations. Since the almost-sure safety does not always exist with the increase of state perturbations, we propose the notion of probabilistic safety over infinite time horizons.

Definition 3 (Probabilistic Safety over Infinite Time Horizons). *The system M_μ is* probabilistically safe over infinite time horizons *with $[l_{\text{inf}}, u_{\text{inf}}]$, where $0 \leq l_{\text{inf}} \leq u_{\text{inf}} \leq 1$, if the probability of not entering S_u falls into $[l_{\text{inf}}, u_{\text{inf}}]$ for all the trajectories from any initial state $s_0 \in S_0$, i.e.,*

$$\forall s_0 \in S_0 . \mathbb{P}_{s_0} \left[\{ \omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \in \mathbb{N} \} \right] \in [l_{\text{inf}}, u_{\text{inf}}].$$

The probabilistic safety is a quantitative property and $l_{\text{inf}}, u_{\text{inf}}$ are called *lower* and *upper* bounds on the safety probabilities over infinite time horizons, respectively. Once both bounds equal one, it implies the almost-sure safety. When the lower bound $l_{\text{inf}} = 0$, indicating that the system reaches the unsafe region at some time step $T < \infty$, it is significant to figure out how the safety probability decreases over the finite time horizon. Therefore, we present the probabilistic safety over finite time horizons as follows.

Definition 4 (Probabilistic Safety over Finite Time Horizons). *The system M_μ is* probabilistically safe over a finite time horizon $T \in [0, \infty)$ *with $[l_{\text{fin}}, u_{\text{fin}}]$, where $0 \leq l_{\text{fin}} \leq u_{\text{fin}} \leq 1$, if the probability of not entering S_u within T falls into $[l_{\text{fin}}, u_{\text{fin}}]$ for all the trajectories starting from any initial state $s_0 \in S_0$,*

$$\forall s_0 \in S_0 . \mathbb{P}_{s_0} \left[\{ \omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \leq T \} \right] \in [l_{\text{fin}}, u_{\text{fin}}].$$

Safety Verification Problems of DNN-Controlled Systems. Consider a DNN-controlled system $M_\mu = (S, S_0, A, \pi, f, R, \mu)$ with an unsafe set $S_u \in S$ such that $S_0 \cap S_u = \emptyset$. We formulate the qualitative and quantitative safety verification problems of M_μ as follows:

1. **Qualitative Verification (QV):** To answer whether M_μ is almost-surely safe.
2. **Quantitative Verification over Infinite Time Horizons (QVITH):** To compute certified lower and upper bounds $l_{\text{inf}}, u_{\text{inf}}$ on the safety probability of M_μ over infinite time horizons.
3. **Quantitative Verification over Finite Time Horizons (QVFTH):** To compute certified lower and upper bounds $l_{\text{fin}}, u_{\text{fin}}$ on the safety probability of M_μ over a finite time horizon T.

3.2 Overview of Our Framework

We first provide an overview of our unified framework designed to address the three safety verification problems. Our framework builds on two fundamental results: (i) all the problems can be reduced to the task of defining BCs under specific conditions, and the defined BCs can be used to certify almost-sure safety for **QV** or safety bounds for **QVITH** and **QVFTH**, respectively, and (ii) these BCs can be implemented and trained in neural forms. The fundamental results are presented in Sects. 4 to 6, respectively.

The synthesis of NBCs has a preset timeout threshold, i.e., it will fail if NBCs cannot be successfully synthesized within the time threshold. The procedure of our framework is sketched in Fig. 1, which consists of the following three steps:

Step 1: QV. We try to synthesize an NBC satisfying conditions in Theorem 1. If such an NBC is successfully synthesized, we can conclude that the system M_μ is almost-surely safe by Theorem 1 and finish the verification. Alternatively, we can resort to synthesizing a k-inductive NBC in Theorem 8 whose conditions are weaker than those in Theorem 1. If the synthesis fails, we proceed to quantitative verification.

Step 2: QVITH. We try to synthesize two NBCs under the conditions in Theorems 2 and 3, respectively. If the synthesis fails, a timeout will be reported and the process will be terminated. Otherwise, we can obtain the lower bound l_{inf} and the upper bound u_{inf} on probabilistic safety over infinite time horizons. Alternatively, we can choose to synthesize the k-inductive variants of NBCs in Theorems 9 and 10. If the lower bound l_{inf} is no less than some preset safety threshold $\delta \in (0, 1)$, we terminate the verification. The purpose of setting δ is to prevent the verification from returning a meaningless lower bound such as 0. If l_{inf} is less than δ, we resort to computing safety bounds over finite time horizons.

Step 3: QVFTH. We try to synthesize two NBCs satisfying conditions in Theorems 4 and 6, respectively. If the synthesis fails, a timeout will be reported and the verification will terminate. Otherwise, we can compute the linear lower and upper bounds on probabilistic safety over finite time horizons according

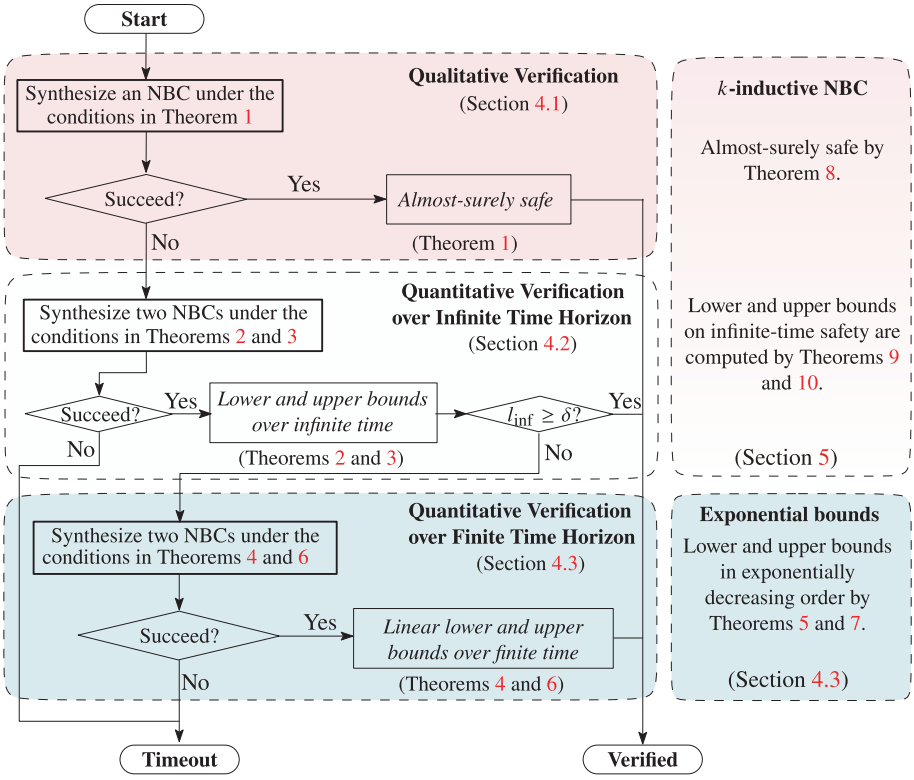

Fig. 1. UniQQ: The unified verification framework.

to the synthesized NBCs. Alternatively, we can choose to synthesize two NBCs satisfying conditions in Theorems 5 and 7 to achieve exponential bounds, which might be tighter than linear ones.

4 Qualitative and Quantitative Safety Verification

In this section, we reduce all three safety verification problems of DNN-controlled systems into a cohesive problem of defining corresponding BCs. We establish specific conditions for candidate BCs and provide formulas for computing lower and upper bounds for quantitative verification based on the defined BCs.

4.1 Qualitative Safety Verification

Theorem 1 (Almost-Sure Safety). *Given an M_μ with an initial set S_0 and an unsafe set S_u, if there exists a barrier certificate $B : S \to \mathbb{R}$ such that for some constant $\lambda \in (0, 1]$, the following conditions hold:*

$$B(s) \leq 0 \qquad \text{for all } s \in S_0, \tag{4}$$
$$B(s) > 0 \qquad \text{for all } s \in S_u, \tag{5}$$
$$B(f(s, \pi(s+\delta))) - B(s) + \lambda \cdot B(s) \leq 0 \qquad \text{for all } (s, \delta) \in S \times W, \tag{6}$$

then M_μ *is almost-surely safe, i.e.,* $\forall s_0 \in S_0.\ \omega \in \Omega_{s_0} \Longrightarrow \omega_t \notin S_u\ \forall t \in \mathbb{N}$.

Intuition. The BC in Theorem 1 is similar to that in Definition 1 except Condition (6), in which we consider all stochastic behaviors of the system from state perturbations. The proof of Theorem 1 resembles that in [43, Proposition 2].

Proof. We prove Theorem 1 by contradiction. Assume that there exists a barrier certificate B satisfying conditions (4)-(6), but the system is unsafe, i.e., there is a time step $T > 0$ and an initial state $s_0 \in S_0$ such that $s_T \in S_u$. Condition (6) implies that for any state $s \in S$ such that $B(s) \leq 0$ and a noise $\delta \in W$, the value of B at the next step is no more than zero, i.e., $B(f(s, \pi(s+\delta))) \leq 0$. As a result, $B(s_T)$ must be no more than zero, which is contradictory to Condition (5). Therefore, the system with a BC in Theorem 1 is almost-surely safe.

4.2 Quantitative Safety Verification over Infinite Time Horizon

Below we present the state-dependent lower and upper bounds on probabilistic safety over infinite time horizons.

Theorem 2 (Lower Bounds on Infinite-time Safety). *Given an* M_μ *with an initial set* S_0 *and an unsafe set* S_u*, if there exists a barrier certificate* $B : S \to \mathbb{R}$ *such that for some constant* $\epsilon \in [0, 1]$*, the following conditions hold:*

$$B(s) \geq 0 \qquad \text{for all } s \in S, \tag{7}$$
$$B(s) \leq \epsilon \qquad \text{for all } s \in S_0, \tag{8}$$
$$B(s) \geq 1 \qquad \text{for all } s \in S_u, \tag{9}$$
$$\mathbb{E}_{\delta \sim \mu}[B(f(s, \pi(s+\delta))) \mid s] - B(s) \leq 0 \qquad \text{for all } s \in S \setminus S_u, \tag{10}$$

then the safety probability over infinite time horizons is bounded from below by

$$\forall s_0 \in S_0.\ \mathbb{P}_{s_0}\left[\{\omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \in \mathbb{N}\}\right] \geq 1 - B(s_0). \tag{11}$$

Intuition. A BC under conditions in Theorem 2 is a non-negative real-valued function satisfying the supermartingale property, i.e., the expected value of the function remains non-increasing at every time step for all states not in S_u (see Condition (10)). The proof of Theorem 2 resembles that in [43, Theorem 15].

Proof (Sketch). To obtain the lower bound in Eq. (11), we first construct a stochastic process $\{X_t\}_{t \geq 0}$ where $X_t = B(s_t)$ with the safe initial state $s_0 \in S_0$ (see Condition (8)). Let κ be the first time that the system enters the unsafe set S_u. Then we prove that the stopped process of $\{X_t\}_{t \geq 0}$ w.r.t. κ is a non-negative supermartingale by Condition (7) and Condition (10). By Condition (9) and Ville's inequality [54], we have that $\mathbb{P}_{s_0}[s_t \in S_u \text{ for some } t \in \mathbb{N}] \leq X_0 = B(s_0)$. Finally, we obtain the lower bound in Eq. (11) by the complementation of the above upper bound.

Theorem 3 (Upper Bounds on Infinite-time Safety). *Given an M_μ with an initial set S_0 and an unsafe set S_u, if there exists a barrier certificate $B : S \to \mathbb{R}$ such that for some constants $\gamma \in (0,1)$, $0 \leq \epsilon' < \epsilon \leq 1$, the following conditions hold:*

$$0 \leq B(s) \leq 1 \qquad \text{for all } s \in S, \tag{12}$$
$$B(s) \geq \epsilon \qquad \text{for all } s \in S_0, \tag{13}$$
$$B(s) \leq \epsilon' \qquad \text{for all } s \in S_u, \tag{14}$$
$$B(s) - \gamma \cdot \mathbb{E}_{\delta\sim\mu}[B(f(s,\pi(s+\delta))) \mid s] \leq 0 \qquad \text{for all } s \in S \setminus S_u, \tag{15}$$

then the safety probability over infinite time horizons is bounded from above by

$$\forall s_0 \in S_0.\ \mathbb{P}_{s_0}\left[\{\omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \in \mathbb{N}\}\right] \leq 1 - B(s_0). \tag{16}$$

Intuition. A BC under conditions in Theorem 3 is a bounded non-negative function satisfying the γ-scaled submartingale property [53], i.e., the expected value of B is increasing at each time step for states not in S_u (Condition (15)). We prove the theorem by Optional Stopping Theorem [56], while the former work [50] is based on fixed-point theory [17].

Proof (Sketch). The proof is similar to that in Theorem 2. To obtain the upper bound in Eq. (16), we first construct a stochastic process $\{Y_t\}_{t\geq 0}$ such that $Y_t = \gamma^t B(s_t)$ with the safe initial state $s_0 \in S_0$ (see Condition (13) and Condition (14)). Let κ be the first time that the system enters the unsafe set S_u. Then we prove that the stopped process of $\{Y_t\}_{t\geq 0}$ w.r.t. κ is a submartingale by Condition (12) and Condition (15). By applying the Optional Stopping Theorem [56], we derive that $\mathbb{P}_{s_0}[s_t \in S_u \text{ for some } t \in \mathbb{N}] \geq B(s_0)$. Finally, we obtain the upper bound in Eq. (16) by the complementation of the derived lower bound.

4.3 Quantitative Safety Verification over Finite Time Horizon

When the safety probability over infinite time horizons exhibits a decline, it becomes advantageous to analyze the decreasing changes over finite time horizons. In the following, we present our theoretical results on finite-time safety verification, starting with two results related to lower bounds.

Theorem 4 (Linear Lower Bounds on Finite-time Safety). *Given an M_μ with an initial set S_0 and an unsafe set S_u, if there exists a barrier certificate $B : S \to \mathbb{R}$ such that for some constants $\lambda > \epsilon \geq 0$ and $c \geq 0$, the following conditions hold:*

$$B(s) \geq 0 \qquad \text{for all } s \in S, \tag{17}$$
$$B(s) \leq \epsilon \qquad \text{for all } s \in S_0, \tag{18}$$
$$B(s) \geq \lambda \qquad \text{for all } s \in S_u, \tag{19}$$
$$\mathbb{E}_{\delta\sim\mu}[B(f(s,\pi(s+\delta))) \mid s] - B(s) \leq c \qquad \text{for all } s \in S, \tag{20}$$

then the safety probability over a finite time horizon T is bounded from below by

$$\forall s_0 \in S_0. \mathbb{P}_{s_0}\left[\{\omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \leq T\}\right] \geq 1 - (B(s_0) + cT)/\lambda.$$

Intuition. A BC in Theorem 4 satisfies the c-martingale property [49], i.e., the expected value of B can increase at every time step as long as it is bounded by a constant c (Condition (20)), which is less conservative than the supermartingle property (Condition (10)), at the cost providing safety guarantees over finite time horizons. We prove the theorem by Ville's Inequality [54] and the proof resembles that in [6, Theorem 9].

Theorem 5 (Exponential Lower Bounds on Finite-time Safety). *Given an M_μ if there exists a function $B : S \to \mathbb{R}$ such that for some constants $\alpha > 0, \beta \in \mathbb{R}$, and $\gamma \in [0, 1)$, the following conditions hold:*

$$B(s) \geq 0 \qquad \text{for all } s \in S, \tag{21}$$
$$B(s) \leq \gamma \qquad \text{for all } s \in S_0, \tag{22}$$
$$B(s) \geq 1 \qquad \text{for all } s \in S_u, \tag{23}$$
$$\alpha \mathbb{E}_{\delta \sim \mu}[B(f(s, \pi(s+\delta))) \mid s] - B(s) \leq \alpha\beta \qquad \text{for all } s \in S \setminus S_u. \tag{24}$$

then the safety probability over a finite time horizon T is bounded from below by

$$\forall s_0 \in S_0. \mathbb{P}_{s_0}\left[\{\omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \leq T\}\right] \geq 1 - \frac{\alpha\beta}{\alpha - 1} + \left(\frac{\alpha\beta}{\alpha - 1} - B(s_0)\right) \cdot \alpha^{-T}.$$

Intuition. A BC in Theorem 5 satisfies that its α-scaled expectation can increase at most $\alpha\beta$ at every time step (Condition (24)). We establish a new result in discrete-time DNN-controlled systems and prove it by the discrete version of Gronwall's Inequality [24], which is inspired by former work [60] in continuous-time dynamical systems.

Then we propose our two results of upper bounds on safety probabilities.

Theorem 6 (Linear Upper Bounds on Finite-time Safety). *Given an M_μ with an initial set S_0 and an unsafe set S_u, if there exists a barrier function $B : S \to \mathbb{R}$ such that for some constants $\beta \in (0, 1), \beta < \alpha < 1 + \beta, c \geq 0$, the following conditions hold:*

$$B(s) \geq 0 \qquad \text{for all } s \in S, \tag{25}$$
$$B(s) \leq \beta \qquad \text{for all } s \in S \setminus S_u, \tag{26}$$
$$\alpha \leq B(s) \leq 1 + \beta \qquad \text{for all } s \in S_u, \tag{27}$$
$$\mathbb{E}_{\delta \sim \mu}[B(f(s, \pi(s+\delta))) \mid s] - B(s) \geq c \qquad \text{for all } s \in S \setminus S_u. \tag{28}$$

then the safety probability over a finite time horizon T is bounded from above by

$$\forall s_0 \in S_0. \mathbb{P}_{s_0}\left[\{\omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \leq T\}\right] \leq 1 - B(s_0) - \frac{1}{2}c \cdot T + \beta.$$

Intuition. A BC in Theorem 6 is non-negative and its value is bounded when states are in S_u (Condition (27)). Moreover, Condition (28) is the inverse of the c-martingale property in Theorem 4, i.e., the expected value of B should increase at least c at every time step.

Theorem 7 (Exponential Upper Bounds on Finite-Time Safety). *Given an M_μ with an initial set S_0 and an unsafe set S_u, if there exists a barrier function $B : S \to \mathbb{R}$ such that for some constants $K' \leq K < 0$, $\epsilon > 0$ and a non-empty interval $[a, b]$, the following conditions hold:*

$$B(s) \geq 0 \qquad \text{for all } s \in S \setminus S_u, \quad (29)$$

$$K' \leq B(s) \leq K \qquad \text{for all } s \in S_u, \quad (30)$$

$$\mathbb{E}_{\delta\sim\mu}[B(f(s, \pi(s+\delta))) \mid s] - B(s) \leq -\epsilon \qquad \text{for all } s \in S \setminus S_u, \quad (31)$$

$$a \leq B(f(s, \pi(s+\delta))) - B(s) \leq b \qquad \text{for all } s \in S \setminus S_u \text{ and } \delta \in W, \quad (32)$$

then the safety probability over a finite time horizon T is bounded from above by

$$\forall s_0 \in S_0.\ \mathbb{P}_{s_0}\left[\{\omega \in \Omega_{s_0} \mid \omega_t \notin S_u \text{ for all } t \leq T\}\right] \leq exp(-\frac{2(\epsilon \cdot T - B(s_0))^2}{T \cdot (b-a)^2}).$$

Intuition. A BC under Conditions (29) to (32) is a difference-bounded ranking supermartingale [16]. Condition (31) is the supermartingale difference condition, i.e., the expectation of B should decrease at least ϵ at each time step, while Condition (32) implies that the update of B should be bounded. We prove this theorem by Hoeffding's Inequality on Supermartingales [28] and the proof resembles that in the work [16].

Remark 1. In this section, we establish relevant theoretical results from the perspectives of qualitative and quantitative verification. In qualitative verification, we prove that an NBC satisfying corresponding conditions serves as a qualitative safety certificate. In the quantitative counterpart, we establish that valid NBCs can be utilized to calculate certified upper and lower bounds on the probabilistic safety of systems. It is worth noting that, for unifying safety verification in Fig. 1, new theoretical results (Theorem 5 and Theorem 6) are established, which mitigates the gaps of existing results [42,43].

Common conditions of different BCs. To clarify the construction of different BCs, we give three common categories of their conditions. The first two categories define the bounds of BCs for initial states and unsafe states, ensuring they are disjoint. The third category specifies the monotonicity of the (expected) BC values for successor states, yielding the possibility of the system reaching the unsafe set.

5 Relaxed k-Inductive Barrier Certificates

We now introduce k-inductive barrier certificates, capable of offering both qualitative and quantitative safety guarantees, while relaxing the strict conditions for

safety through the utilization of the k-induction principle [11,20]. Prior to presenting our theoretical results, we first define the notion of k-inductive update functions as follows.

Definition 5 (k-inductive Update Functions). *Given an $M_\mu = (S, S_0, A, \pi, f, R, \mu)$, a k-inductive update function $g^k_{\pi,f}$ with respect to π, f is defined recursively, i.e.,*

$$g^k_{\pi,f}(s_t, \Delta^k_t) = \begin{cases} g_{\pi,f}(g^{k-1}_{\pi,f}(s_t, \Delta^{k-1}_t), \delta_{t+k-1}) & \text{if } k > 1 \\ f(s_t, \pi(s_t + \delta_t)) & \text{if } k = 1 \\ s_t & \text{if } k = 0 \end{cases}$$

where $\Delta^k_t = [\delta_t, \delta_{t+1}, \ldots, \delta_{t+k-1}]$ is a noise vector of length k with each $\delta_t \sim \mu$, and $g_{\pi,f}(s_t, \delta_t) := f(s_t, \pi(s_t + \delta_t))$.

Intuitively, $g^k_{\pi,f}$ computes the value of a state after k steps given a k-dimensional noise vector $\Delta^k \in W^k \subseteq \mathbb{R}^{n\times k}$, where $W = \mathsf{supp}(\mu)$ is the support of μ. To calculate the expectation w.r.t. k-dimensional noises, we denote by μ^k the product measure on W^k.

5.1 k-Inductive Barrier Certificates for Qualitative Safety

Theorem 8 (k-inductive Variant of Almost-Sure Safety). *Given an M_μ with an initial set S_0 and an unsafe set S_u, if there exists a k-inductive barrier certificate $B : S \to \mathbb{R}$ such that the following conditions hold:*

$$\textstyle\bigwedge_{0\le i<k} B(g^i_{\pi,f}(s, \Delta^i)) \le 0 \qquad \forall (s, \Delta^i) \in S_0 \times W^i, \tag{33}$$

$$B(s) > 0 \qquad \forall s \in S_u, \tag{34}$$

$$\textstyle\bigwedge_{0\le i<k}(B(g^i_{\pi,f}(s, \Delta^i)) \le 0) \implies B(g^k_{\pi,f}(s, \Delta^k)) \le 0 \qquad \forall (s, \Delta^i) \in S \times W^i, \tag{35}$$

then the system M_μ is almost-surely safe, i.e., $\forall s_0 \in S_0.\ \omega \in \Omega_{s_0} \implies \omega_t \notin S_u\ \forall t \in \mathbb{N}$.

Intuition. Condition (33) implies that the state sequences starting from the safe set will remain in the safe set for the next $k-1$ consecutive time steps, while Condition (35) means that for any k consecutive time steps, if the system is safe, then the system will still be safe at the $(k+1)$-th time step. We prove the theorem by contradiction.

Note that Condition (35) contains an implication, in order to compute the k-inductive BC, we replace it with its sufficient condition:

$$-B(g^k_{\pi,f}(s, \Delta^k)) - \textstyle\sum_{0\le i<k} \tau_i \cdot (-B(g^i_{\pi,f}(s, \Delta^i))) \ge 0,\ \forall (s, \Delta^i) \in S \times W^i. \tag{36}$$

If there exist $\tau_0, \ldots, \tau_{k-1} \ge 0$ satisfying Eq. (36), Condition (35) is satisfied.

5.2 k-Inductive Barrier Certificates for Quantitative Safety

Theorem 9 (k-inductive Lower Bounds on Infinite-time Safety). *Given an M_μ, if there exists a k-inductive barrier certificate $B : S \to \mathbb{R}$ such that for some constants $k \in \mathbb{N}_{\geq 1}$, $\epsilon \in [0, 1]$ and $c \geq 0$, the following conditions hold:*

$$B(s) \geq 0 \qquad \text{for all } s \in S \tag{37}$$
$$B(s) \leq \epsilon \qquad \text{for all } s \in S_0, \tag{38}$$
$$B(s) \geq 1 \qquad \text{for all } s \in S_u, \tag{39}$$
$$\mathbb{E}_{\delta \sim \mu}[B(f(s, \pi(s+\delta))) \mid s] - B(s) \leq c \qquad \text{for all } s \in S, \tag{40}$$
$$\mathbb{E}_{\Delta^k \sim \mu^k}[B(f^k_{\pi,f}(s, \Delta^k)) \mid s] - B(s) \leq 0 \qquad \text{for all } s \in S, \tag{41}$$

then the safety probability over infinite time horizons is bounded from below by

$$\forall s_0 \in S_0.\ \mathbb{P}_{s_0}\left[\{\omega_0 \in \Omega_{s_0} \mid \omega_t \notin S_u \quad \text{for all } t \in \mathbb{N}\}\right] \geq 1 - kB(s_0) - \frac{k(k-1)c}{2}.$$

Intuition. Condition (40) requires the barrier certificate to be a c-martingale at every time step and Condition (41) requires the barrier certificate sampled after every k-th step to be a supermartingale. We prove the theorem by Ville's Inequality [54].

Theorem 10 (k-inductive Upper Bounds on Infinite-time Safety). *Given an M_μ, if there exists a barrier certificate $B : S \to \mathbb{R}$ such that for some constant $\gamma \in (0, 1)$, $0 \leq \epsilon' < \epsilon \leq 1$, $c \leq 0$ the following conditions hold:*

$$0 \leq B(s) \leq 1 \qquad \text{for all } s \in S \tag{42}$$
$$B(s) \geq \epsilon \qquad \text{for all } s \in S_0, \tag{43}$$
$$B(s) \leq \epsilon' \qquad \text{for all } s \in S_u, \tag{44}$$
$$\mathbb{E}_{\delta \sim \mu}[B(f(s, \pi(s+\delta))) \mid s] - B(s) \geq c \qquad \text{for all } s \in S, \tag{45}$$
$$B(s) - \gamma^k \cdot \mathbb{E}_{\Delta^k \sim \mu^k}[B(g^k_{\pi,f}(s, \Delta^k)) \mid s] \leq 0 \qquad \text{for all } s \in S \setminus S_u, \tag{46}$$

then the safety probability over infinite time horizons is bounded from above by

$$\forall s_0 \in S_0.\ \mathbb{P}_{s_0}\left[\{\omega \in \Omega_{s_0} \mid \omega_t \notin X_u \quad \text{for all } t \in \mathbb{N}\}\right] \leq 1 - kB(s_0) - \frac{k(k-1)c}{2}.$$

Intuition. This BC is non-negative and bounded (Condition 42). Condition (45) is the inverse of the c-martingale property, while Condition (46) requires the barrier certificate sampled after every k-th step to be a γ^k-scaled submartingale. We prove the theorem by the Optional Stopping Theorem [56].

Remark 2. To make the probabilistic bounds in Theorem 9 and Theorem 10 non-trivial, the value of k should be bounded by

$$1 \leq k \leq \frac{(c - 2B(s_0)) + \sqrt{4B(s_0)^2 + c^2 - 4c(B(s_0) - 2)}}{2c}.$$

Remark 3. In this section, we relax constraints to facilitate the synthesis of valid NBCs by defining their k-inductive variants [6]. Thus, synthesizing a valid NBC becomes more manageable under these k-inductive conditions, while ensuring safety guarantees. Besides, to our best knowledge, Theorem 10 is the first relaxation conclusion for upper bounds on infinite-time safety.

6 Synthesis of Neural Barrier Certificates

In this section, we show that the BCs defined in the previous sections for DNN-controlled systems can be implemented and synthesized in the form of DNNs, akin to those for linear or nonlinear stochastic systems [69].

We adopt the CEGIS-based method [2] to train and validate target NBCs. Figure 2 sketches the workflow. In each loop iteration, we train a candidate BC in the form of a neural network which is then passed to the validation. If the validation result is false, we compute a set of counterexamples for future training. This iteration is repeated until a trained candidate is validated or a given timeout is reached. Moreover, we propose a simulation-guided training method by adding additional terms to the loss functions to improve the tightness of upper and lower bounds calculated by the trained NBCs.

We present the synthesis of NBCs in Theorem 2 for probabilistic safety over infinite time horizons, as an example. We defer to the full version [71] the synthesis of other NBCs.

6.1 Training Candidate NBCs

Two pivotal factors in the training phase are the generation of training data and the construction of the loss function.

Training Data Discretization. As the state space S is possibly continuous and infinite, we choose a finite set of states for training candidate NBCs. This can be achieved by discretizing the state space S and constructing a *discretization* $\tilde{S} \subseteq S$ such that for each $s \in S$, there is a $\tilde{s} \in \tilde{S}$ with $||s - \tilde{s}||_1 < \tau$, where $\tau > 0$ is called the granularity of $\tilde{S}$. As S is compact and thus bounded, this discretization can be computed by simply picking the vertices of a grid with sufficiently small cells. For the retraining after validation failure, $\tilde{S}$ will be reconstructed with counterexamples and a smaller granularity τ. Once the discretization $\tilde{S}$ is obtained, we construct two finite sets $\tilde{S}_0 := \tilde{S} \cap S_0$ and $\tilde{S}_u := \tilde{S} \cap S_u$ used for the training process.

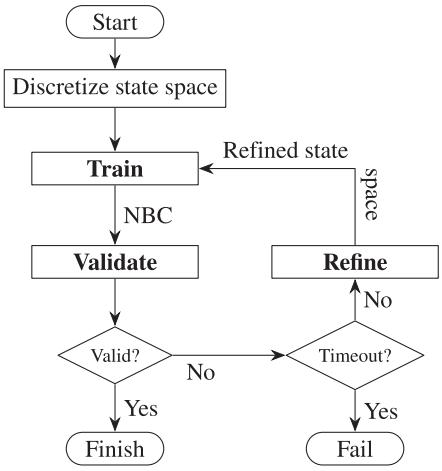

Fig. 2. CEGIS-based NBC synthesis [2].

Loss Function Construction. A candidate NBC is initialized as a neural network h_θ w.r.t. the network parameter θ. h_θ is trained by minimizing the following loss function:

$$\mathcal{L}(\theta) := k_1 \cdot \mathcal{L}_1(\theta) + k_2 \cdot \mathcal{L}_2(\theta) + k_3 \cdot \mathcal{L}_3(\theta) + k_4 \cdot \mathcal{L}_4(\theta) + k_5 \cdot \mathcal{L}_5(\theta)$$

where $k_i \in \mathbb{R},\ i = 1, \cdots, 5$ are the algorithmic parameters balancing the loss terms.

The first loss term is defined via the condition in Condition (7) as:

$$\mathcal{L}_1(\theta) = \frac{1}{|\tilde{S}|} \sum_{s \in \tilde{S}} (\max\{0 - h_\theta(s), 0\})$$

Intuitively, a loss will incur if either $h_\theta(s)$ is less than zero for any $s \in \tilde{S}$.

Correspondingly, the second and third loss terms are defined via Condition (8) and (9) as:

$$\mathcal{L}_2(\theta) = \frac{1}{|\tilde{S}_0|} \sum_{s \in \tilde{S}_0} (\max\{h_\theta(s) - \epsilon, 0\}), \text{and } \mathcal{L}_3(\theta) = \frac{1}{|\tilde{S}_u|} \sum_{s \in \tilde{S}_u} (\max\{1 - h_\theta(s), 0\}).$$

The fourth loss term is defined via the condition in Condition (10) as:

$$\mathcal{L}_4(\theta) = \frac{1}{|\tilde{S} \setminus \tilde{S}_u|} \sum_{s \in \tilde{S} \setminus \tilde{S}_u} \left(\max\{ \sum_{s' \in \mathcal{D}_s} \frac{h_\theta(s')}{N} - h_\theta(s) + \zeta, 0\} \right)$$

where for each $s \in \tilde{S} \setminus \tilde{S}_u$, $\mathcal{D}_s$ is the set of its successor states such that $\mathcal{D}_s := \{s' \mid s' = f(s, \pi(s + \delta_i)), \delta_i \sim \mu, i \in [1, N]\}$, $N > 0$ is the sample number of successor states. We use the mean of $h_\theta(\cdot)$ at the N successor states to approximate the expected value $\mathbb{E}_{\delta \sim \mu}[B(f(s, \pi(s + \delta)))]$ for each $s \in \tilde{S} \setminus \tilde{S}_u$, and $\zeta > 0$ to tighten the condition.

Simulation-Guided Loss Term. A trained BC that satisfies the above four conditions can provide lower bounds on probabilistic safety over infinite time horizons for the system. However, these conditions have nothing to do with the tightness of lower bounds and we may obtain a trivial zero-valued lower bound by the trained BC.

To assure the tightness of lower bounds from trained NBCs, we propose a simulation-guided method based on Eq. (11). For each $s_0 \in \tilde{S}_0$, we execute the control system $N' > 0$ episodes, and calculate the safety frequency $\mathbb{ff}_s$ of all the N' trajectories over infinite time horizons. Based on the statistical results, the last loss term is defined as:

$$\mathcal{L}_5(\theta) = \frac{1}{|\tilde{S}_0|} \sum_{s \in \tilde{S}_0} (\max\{\mathbb{ff}_s + h_\theta(s) - 1, 0\})$$

Intuitively, this term is to enforce the value of the derived lower bound to approach the statistical result as closely as possible, ensuring its tightness.

We emphasize that our simulation-guided method plus the NBC validation (see next section) is sound, as we will validate the trained BC to ensure it satisfies all the BC conditions (see also Theorem 12).

6.2 NBC Validation

A candidate NBC h_θ is valid if it meets the Conditions (7) to (10). The first three conditions condition can be checked by the following constraint

$$\inf_{s\in S} h_\theta(s) \geq 0 \ \wedge \ \sup_{s\in S_0} h_\theta(s) \leq \epsilon \ \wedge \ \inf_{s\in S_u} h_\theta(s) \geq 1$$

using the interval bound propagation approach [23,59]. When any state violates the above equation, it is treated as a counterexample and added to $\tilde{S}$ for future training.

For Condition (10), Theorem 11 reduces the validation from infinite states to finite ones, which are easier to check.

Theorem 11. *Given an M_μ and a function $B : S \to \mathbb{R}$, we have $\mathbb{E}_{\delta\sim\mu}[B(f(s,\pi(s+\delta))) \mid s] - B(s) \leq 0$ for any state $s \in S \setminus S_u$ if the formula below*

$$\mathbb{E}_{\delta\sim\mu}[B(f(\tilde{s},\pi(\tilde{s}+\delta))) \mid \tilde{s}] \leq B(\tilde{s}) - \zeta \tag{47}$$

holds for any state $\tilde{s} \in \tilde{S}\setminus\tilde{S}_u$, where $\zeta = \tau\cdot L_B\cdot(1+L_f\cdot(1+L_\pi))$ with L_f, L_π, L_B being the Lipschitz constants of f, π and B, respectively.

To check the satisfiablility of Eq. (47) in h_θ and a state $\tilde{s}$, we need to compute the expected value $\mathbb{E}_{\delta\sim\mu}[h_\theta(f(\tilde{s},\pi(\tilde{s}+\delta))) \mid \tilde{s}]$. However, it is difficult to compute its closed form because h_θ is provided in the form of neural networks. Hence, We bound the expected value $\mathbb{E}_{\delta\sim\mu}[h_\theta(f(\tilde{s},\pi(\tilde{s}+\delta))) \mid \tilde{s}]$ via interval arithmetic [23, 59] instead of computing it, which is inspired by the work [35,72]. In particular, given the noise distribution μ and its support $W = \{\delta \in \mathbb{R}^n \mid \mu(\delta) > 0\}$, we first partition W into finitely $m \geq 1$ cells, i.e., cell(W) $= \{W_1, \cdots, W_m\}$, and use maxvol $= \max_{W_i\in\text{cell(W)}} \text{vol}(W_i)$ to denote the maximal volume with respect to the Lebesgue measure of any cell in the partition, respectively. For the expected value in Eq. (47), we bound it from above:

$$\mathbb{E}_{\delta\sim\mu}[h_\theta(f(\tilde{s},\pi(\tilde{s}+\delta))) \mid \tilde{s}] \leq \sum_{W_i\in\text{cell(W)}} \text{maxvol}\cdot\sup_\delta F(\delta)$$

where, $F(\delta) = h_\theta(f(\tilde{s},\pi(\tilde{s}+\delta)))$. The supremum can be calculated via interval arithmetic. We refer interested readers to [35,72] for more details.

Theorem 12 (Soundness). *If a trained NBC is valid, it can certify the almost-sure safety for the qualitative verification, or the derived bound by the NBC is a certified lower/upper bound on the safety probability for the quantitative case.*

The proof of soundness is straightforward by the NBC validation.

7 Evaluation

Our experimental goals encompass evaluating the effectiveness of (i) the qualitative and quantitative verification methods within our framework, (ii) the k-inductive BCs, and (iii) the simulation-guided training method, respectively.

7.1 Benchmarks and Experimental Setup

We assess the effectiveness of our approach on four classic DNN-controlled tasks from public benchmarks: Pendulum and Cartpole from the DRL training platform OpenAI's Gym [12], while B1 and Tora commonly used by the state-of-the-art safety verification tools [30]. All experiments are executed on a workstation running Ubuntu 18.04, with a 32-core AMD Ryzen Threadripper CPU, 128 GB RAM, and a single 24564MiB GPU.

Table 1. Qualitative verification results.

Task	Perturbation	Verification	k	#Fail.
CP	0	✓	1	0
	$r = 0.01$	**Unknown**	1	0
	$r = 0.01$	✓	2	0
	$r = 0.03$	**Unknown**	1	207
PD	$r = 0$	✓	1	0
	$r = 0.01$	**Unknown**	1	675
	$r = 0.03$	**Unknown**	1	720
Tora	$r = 0$	✓	1	0
	$r = 0.02$	**Unknown**	1	0
	$r = 0.02$	✓	2	0
	$r = 0.04$	**Unknown**	1	1113
B1	$r = 0$	✓	1	0
	$r = 0.1$	✓	1	0
	$r = 0.2$	**Unknown**	1	43

The NBCs in this work are small fully-connected feedforward networks (FNNs) i.e., four-layer ReLU FNNs with $4{\times}64{\times}64{\times}1$. For the safety verification of DNN-controlled systems, we consider state perturbations of uniform noises with zero means and different radii. Specifically, for each state $s = (s_1, \ldots, s_n)$, we add noises $X_1, \ldots, X_n$ to each dimension of s and obtain the perturbed state $(s_1 + X_1, \ldots, s_n + X_n)$, where $X_i \sim \mathbf{U}(-r, r)$ $(1 \leq i \leq n,\ r \geq 0)$. We adopt the CEGIS-based method in Fig. 2 to train and validate target NBCs. For qualitative and various quantitative safety verification of these four systems, each synthesis of an NBC requires 3 iterations on average and each iteration produces an average of 1827 counterexamples.

For qualitative evaluations, the existence of an NBC in Theorem 1 can ensure the almost-sure safety of the whole system. Due to the data sparsity of an initial state, we randomly choose 10,000 initial states (instead of a single one) from the initial set S_0. For quantitative evaluations, to measure the quantitative safety probabilities from the system level, we calculate the mean values of lower/upper bounds by NBCs on these 10,000 states under different perturbations. The correctness of such system-level safety bounds is witnessed by Theorem 12 as each lower/upper bound on a single state s_0 is a certified bound for the exact safety probability from s_0, and thus the same holds on the system level. We also simulate 10,000 episodes starting from each of these 10,000 initial states under different perturbations and use the statistical results as the baseline.

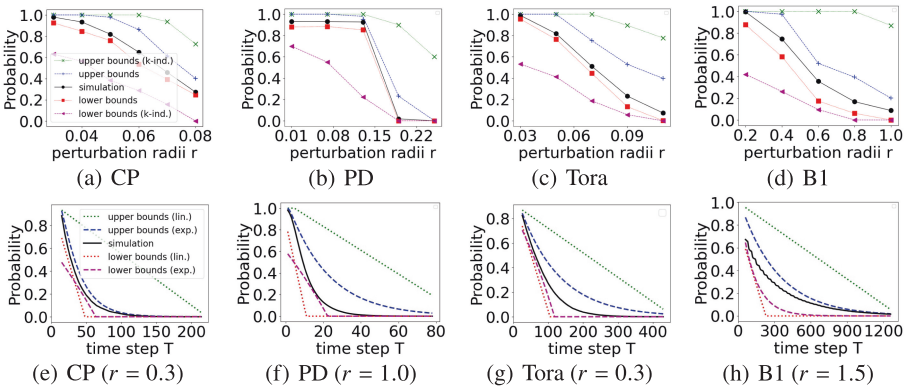

Fig. 3. The certified upper and lower bounds over infinite (a-d) and finite (e-h) time horizons, respectively, and their comparison with the simulation results. (Color figure online)

7.2 Effectiveness of Qualitative Safety Verification

Table 1 shows the qualitative verification results under different perturbation radii r's and induction bounds k's. Given a perturbed DNN-controlled system, we verify its qualitative safety by training an NBC under the conditions in Theorem 1. Once such an NBC is trained and validated, the system is verified to be almost-surely safe, marked as ✓. If no valid barrier certificates are trained within a given timeout, the result is marked as **Unknown**.

As for simulation, we record the number of those episodes where the system enters the unsafe region, marked as the column **#Fail.** in the table. We can observe that for the systems that are successfully verified by NBCs, no failed episodes are detected by simulation. For systems with failed episodes by simulation, no corresponding NBCs can be trained and validated. The consistency experimentally reflects the effectiveness of our approach.

Furthermore, we note that for CP with $r = 0.01$ and Tora with $r = 0.02$, there are no failed episodes, but no NBCs in Theorem 1 can be synthesized for these systems. By applying Theorem 8, we find the 2-inductive NBCs, which ensures the safety of the systems. It demonstrates that k-inductive variants can relax the conditions of NBCs and thus ease the synthesis of valid NBCs for qualitative safety verification.

As the perturbation radius increases, ensuring almost-sure safety becomes challenging, and qualitative verification only results in the conclusion of **Unknown**. Consequently, we proceed to conduct quantitative verification over infinite time horizons.

7.3 Effectiveness of Quantitative Safety Verification over Infinite Time Horizon

Figure 3 (a-d) show the certified upper and lower bounds and simulation results (i.e., black lines marked with '•') over infinite time horizons. The red lines marked with '■' and blue lines marked with '+' represent the mean values of the lower bounds in Theorem 2 and the upper bounds in Theorem 3 on the chosen 10,000 initial states calculated by the corresponding NBCs, respectively. The purple lines marked with '▲' and green lines marked with '×' represent the mean values of the 2-inductive upper and lower bounds calculated by the corresponding NBCs in Theorems 10 and 9, respectively. We can find that the certified bounds enclose the simulation outcomes, demonstrating the effectiveness of our trained NBCs.

Table 2 shows a comparison of average synthesis time (in seconds) for different NBCs. The synthesis time includes both training time and validation time. On average, the training time is 846 s and the validation time is 498 s. We observe that the synthesis time of 2-inductive NBCs is 25% faster than that of normal NBCs, at the sacrifice of tightness. Note that the tightness of certified bounds depends on specific systems and perturbations. Investigating what factors influence the tightness to yield tighter bounds is an interesting future work to explore.

Table 2. Synthesis time for different NBCs.

Task	Lower	2-Lower	Upper	2-Upper
CP	2318.5	1876.0	2891.9	2275.3
PD	1941.6	1524.0	2282.7	1491.5
Tora	280.3	218.5	895.1	650.7
B1	587.4	313.6	1127.3	840.1

Approaching zero for infinite time horizons, the lower bounds indicate a declining trend in safety probabilities over time. Therefore, we proceed to conduct quantitative verification over finite time horizons, providing both linear and exponential lower and upper bounds.

7.4 Effectiveness of Quantitative Safety Verification over Finite Time Horizon

Figure 3 (e-h) depict the certified upper and lower bounds and simulation results (i.e., black lines) over finite time horizons from the system level. Fix a sufficiently large noise level for each system, the x-axis represents the time horizon, while the y-axis corresponds to the safety probabilities. The purple lines and blue lines represent the mean values of the exponential lower and upper bounds calculated by the corresponding NBCs in Theorem 5 and Theorem 7, respectively. The red lines and green lines represent the mean values of the linear lower and upper bounds calculated by the corresponding NBCs in Theorem 4 and Theorem 6, respectively. The results indicate that our computed certified bounds encapsulate the statistical outcomes. Moreover, the exponential upper bounds are always tighter than the linear upper bounds, and the exponential lower bounds become tighter than the linear ones with the increase of time. It is worth exploring the factors to generate tighter results in future work.

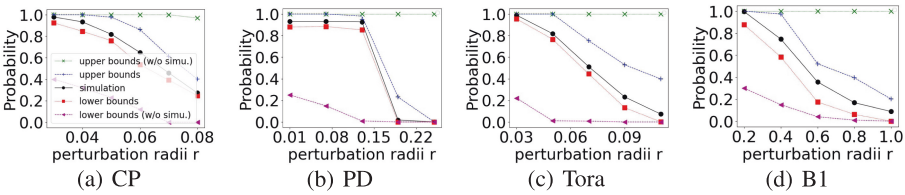

Fig. 4. The certified bounds w/ and w/o simulation-guided loss terms over infinite time horizons. (Color figure online)

7.5 Effectiveness of Simulation-Guided Loss Term

The simulation-guided loss term is proposed in Sect. 6.1 to tighten the certified bounds calculated by NBCs. To evaluate its effectiveness, we choose NBCs in Theorems 2 and 3, and train them with and without the simulation-guided loss terms. The comparison between them is shown in Fig. 4. The red lines marked with '■' and blue lines marked with '+' represent the mean values of the bounds in Theorems 2 and 3 on initial states calculated by the corresponding NBCs trained with the simulation-guided loss terms, respectively. The purple lines with '▲' and green lines with '×' represent the mean values of the bounds calculated by the NBCs trained without the simulation-guided loss terms. Apparently, the upper and lower bounds derived by NBCs trained without the simulation-guided loss terms are looser than the bounds trained with these terms. Specifically, the results computed by NBCs with simulation-guided loss terms can achieve an average improvement of 47.5% for lower bounds and 31.7% for upper bounds, respectively. Hence, it is fair to conclude that accounting for simulation-guided loss terms is essential when conducting quantitative safety verification.

8 Related Work

Barrier Certificates for Stochastic Systems. Our unified safety verification framework draws inspiration from research on the formal verification of stochastic systems employing barrier certificates. Prajna *et al.* [42–44] propose the use of barrier certificates in the safety verification of stochastic systems. This idea has been further expanded through data-driven approaches [45] and k-inductive variants [6]. As the dual problem of computing safety probabilities, computing reachability probabilities in stochastic dynamical systems has been studied for both infinite [22,62,63] and finite time horizons [60,61]. Alireza *et al.* [3] represent non-negative repulsing supermartingales as neural networks and use them to derive upper bounds on the finite-time reachability probability. Probabilistic programs, viewed as stochastic models, have their reachability and termination probabilities investigated using proof rules [21] and martingale-based approaches [7,15,16], where the latter are subsequently unified through order-theoretic fixed-point approaches [50,51,53].

Formal Verification of DNN-Controlled Systems. Modeling DNN-controlled systems as Markov Decision Processes (MDPs) and verifying these models using probabilistic model checkers, such as PRISM [32] and Storm [27], constitutes a quantitative verification approach. Bacci and Parker [9,10] employ abstract interpretation to construct interval MDPs and yield safety probabilities within bounded time. Carr *et al.* [14] propose probabilistic verification of DNN-controlled systems by constraining the analysis to partially observable finite-state models. Amir *et al.* propose a scalable approach based on DNN verification techniques to first support complex properties such as liveness [5].

Reachability analysis is a pivotal qualitative approach in the safety verification of DNN-controlled systems. Bacci *et al.* [8] introduce a linear over-approximation-based method for calculating reachable set invariants over an infinite time horizon for DNN-controlled systems. Other reachability analysis approaches, such as Verisig [30] and Polar [29], focus solely on bounded time. These approaches do not consider perturbations as they assume actions on states to be deterministic.

Barrier Certificates for Training and Verifying DNN Controllers. BC-based methods [1,41] have recently been investigated for training and verifying DNN controllers. The key idea is to train a safe DNN controller through interactive computations of corresponding barrier certificates to ensure qualitative safety [19,64]. Vishnu *et al.* [40] present a data-driven algorithm for training a neural network to represent the closure certificates in [39]. Existing BC-based approaches for the verification of DNN-controlled systems focus solely on qualitative aspects but neglect the consideration of perturbations [40,48,65]. Our approach complements them by accommodating the inherent uncertainty in DNN-controlled systems.

9 Conclusion and Future Work

We have systematically studied the BC-based qualitative and quantitative safety verification of DNN-controlled systems. This involves unifying and transforming the verification problems into a general task of training corresponding neural certificate barriers. We have also defined the conditions that a trained certificate should satisfy, along with the corresponding lower and upper bounds presented in both linear and exponential forms and k-inductive variants. Through the unification of these verification problems, we have established a comprehensive framework for delivering various safety guarantees, whether qualitatively or quantitatively, in a unified manner.

Our framework sheds light on the quest for scalable and multipurpose safety verification of DNN-controlled systems. It accommodates both qualitative and quantitative aspects in verified results, spans both finite and infinite time horizons, and encompasses certified bounds presented in both linear and exponential forms. Our work also showcases the potential to circumvent verification challenges posed by DNN controllers. From our experiments, we acknowledge that both qualitative and quantitative verification results are significantly dependent

on the quality of the trained NBCs. Our next step is to explore more sophisticated deep learning methods and hyperparameter settings (e.g., the architecture of NBCs and the k-inductive horizon) to train valid NBCs for achieving more precise verification results.

Acknowledgments. We thank the anonymous reviewers for their valuable comments. The work has been supported by the NSFC Programs (62161146001, 62372176), Huawei Technologies Co., Ltd., the Shanghai International Joint Lab (22510750100), the Shanghai Trusted Industry Internet Software Collaborative Innovation Center, and the National Research Foundation, Singapore, under its RSS Scheme (NRF-RSS2022-009).

References

1. Abate, A., Ahmed, D., Edwards, A., Giacobbe, M., Peruffo, A.: FOSSIL: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks. In: HSCC, pp. 24:1–24:11 (2021)
2. Abate, A., David, C., Kesseli, P., Kroening, D., Polgreen, E.: Counterexample guided inductive synthesis modulo theories. In: CAV, pp. 270–288 (2018)
3. Abate, A., Edwards, A., Giacobbe, M., Punchihewa, H., Roy, D.: Quantitative verification with neural networks. In: CONCUR. LIPIcs, vol. 279, pp. 22:1–22:18 (2023)
4. Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., Tabuada, P.: Control barrier functions: Theory and applications. In: ECC, pp. 3420–3431 (2019)
5. Amir, G., Schapira, M., Katz, G.: Towards scalable verification of deep reinforcement learning. In: FMCAD, pp. 193–203 (2021)
6. Anand, M., Murali, V., Trivedi, A., Zamani, M.: k-inductive barrier certificates for stochastic systems. In: HSCC, pp. 12:1–12:11 (2022)
7. Asadi, A., Chatterjee, K., Fu, H., Goharshady, A.K., Mahdavi, M.: Polynomial reachability witnesses via stellensätze. In: PLDI, pp. 772–787 (2021)
8. Bacci, E., Giacobbe, M., Parker, D.: Verifying reinforcement learning up to infinity. In: IJCAI, pp. 2154–2160 (2021)
9. Bacci, E., Parker, D.: Probabilistic guarantees for safe deep reinforcement learning. In: FORMATS, pp. 231–248 (2020)
10. Bacci, E., Parker, D.: Verified probabilistic policies for deep reinforcement learning. In: NFM, pp. 193–212 (2022)
11. Brain, M., Joshi, S., Kroening, D., Schrammel, P.: Safety verification and refutation by k-invariants and k-induction. In: SAS, pp. 145–161 (2015)
12. Brockman, G., et al.: OpenAI Gym (2016). arXiv:1606.01540
13. Calinescu, R., Ghezzi, C., Kwiatkowska, M., Mirandola, R.: Self-adaptive software needs quantitative verification at runtime. Commun. ACM **55**(9), 69–77 (2012)
14. Carr, S., Jansen, N., Topcu, U.: Task-aware verifiable RNN-based policies for partially observable markov decision processes. Artif. Intell. Res. **72**, 819–847 (2021)
15. Chakarov, A., Sankaranarayanan, S.: Probabilistic program analysis with martingales. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 511–526. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39799-8_34
16. Chatterjee, K., Fu, H., Novotný, P., Hasheminezhad, R.: Algorithmic analysis of qualitative and quantitative termination problems for affine probabilistic programs. In: POPL, pp. 327–342 (2016)

17. Cousot, P., Cousot, R.: Constructive versions of tarski's fixed point theorems. Pac. J. Math. **82**(1), 43–57 (1979)
18. Dawson, C., Gao, S., Fan, C.: Safe control with learned certificates: a survey of neural lyapunov, barrier, and contraction methods for robotics and control. IEEE Trans. Robot. **39**, 1749–1767 (2023)
19. Deshmukh, J., Kapinski, J., Yamaguchi, T., Prokhorov, D.: Learning deep neural network controllers for dynamical systems with safety guarantees. In: ICCAD, pp. 1–7 (2019)
20. Donaldson, A.F., Haller, L., Kroening, D., Rümmer, P.: Software verification using k-induction. In: SAS, pp. 351–368 (2011)
21. Feng, S., Chen, M., Su, H., Kaminski, B.L., Katoen, J., Zhan, N.: Lower bounds for possibly divergent probabilistic programs. Proc. ACM Program. Lang. **7**(OOPSLA1), 696–726 (2023)
22. Feng, S., Chen, M., Xue, B., Sankaranarayanan, S., Zhan, N.: Unbounded-time safety verification of stochastic differential dynamics. In: CAV, pp. 327–348 (2020)
23. Gowal, S., et al.: On the effectiveness of interval bound propagation for training verifiably robust models. CoRR arXiv: 1810.12715 (2018)
24. Gronwall, T.H.: Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. Annals Math. 292–296 (1919)
25. Hahn, E.M., et al.: The 2019 comparison of tools for the analysis of quantitative formal models: (QComp 2019 competition report). In: TACAS, pp. 69–92 (2019)
26. Hamers, R., Jongmans, S.: Discourje: Runtime verification of communication protocols in clojure. In: TACAS, pp. 266–284 (2020)
27. Hensel, C., Junges, S., Katoen, J.P., Quatmann, T., Volk, M.: The probabilistic model checker storm. Inter. J. Softw. Tools Technol. Trans. 1–22 (2021)
28. Hoeffding, W.: Probability inequalities for sums of bounded random variables. The collected works of Wassily Hoeffding, pp. 409–426 (1994)
29. Huang, C., Fan, J., Chen, X., Li, W., Zhu, Q.: Polar: a polynomial arithmetic framework for verifying neural-network controlled systems. In: ATVA, pp. 414–430 (2022)
30. Ivanov, R., Carpenter, T., Weimer, J., Alur, R., Pappas, G., Lee, I.: Verisig 2.0: Verification of neural network controllers using taylor model preconditioning. In: CAV, pp. 249–262 (2021)
31. Jin, P., Tian, J., Zhi, D., et al.: Trainify: a CEGAR-driven training and verification framework for safe deep reinforcement learning. In: CAV, pp. 193–218 (2022)
32. Kwiatkowska, M., Norman, G., Parker, D.: Prism 4.0: verification of probabilistic real-time systems. In: CAV, pp. 585–591 (2011)
33. Kwiatkowska, M., Norman, G., Parker, D.: Probabilistic model checking and autonomy. Annu. Rev. Control Robot. Auton. Syst. **5**, 385–410 (2022)
34. Lavaei, A., Soudjani, S., Frazzoli, E.: Safety barrier certificates for stochastic hybrid systems. In: ACC, pp. 880–885 (2022)
35. Lechner, M., Zikelic, D., Chatterjee, K., Henzinger, T.A.: Stability verification in stochastic control systems via neural network supermartingales. In: AAAI, pp. 7326–7336 (2022)
36. Lillicrap, T., et al.: Continuous control with deep reinforcement learning. CoRR abs/ arXiv: 1509.02971 (2015)
37. Mathiesen, F.B., Calvert, S.C., Laurenti, L.: Safety certification for stochastic systems via neural barrier functions. IEEE Control Syst. Lett. **7**, 973–978 (2022)
38. Meng, Y., Qin, Z., Fan, C.: Reactive and safe road user simulations using neural barrier certificates. In: IROS, pp. 6299–6306 (2021)

39. Murali, V., Trivedi, A., Zamani, M.: Closure certificates. In: HSCC, pp. 10:1–10:11 (2024)
40. Nadali, A., Murali, V., Trivedi, A., Zamani, M.: Neural closure certificates. In: AAAI, pp. 21446–21453 (2024)
41. Peruffo, A., Ahmed, D., Abate, A.: Automated and formal synthesis of neural barrier certificates for dynamical models. In: TACAS, pp. 370–388 (2021)
42. Prajna, S., Jadbabaie, A.: Safety verification of hybrid systems using barrier certificates. In: Alur, R., Pappas, G.J. (eds.) HSCC, pp. 477–492 (2004)
43. Prajna, S., Jadbabaie, A., Pappas, G.J.: A framework for worst-case and stochastic safety verification using barrier certificates. IEEE Trans. Automat. Contr. **52**(8), 1415–1428 (2007)
44. Prajna, S., Rantzer, A.: On the necessity of barrier certificates. IFAC Proc. Vol. **38**(1), 526–531 (2005)
45. Salamati, A., Lavaei, A., Soudjani, S., Zamani, M.: Data-driven safety verification of stochastic systems via barrier certificates. In: ADHS, pp. 7–12 (2021)
46. Samek, W., Montavon, G., Lapuschkin, S., et al.: Explaining deep neural networks and beyond: a review of methods and applications. Proc. IEEE **109**(3), 247–278 (2021)
47. Seshia, S.A., Sadigh, D., Sastry, S.S.: Toward verified artificial intelligence. Commun. ACM **65**(7), 46–55 (2022)
48. Sha, M., et al.: Synthesizing barrier certificates of neural network controlled continuous systems via approximations. In: DAC, pp. 631–636 (2021)
49. Steinhardt, J., Tedrake, R.: Finite-time regional verification of stochastic non-linear systems. Int. J. Robotics Res. **31**(7), 901–923 (2012)
50. Takisaka, T., Oyabu, Y., Urabe, N., Hasuo, I.: Ranking and repulsing supermartingales for reachability in probabilistic programs. In: ATVA, pp. 476–493 (2018)
51. Takisaka, T., Oyabu, Y., Urabe, N., Hasuo, I.: Ranking and repulsing supermartingales for reachability in randomized programs. ACM Trans. Prog. Lang. Syst. **43**(2), 5:1–5:46 (2021)
52. Tschaikowski, M., Tribastone, M.: Tackling continuous state-space explosion in a markovian process algebra. Theoret. Comput. Sci. **517**, 1–33 (2014)
53. Urabe, N., Hara, M., Hasuo, I.: Categorical liveness checking by corecursive algebras. In: LICS, pp. 1–12 (2017)
54. Ville, J.: Etude critique de la notion de collectif (1939)
55. Wan, X., Zeng, L., Sun, M.: Exploring the vulnerability of deep reinforcement learning-based emergency control for low carbon power systems. In: IJCAI, pp. 3954–3961 (2022)
56. Williams, D.: Probability with martingales. Cambridge university press (1991)
57. Winkler, T., Gehnen, C., Katoen, J.: Model checking temporal properties of recursive probabilistic programs. In: FOSSACS, pp. 449–469 (2022)
58. Xia, J., Hu, M., Chen, X., Chen, M.: Accelerated synthesis of neural network-based barrier certificates using collaborative learning. In: Proceedings of the 59th ACM/IEEE Design Automation Conference, pp. 1201–1206 (2022)
59. Xu, K., et al.: Automatic perturbation analysis for scalable certified robustness and beyond. In: NeurIPS (2020)
60. Xue, B.: A new framework for bounding reachability probabilities of continuous-time stochastic systems. CoRR abs/ arxiv: 2312.15843 (2023)
61. Xue, B., Fränzle, M., Zhan, N.: Inner-approximating reachable sets for polynomial systems with time-varying uncertainties. IEEE Trans. Autom. Control **65**(4), 1468–1483 (2020)

62. Xue, B., Li, R., Zhan, N., Fränzle, M.: Reach-avoid analysis for stochastic discrete-time systems. In: ACC, pp. 4879–4885 (2021)
63. Xue, B., Zhan, N., Fränzle, M.: Reach-avoid analysis for polynomial stochastic differential equations. IEEE Trans. Autom. Control (2023)
64. Yang, Z., et al.: An iterative scheme of safe reinforcement learning for nonlinear systems via barrier certificate generation. In: CAV, pp. 467–490 (2021)
65. Zeng, X., Yang, Z., Zhang, L., Tang, X., Zeng, Z., Liu, Z.: Safety verification of nonlinear systems with bayesian neural network controllers. In: AAAI, pp. 15278–15286 (2023)
66. Zhang, H., Gu, J., Zhang, Z., Du, L., et al.: Backdoor attacks against deep reinforcement learning based traffic signal control systems. Peer Peer Netw. Appl. **16**(1), 466–474 (2023)
67. Zhang, H., Chen, H., Boning, D.S., Hsieh, C.: Robust reinforcement learning on state observations with learned optimal adversary. In: ICLR (2021)
68. Zhang, H., et al.: Robust deep reinforcement learning against adversarial perturbations on state observations. In: NeurIPS, pp. 21024–21037 (2020)
69. Zhao, H., Qi, N., Dehbi, L., Zeng, X., Yang, Z.: Formal synthesis of neural barrier certificates for continuous systems via counterexample guided learning. ACM Trans. Embed. Comput. Syst. **22**(5s), 146:1–146:21 (2023)
70. Zhao, H., Zeng, X., Chen, T., Liu, Z.: Synthesizing barrier certificates using neural networks. In: HSCC, pp. 1–11 (2020)
71. Zhi, D., Wang, P., Liu, S., Ong, L., Zhang, M.: Unifying qualitative and quantitative safety verification of dnn-controlled systems. CoRR abs/ arXiv: 2404.01769 (2024)
72. Zikelic, D., Lechner, M., Henzinger, T.A., Chatterjee, K.: Learning control policies for stochastic systems with reach-avoid guarantees. In: AAAI, pp. 11926–11935 (2023)