Safety Verification of Nonlinear Systems with Bayesian Neural Network Controllers

Xia Zeng[1], Zhengfeng Yang[2*], Li Zhang[2], Xiaochao Tang[2], Zhenbing Zeng[3], Zhiming Liu[1]

[1]School of Computer and Information Science, Southwest University, Chongqing, China
[2]Shanghai Key Lab of Trustworthy Computing, East China Normal University, Shanghai, China
[3]Department of Mathematics, Shanghai University, Shanghai, China
xzeng0712@swu.edu.cn, zfyang@sei.ecnu.edu.cn, zbzeng@shu.edu.cn, zhimingliu88@swu.edu.cn

Abstract

Bayesian neural networks (BNNs) retain NN structures with a probability distribution placed over their weights. With the introduced uncertainties and redundancies, BNNs are proper choices of robust controllers for safety-critical control systems. This paper considers the problem of verifying the safety of nonlinear closed-loop systems with BNN controllers over unbounded-time horizon. In essence, we compute a safe weight set such that as long as the BNN controller is always applied with weights sampled from the safe weight set, the controlled system is guaranteed to be safe. We propose a novel two-phase method for the safe weight set computation. First, we construct a reference safe control set that constraints the control inputs, through polynomial approximation to the BNN controller followed by polynomial-optimization-based barrier certificate generation. Then, the computation of safe weight set is reduced to a range inclusion problem of the BNN on the system domain w.r.t. the safe control set, which can be solved incrementally and the set of safe weights can be extracted. Compared with the existing method based on invariant learning and mixed-integer linear programming, we could compute safe weight sets with larger radii on a series of linear benchmarks. Moreover, experiments on a series of widely used nonlinear control tasks show that our method can synthesize large safe weight sets with probability measure as high as 95% even for a large-scale system of dimension 7.

Introduction

Deep neural networks (DNNs) are capable of dealing with decision making tasks in a variety of areas, such as medical diagnosis, face recognition and so on. Especially, DNNs have been used successfully in control of cyber-physical systems such as unmanned aerial vehicles, self-driving cars, etc (Squires, Pierpaoli, and Egerstedt 2018). However, DNNs are still not widely applied to safety-critical systems in practice because they may behave in unexpected ways in response to inputs leaving known grounds. In fact, uncertainty is everywhere in the real-world situation including modeling uncertainty (also called epistemic uncertainty) and data uncertainty (Oliver Dürr 2020; Han, Jasour, and Williams 2022). Bayesian neural networks (BNNs) retain the structures of neural networks and place distributions over their weights (Neal 1996). This allows learning uncertainty in the data and the network's prediction, while preserving the strong modelling capabilities of DNNs (Lee et al. 2019; McAllister et al. 2017). BNNs are proper choice for controllers of safety-critical systems by introducing redundancies to tackle the safe control problem more robustly (Lee et al. 2019).

Because of the probabilistic nature of BNNs, the closed-loop systems may exhibit unsafe behaviors under the BNN controllers. So safety verification of BNN controlled systems amounts to computing a *safe weight set* from the support set of the distribution of the BNN's weights, such that when restricted to the safe weight set, the BNN controller not only retains the redundancies of control but also guarantee the safe behaviors of the systems. A similar research topic is verifying safety of the closed-loop systems under DNN controllers, which has been extensively studied (Zhao et al. 2021; Deshmukh et al. 2019; Ivanov et al. 2019; Tran et al. 2020; Yang et al. 2021). Comparably speaking, the safety verification of BNN controllers has drawn less attention except for (Lechner et al. 2021). Verifying even simple properties about DNN is NP-complete (Katz et al. 2017), and verification for BNN means at least verifying any deterministic DNN sampled from the BNN's posterior distribution (Wicker et al. 2020). So it is conceivable that the verification of closed-loop systems with BNN controllers is a big challenge, and results on verification of systems with DNN controllers cannot be employed straightforwardly.

Recently, (Lechner et al. 2021) first defined the safety verification problem for discrete-time dynamical systems with BNN controllers. They compute the safe weight set W_ϵ with width ϵ, and iteratively increase ϵ to explore a W_ϵ as large as possible, which ensures the infinite time horizon safety condition, i.e. existence the safe inductive invariants. They use supervised learning to synthesize neural network invariants as safety certificates, whose rigorousness is checked by mixed-integer linear programming (MILP) solvers. For their method to work, the original continuous dynamical systems in the safety control benchmarks need to be discretized first. In addition, the scalability and efficiency of the method is limited by MILP/SMT-solver-based inductive invariants checking. Furthermore, when seeking for larger safe weight sets, results about smaller safe weight sets and corresponding invariants cannot be reused, which makes their iterative

*Corresponding author.

synthesis framework less systematic and efficient.

In this paper, we study the problem of safety verification for nonlinear *continuous* dynamical systems with BNN controllers over unbounded-time horizon. To begin with, we introduce a concept of *safe control set* such that when all control inputs are restricted to this set then safety of the controlled system is guaranteed. We propose a novel *two-phase* method to yield safe weight sets, which starts with the construction of a safe control set by means of polynomial optimization solving method, and proceeds by identifying the weights from the BNN's posterior distribution to construct the safe weight set via range inclusion computation. In the first phase, we build a DNN with weights selected as the mean of the BNN's posterior distribution, and compute its polynomial approximation as a reference controller, and then the safe control set is obtained by maximizing a perturbation interval around the polynomial reference controller in order to capture the largest possible range of safe control actions. In the second phase, we sample weights $\mathbf{w}$ from the BNN's posterior distribution, and try to verify that the DNN with weights taken from a neighborhood of $\mathbf{w}$ maps all states in the system domain to the obtained safe control set, and then we accumulate the verified neighborhood of $\mathbf{w}$ to obtain the safe weight set incrementally.

Compared with (Lechner et al. 2021), our method has the following prominent features. First, it can directly deal with the continuous dynamical systems commonly studied for safety control problems without discretization. Second, it computes the safe control set as a reference and incrementally constructs the safe weight set by accumulating small safe weight subsets rather than treats the whole safe weight set as a hyper-rectangle, thus being able to capture finer and larger safe weight sets, which is confirmed by the experimental results. Third, on the solving efficiency and scalability, our method based on polynomial abstraction and polynomial optimization is more efficient and applies straightforwardly to nonlinear systems even with high dimension.

In summary, the main contributions of this paper are as follows: 1) We formally define the infinite-time-horizon-safety control problem for continuous dynamical systems with BNN controllers, and introduce a methodology for synthesizing the safe weight set of the BNN ensuring the safety of the controlled system; 2) We introduce the notion of reference safe control set as a bridge between the safety of the system and the safe weight set of the BNN, which enables us to construct the safe weight set incrementally by accumulating any verified safe weight subset; 3) We present a computational method to yield a reference safe control set by solving a polynomial optimization problem resulting from the barrier certificate synthesis, and develop an interval-analysis-based algorithm combined with Lipschitz constant estimation for yielding the safe weight sets; 4) We evaluate our approach on a set of benchmarks, demonstrating that it compares favorably with the existing algorithm.

Related Work

Neural network (NN) controllers have been used in control of continuous dynamical systems for long, and recently regained popularity with the prosperity of deep reinforcement learning (Duan et al. 2016; Lillicrap et al. 2016). To improve the reliability and interpretability of NN control, researchers are beginning to introduce measurement of uncertainty in control actions, and thus the adoption of BNN controllers is on its way. In particular, (McAllister et al. 2017) systematically discussed the advantages of Bayesian deep learning for different layers of the autonomous driving architecture, and strongly advocated the Bayesian decision approach; (Lee et al. 2019) used an ensemble BNN structure to compute the prediction uncertainty of end-to-end BNN controllers based on different sensors, and choose the least uncertain one as the applied control action to a safety-critical system; (Loquercio, Segu, and Scaramuzza 2020) presented a general framework for uncertainty estimation of neural network predictions based on Bayesian belief networks, and demonstrated its effectiveness on a steering angle prediction task.

With the boom of NN control, stability and safety issues of NN-controlled systems are drawing more and more attention in the control and AI community (Anand et al. 2021; Dawson, Gao, and Fan 2022; Everett 2021). Compared to the research on safe NN controllers, formal verification for pure BNN or BNN-controlled systems is much newer and there are a few works on this topic: (Wicker et al. 2020) studied the probabilistic safety for BNNs, by computing the probability of weights w.r.t. the posterior distribution using linear bound propagation, such that a given input set is mapped to a target set under these weights; (Cardelli et al. 2019) studied a similar robustness problem to (Wicker et al. 2020) for BNNs, but adopted a statistical verification approach to compute the probability of robustness safety with a prescribed error bound and confidence, by sampling from the posterior distribution; (Michelmore et al. 2020) extended the statistical verification approach to end-to-end driving with BNN controllers, which can compute both real time decision uncertainty and bounded-time trajectory safety, with statistical guarantees; (Wicker et al. 2021) computed reach-avoid probabilities for closed-loop systems with iterative BNN predictions, where BNN is used to model system dynamics rather than controllers.

Cheng et al. pointed out that in safety-critical applications such as autonomous driving, it is necessary to provide extremely low safety failure probability on the order 10^{-8} over a given planning horizon (Cheng, Murray, and Burdick 2021). Therefore it is desirable to pursue absolute safety with an infinite time horizon. To the best of our knowledge, (Lechner et al. 2021) proposed the first and only infinite time horizon safety verification approach for closed-loop system with BNN controllers. Our approach differs from it in that we consider continuous dynamical systems and propose a novel efficient framework for incremental construction of the safe weight set based on polynomial optimization.

Preliminaries and Problem Statement

Throughout this paper, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}[\mathbf{x}]$ denotes the polynomial ring with coefficients in $\mathbb{R}$ over $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$, and $\Sigma[\mathbf{x}] \subset \mathbb{R}[\mathbf{x}]$ denotes the space of Sum-of-Squares polynomials. For a random vector $\mathbf{x} \sim p(\mathbf{x})$, we use $supp(p(\mathbf{x}))$ to denote the support of the distribution $p(\mathbf{x})$ of $\mathbf{x}$. Let $\|\cdot\|$ denote the 2-norm.

Bayesian Neural Networks

A typical neural network (NN for short) consists of an input layer, an output layer, and multiple hidden layers in between. Each neuron in some hidden layer ℓ_i is assigned by a linear combination of the neuron outputs of the previous layer, and then applying a non-linear activation function ϕ, i.e., $\ell_i(\mathbf{x}) = \phi(\mathbf{w}_i\mathbf{x} + \mathbf{b}_i)$, $\mathbf{w}_i \in \mathbb{R}^{n_i \times n_{i-1}}, \mathbf{b}_i \in \mathbb{R}^{n_i}$, where n_i is the number of neurons in layer l_i. So a neural network is essentially a composite function of its layers, i.e., $\pi = \ell_k \circ \cdots \circ \ell_1$.

Despite NN is an important machine learning (ML) model, it lacks reflection of the uncertainty when leaving known grounds. The Bayesian neural network (BNN for short) models the epistemic uncertainty well by placing a probability distribution on the weights and biases of a traditional NN. For simplicity, we denote a BNN by $\pi^{\mathbf{w}} = [\ell_1^{\mathbf{w}_1,\mathbf{b}_1}, \cdots, \ell_k^{\mathbf{w}_k,\mathbf{b}_k}]$, and call $\mathbf{w}$, the vector random variable collecting all the weights and biases in the network, the *weights* of a BNN. For any sampled value w of $\mathbf{w}$, a BNN becomes a deterministic NN, denoted by π^w.

When training a BNN model, it is assumed its weights obey some prior distribution, i.e., $\mathbf{w} \sim p(\mathbf{w})$. Then learning the BNN is essentially computing the posterior distribution $q(\mathbf{w}|\mathcal{D})$ under the Bayes rule. As the function of the BNN model is highly nonlinear, computing such a conditional distribution $q(\mathbf{w}|\mathcal{D})$ is of high complexity and infeasible in general. The commonly used BNN training algorithms are based on efficient approximate inference techniques, e.g., variational inference (VI) (Blundell et al. 2015), Monte Carlo dropout (MC dropout) (Brooks et al. 2011), etc.

Problem Statement

Consider a continuous dynamical system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, where $\mathbf{f} = (f_1, \cdots, f_n)^T$ is the vector field defined on the state space $D \subset \mathbb{R}^n$. We assume that $\mathbf{f}$ satisfies the local Lipschitz condition, so that the system has a unique solution $\mathbf{x}(t, \mathbf{x}_0)$ in D for $\mathbf{x}_0 \in D$. In many contexts, a dynamical system is equipped with a domain $\Psi \subset D$ and an initial set $\Theta \subset \Psi$, represented as a triple $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$, called a constrained continuous dynamical system (CCDS).

Definition 1 (Safety) *For a CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ and a given unsafe region X_u, the system is safe if for all $\mathbf{x}_0 \in \Theta$, there does not exist $t_1 > 0$ s.t. $\forall t \in [0, t_1].\mathbf{x}(t, \mathbf{x}_0) \in \Psi$ and $\mathbf{x}(t_1, \mathbf{x}_0) \in X_u$, that is, the system's trajectory never reaches X_u from Θ as long as it remains in Ψ.*

A controlled continuous dynamical system is modeled by first-order ordinary differential equations (ODE)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad \text{with } \mathbf{u} = \mathbf{k}(\mathbf{x}) \tag{1}$$

where $\mathbf{x} \in \Psi$, $\mathbf{f}$ is a locally Lipschitz continuous vector filed, $\mathbf{u} \in U$ with $U \subset \mathbb{R}^m$ an *admissible control set*, and $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$ is a locally Lipschitz continuous controller function. In this paper, we will adopt the notion of *state-wise admissible control set*, i.e. $U_{\mathbf{x}} \subseteq \mathbb{R}^m$ defined per state $\mathbf{x}$, which is used to restrict the control input $\mathbf{u}$ such that for all $\mathbf{x} \in \Psi$, $\mathbf{u} = \mathbf{k}(\mathbf{x}) \in U_{\mathbf{x}}$.

In this paper, we investigate the safety of the closed-loop system (1) with a BNN controller $\mathbf{k}$. We focus on systems in form of ODE mainly to validate the functionality of the proposed BNN safety verification approach, which can be extended to deal with more practical stochastic systems in future. Since the output of a BNN $\pi^{\mathbf{w}}$ is probabilistic, we must clarify how to apply it to (1). To this end, we introduce the following two definitions:

Definition 2 (Continuous Instantiation of BNN) *Given a BNN $\pi^{\mathbf{w}}$ with weights $\mathbf{w} \sim q(\mathbf{w})$ and a set $W \subset supp(q(\mathbf{w}))$, a locally Lipschitz continuous function $\mathbf{w}(t)$ is called a continuous instantiation of $\pi^{\mathbf{w}}$ from W, if $\forall t \geq 0.\mathbf{w}(t) \in W$.*

Definition 3 (Continuous BNN Controller) *Given a BNN $\pi^{\mathbf{w}}$ with weights $\mathbf{w} \sim q(\mathbf{w})$ and a set $W \subset supp(q(\mathbf{w}))$, any continuous instantiation of $\pi^{\mathbf{w}}$ from W can be used as a controller in (1) in such a way that for all $t \geq 0$ and $\mathbf{x}(t) \in \Psi$, $\mathbf{u} = \pi^{\mathbf{w}(t)}(\mathbf{x}(t))$.*

We consider the safety of (1) with BNN controllers $\pi^{\mathbf{w}}$ as per Definition 3. Since $supp(q(\mathbf{w}))$ is usually an unbounded set for distributions such as Gaussian, the safety of (1) cannot be guaranteed if $\mathbf{w}$ is arbitrarily instantiated. Therefore it is necessary to capture a *safe weight set* from $supp(q(\mathbf{w}))$, which gives the following safety problem that we focus on:

Problem 1 (Safety for BNN Control) *Given a controlled CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ with $\mathbf{f}$ defined by (1), an unsafe region X_u, and a BNN controller $\pi^{\mathbf{w}}$ with $\mathbf{w} \sim q(\mathbf{w})$, determine a set $W \subset supp(q(\mathbf{w}))$, such that for any continuous instantiation $\mathbf{w}(t)$ of the BNN from the set W, the system with controller $\pi^{\mathbf{w}(t)}$ is safe as per Definition 1. Such a set W is called a safe weight set.*

In this work, we propose a method to compute a safe weight set W such that the safety property of the controlled system is guaranteed. Meanwhile, we compute the probability measure of W w.r.t $q(\mathbf{w})$ to evaluate how restrictive the safe weight set is. The higher probability measure W has, the more flexibility it gives to the implementation of the BNN controller.

Barrier Certificate for Safety Verification

The concept of *barrier certificate* plays an important role in safety verification of continuous systems. The essential idea is to use the zero level set of a differentiable function $B(\mathbf{x})$ as a barrier to separate all the reachable states from the unsafe set. The following theorem states the conditions that must be satisfied by a barrier certificate.

Theorem 1 (Prajna, Jadbabaie, and Pappas 2007) *Given a CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ and an unsafe region X_u, if there exists a real-valued differentiable function $B : \Psi \to \mathbb{R}$ satisfying the following conditions: i) $\forall \mathbf{x} \in \Theta.B(\mathbf{x}) \geq 0$; ii) $\forall \mathbf{x} \in X_u.B(\mathbf{x}) < 0$; iii) $\forall \mathbf{x} \in \Psi.\,(B(\mathbf{x}) = 0 \Rightarrow \mathcal{L}_{\mathbf{f}}B(\mathbf{x}) > 0)$, where $\mathcal{L}_{\mathbf{f}}B(\mathbf{x})$ denotes the Lie-derivative of $B(\mathbf{x})$ along the vector field $\mathbf{f}(\mathbf{x})$, i.e., $\mathcal{L}_{\mathbf{f}}B(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial B}{\partial x_i} \cdot f_i(\mathbf{x})$, then $B(\mathbf{x})$ is called a barrier certificate of $\mathcal{C}$, and the safety of $\mathcal{C}$ is guaranteed.*

For a controlled CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ with $\mathbf{f}$ defined by (1), a feedback controller $\mathbf{k}$ can be used to ensure the safety of $\mathcal{C}$, if there exists a barrier certificate for the closed-loop system

under $\mathbf{k}$. In this paper, we solve Problem 1 by synthesizing barrier certificates for $\mathcal{C}$ controlled by a BNN $\pi^{\mathbf{w}}$.

To facilitate the presentation in the rest of this paper, we will make the following two assumptions.

Assumption 1 *The BNN controller $\pi^{\mathbf{w}}$ has one output neuron, i.e., the controller $\mathbf{u}$ in (1) is a scalar.*

Assumption 2 *The controlled CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ is of polynomial form, that is, $\mathbf{f}$ is polynomial, Θ, Ψ, as well as the unsafe set X_u, are all semi-algebraic sets defined by polynomial equations and inequalities, represented by $\Theta := \{\mathbf{x} \in \mathbb{R}^n \mid q_i(\mathbf{x}) \geq 0,\ 1 \leq i \leq \kappa\}$,$\Psi := \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \geq 0,\ 1 \leq j \leq \iota\}$, $X_u := \{\mathbf{x} \in \mathbb{R}^n \mid g_k(\mathbf{x}) \geq 0,\ 1 \leq k \leq \nu\}$, where $q_i(\mathbf{x})$, $h_j(\mathbf{x})$ and $g_k(\mathbf{x})$ are polynomials.*

We remark that the proposed approach is not limited by these assumptions. For non-polynomial systems, we can abstract the non-polynomial term in the system specification into a polynomial inclusion by means of the Taylor model (Chen, Ábrahám, and Sankaranarayanan 2012).

Main Results

In this section, we first introduce a proposition that lays the foundation of our method and then present the detailed design of our method for safe weight set computation. All proofs and algorithm details can be found at https://github.com/Estellaly/Safety-Verification-of-Bnn-Controllers.

Framework of the Safe Weight Set Synthesis

Our idea is to first construct a state-wise admissible control set $U_{\mathbf{x}}$ for each $\mathbf{x} \in \Psi$ such that the system is guaranteed to be safe when the range of $\mathbf{k}(\mathbf{x})$ is restricted to $U_{\mathbf{x}}$. We next try to extract a set W of weights such that for any continuous instantiation $\pi^{\mathbf{w}(t)}$ from W, and for any $\mathbf{x} \in \Psi$, $\pi^{\mathbf{w}(t)}(\mathbf{x}) \in U_{\mathbf{x}}$. Then such W is a safe weight set by definition. To this end, we formally introduce the following definition and propositions.

Definition 4 (Safe Control Set) *For a CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ with $\mathbf{f}$ defined by (1) and a given unsafe region X_u, a state-wise admissible control set $U_{\mathbf{x}}$ is called a safe control set if the closed-loop system under any continuous controller $\mathbf{k}(\mathbf{x})$ s.t. $\mathbf{k}(\mathbf{x}) \in U_{\mathbf{x}}, \forall \mathbf{x} \in \Psi$ is guaranteed to be safe.*

Proposition 1 *For a controlled CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ with $\mathbf{f}$ defined by (1), if there exists a state-wise admissible control set $U_{\mathbf{x}}$ such that there exists a uniform barrier certificate for the closed-loop system with any controller $\mathbf{k}$ satisfying $\forall \mathbf{x} \in \Psi.\mathbf{k}(\mathbf{x}) \in U_{\mathbf{x}}$, then $U_{\mathbf{x}}$ is a safe control set.*

Proposition 2 *Given a controlled CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$, a BNN controller $\pi^{\mathbf{w}}$ with $\mathbf{w} \sim q(\mathbf{w})$, and a safe control set $U_{\mathbf{x}}$, a set $W \subset supp(q(\mathbf{w}))$ is a safe weight set if $\pi^W(\mathbf{x}) \subset U_{\mathbf{x}}$ for all $\mathbf{x} \in \Psi$, where $\pi^W(\mathbf{x}) = \{\pi^{\mathbf{w}}(\mathbf{x}) \mid \mathbf{w} \in W\}$.*

Based on Propositions 1 and 2, our safe weight set construction method consists of two main steps, i.e. safe control set construction and safe weight set extraction, as illustrated by the framework in Fig. 1. The box in the bottom left of Fig. 1 shows the safe control set construction process. We construct the safe control set $U_{\mathbf{x}}$ as a polynomial interval model. Briefly, we use π^{μ} with μ the mean weight of the BNN's posterior as a reference controller, and compute a polynomial approximation $p(\mathbf{x})$ of π^{μ} for tractability. Then $U_{\mathbf{x}}$ is expressed as $p(\mathbf{x})$ plus an interval $[-\gamma, \gamma]$, with γ computed through polynomial optimization which simultaneously ensures the existence of barrier certificates for the safety of $p(\mathbf{x}) + [-\gamma, \gamma]$. The right part of Fig. 1 corresponds to the safe weight set extraction process. Briefly, we sample w' from the BNN's posterior and construct a neighborhood around w' as a candidate safe weight subset, and then we check the validity of such a candidate by solving a range inclusion problem for the BNN $\pi^{\mathbf{w}}$ on the system domain Ψ w.r.t. the safe control set $p(\mathbf{x}) + [-\gamma, \gamma]$. We next present the detailed procedures.

Safe Control Set Construction

Given a controlled CCDS $\mathcal{C} = (\mathbf{f}, \Theta, \Psi)$ with $\mathbf{f}$ defined by (1), a given unsafe region X_u, and a BNN controller $\pi^{\mathbf{w}(t)}$ with $\mathbf{w}(t) \sim q(\mathbf{w})$, we synthesize a safe control set using the concept of barrier certificates in this subsection.

Polynomial approximation for BNN controllers Let π be the BNN with a single output, and the case with multiple outputs can be dealt with similarly. Let μ be the mean of $q(\mathbf{w})$, and $\pi^{\mu}(\mathbf{x})$ corresponds to a deterministic neural network. We consider computing a polynomial $p(\mathbf{x})$ to fit $\pi^{\mu}(\mathbf{x})$ by sampling-based method. At first, we construct a template polynomial $p(\mathbf{x}|\theta)$ with dimension n, which is consistent with the system dimension, and a pre-assigned total degree τ. Let $[\mathbf{x}]_\tau$ be the vector consisting of monomials with degree at most τ, and the dimension of $[\mathbf{x}]_\tau$ is $\binom{n+\tau}{\tau}$. Then the polynomial template is written as $p(\mathbf{x}|\theta) = \theta^T[\mathbf{x}]_\tau$. Next, we construct a dataset $D = \{\mathbf{x}_i\}_{i=1}^N$ by collecting the sampling points on the trajectories of the system under the controller π^{μ} to ensure good quality of the data. Then the problem is reduced to fitting the polynomial $p(\mathbf{x}|\theta)$ for $\pi^{\mu}(\mathbf{x})$ on the dataset $D = \{\mathbf{x}_i\}_{i=1}^N$. Solving for the coefficient parameter θ is through the least square programming: $\theta^* = \arg\min_\theta \frac{1}{N}\sum_{i=1}^N (\pi^{\mu}(\mathbf{x}) - p(\mathbf{x}_i|\theta))^2$. Thus, we could obtain the polynomial approximation $p(\mathbf{x})$ for the controller $\pi^{\mu}(\mathbf{x})$.

Maximal safe control radius around $p(\mathbf{x})$ The remaining task is to find a safe radius around the polynomial $p(\mathbf{x})$ and construct the safe control set. We introduce a polynomial interval model $\mathcal{I}(\mathbf{x}) = p(\mathbf{x}) + [-\gamma, \gamma]$ with a parameter $\gamma \geq 0$, and try to make $\mathcal{I}(\mathbf{x})$ a safe control set as per Definition 4. For the model $\mathcal{I}(\mathbf{x})$, the state-wise admissible control set is given by $U_{\mathbf{x}} = \mathcal{I}(\mathbf{x}) = [p(\mathbf{x}) - \gamma, p(\mathbf{x}) + \gamma]$. We employ barrier certificates to compute a proper γ and establish the safety for the closed-loop system under all controllers $\mathbf{k}(\mathbf{x})$ satisfying $\forall \mathbf{x} \in \Psi, \mathbf{k}(\mathbf{x}) \in [p(\mathbf{x}) - \gamma, p(\mathbf{x}) + \gamma]$. In fact, any such $\mathbf{k}(\mathbf{x})$ can be expressed in the form $p(\mathbf{x}(t)) + \epsilon_t$ with ϵ_t continuously varying within $[-\gamma, \gamma]$ for $t \geq 0$. Then we try to synthesize a barrier certificate for the closed-loop system with polynomial controller $p(\mathbf{x}) + \epsilon$ with $\epsilon \in [-\gamma, \gamma]$ by abstracting t away for simplification. We assume a polynomial barrier certificate template $B(\mathbf{x}|\mathbf{b}) = \mathbf{b}^T[\mathbf{x}]_d$ of degree d in $\mathbf{x}$, with $\mathbf{b}$ a vector of dimension $\binom{n+d}{d}$ and the canonical

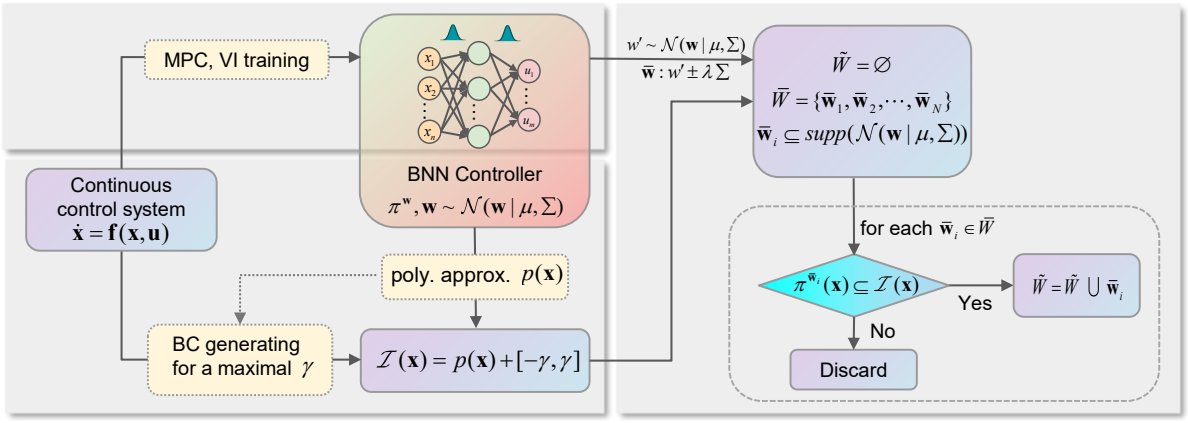

Figure 1: The framework of safe weight set generation for BNN controllers

basis $[\mathbf{x}]_d$ of monomials. In order to obtain a largest possible safe control set, we maximize γ while imposing the constraints for the existence of barrier certificate $B(\mathbf{x}|\mathbf{b})$ on the closed-loop system with controller $p(\mathbf{x}) + \epsilon$, which results in the following optimization problem:

$$\left.\begin{array}{l} \gamma^*_{opt} = \max\limits_{\mathbf{b},\gamma} \gamma \\ \text{s.t. } B(\mathbf{x}|\mathbf{b}) \geq 0, \forall \mathbf{x} \in \Theta, \\ \quad B(\mathbf{x}|\mathbf{b}) = 0 \implies \mathcal{L}_{\mathbf{f}(\mathbf{x},p(\mathbf{x})+\epsilon)} B(\mathbf{x}|\mathbf{b}) > 0, \\ \qquad\qquad \forall \mathbf{x} \in \Psi \wedge \forall \epsilon \in [-\gamma, \gamma], \\ \quad B(\mathbf{x}|\mathbf{b}) < 0, \forall \mathbf{x} \in X_u. \end{array}\right\} \quad (2)$$

Then, the Sum-of-Squares (SOS) relaxation technique (Papachristodoulou et al. 2021) is applied to encode the optimization problem (2) as an SOS program. We have encoded the constraint $\forall \epsilon \in [-\gamma, \gamma]$ for ϵ involved in the polynomial $\mathcal{L}_{\mathbf{f}(\mathbf{x},p(\mathbf{x})+\epsilon)} B(\mathbf{x}|\mathbf{b})$ into $\hat{h}(\gamma, \epsilon) \geq 0$ with $\hat{h}(\gamma, \epsilon) := (\gamma + \epsilon)(\gamma - \epsilon)$. Thus, the problem (2) can be transformed into the following optimization problem

$$\left.\begin{array}{l} \gamma^* = \max\limits_{\mathbf{b},\gamma} \gamma \\ \text{s.t. } B(\mathbf{x}|\mathbf{b}) - \sum_j \hat{\sigma}_j(\mathbf{x}) q_j(\mathbf{x}) \in \Sigma[\mathbf{x}], \\ \quad \mathcal{L}_{\mathbf{f}(\mathbf{x},\epsilon)} B(\mathbf{x}|\mathbf{b}) - \lambda(\mathbf{x}) B(\mathbf{x}|\mathbf{b}) - \sum_j \rho_j(\mathbf{x}) h_j(\mathbf{x}) \\ \qquad\qquad - \xi(\mathbf{x}, \epsilon) \hat{h}(\gamma, \epsilon) - \epsilon_1 \in \Sigma[\mathbf{x}], \\ \quad -B(\mathbf{x}|\mathbf{b}) - \epsilon_2 - \sum_j \kappa_j(\mathbf{x}) g_j(\mathbf{x}) \in \Sigma[\mathbf{x}], \end{array}\right\} \quad (3)$$

where $\epsilon_1, \epsilon_2 > 0$, the entries of $\hat{\sigma}_j(\mathbf{x})$, $\rho_j(\mathbf{x})$, $\kappa_j(\mathbf{x}) \in \Sigma[\mathbf{x}]$, $\xi(\mathbf{x}, \epsilon) \in \Sigma[\mathbf{x}, \epsilon]$, and $\lambda(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$. Clearly, the feasibility of the constraints in (3) is sufficient to imply the feasibility of the constraints in (2), and thus the optimum of (3) is a lower bound of the optimum of (2), i.e., $\gamma^* \leq \gamma^*_{opt}$. Therefore, the safe control set generation problem is transformed into a non-convex bilinear matrix inequalities (BMI) problem (3), which can be solved by a Matlab package PENBMI (Kocvara, Stingl, and GbR 2005). If (3) is feasible then the solution $\mathbf{b}^*$ and optimum γ^* result in a barrier certificate $B(\mathbf{x}|\mathbf{b}^*)$ for all controllers $p(\mathbf{x}) + \epsilon$ with $\epsilon \in [-\gamma^*, \gamma^*]$. Furthermore, we have

Proposition 3 *$\mathcal{I}(\mathbf{x}) = p(\mathbf{x}) + [-\gamma^*, \gamma^*]$ with γ^* produced by (3) is a safe control set.*

Safe Weight Set Construction for BNN

We use the computed safe control set $\mathcal{I}(\mathbf{x}) = p(\mathbf{x}) + [-\gamma^*, \gamma^*]$ to extract a safe weight set according to Proposition 2. The computation of the exact safe weight set W is not straightforward. We provide a method to compute a verified subset $\tilde{W}$ of the exact set, and meanwhile calculate the probability measure of $\tilde{W}$ w.r.t. the BNN's posterior distribution.

The intuitive idea can be illustrated by Fig. 2. For a BNN $\pi^{\mathbf{w}}$ with $\mathbf{w} \sim \mathcal{N}(\mathbf{w}|\mu, \Sigma)$ obtained by VI, we build small safe weight subset candidates which are hyper-rectangles centered at sampled points from $\mathcal{N}(\mathbf{w}|\mu, \Sigma)$, i.e., $\bar{\mathbf{w}}_i = [w'_i - \lambda\Sigma, w'_i + \lambda\Sigma]$ with $w'_i \sim supp(\mathcal{N}(\mathbf{w}|\mu, \Sigma))$. Next, according to Proposition 2, we need to check that for each candidate $\bar{\mathbf{w}}_i$ the inclusion condition $\forall \mathbf{x} \in \Psi . \pi^{\bar{\mathbf{w}}_i}(\mathbf{x}) \subset \mathcal{I}(\mathbf{x})$ holds, which can be reduced to checking the validity of $\forall \mathbf{x} \in \bar{\mathbf{x}}_j . \pi^{\bar{\mathbf{w}}_i}(\mathbf{x}) \subset \mathcal{I}(\mathbf{x}), \forall j = 1, \cdots, s$, where $\{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots, \bar{\mathbf{x}}_s\}$ is a partition of Ψ. Finally, $\tilde{W}$ is taken as the union of all verified candidates $\bar{\mathbf{w}}_i$.

Range inclusion for BNN Let χ be an arbitrary partition block of Ψ and $\bar{\mathbf{w}}$ be a safe weight subset candidate. From the above discussion, we need to verify $\forall \mathbf{x} \in \chi . \pi^{\bar{\mathbf{w}}}(\mathbf{x}) \subset \mathcal{I}(\mathbf{x})$. Here we present an efficient way to check this inclusion using Lipschitz constant estimation. The local Lipschitz constant of a function φ w.r.t. a nominal input $\mathbf{x}_0 \in \chi$ is the minimal $L_\varphi > 0$ s.t. $\|\varphi(\mathbf{x}) - \varphi(\mathbf{x}_0)\| \leq L_\varphi \|\mathbf{x} - \mathbf{x}_0\|, \forall \mathbf{x} \in \chi$, denoted as $L_\varphi(\mathbf{x}_0, \chi)$. Then, the following proposition can be used to verify the desired inclusion relation.

Proposition 4 *Let $\phi^{\mathbf{w}}(\mathbf{x}) = \pi^{\mathbf{w}}(\mathbf{x}) - p(\mathbf{x})$. For a nominal input $\mathbf{x}_0 \in \chi$ we denote the local Lipschitz constants for $\pi^{\mathbf{w}}(\mathbf{x})$ and $p(\mathbf{x})$ as $L^{\mathbf{w}}_\pi(\mathbf{x}_0, \chi)$ and $L_p(\mathbf{x}_0, \chi)$ respectively. Further, let $L = \max_{\mathbf{w} \in \bar{\mathbf{w}}} L^{\mathbf{w}}_\pi(\mathbf{x}_0, \chi) + L_p(\mathbf{x}_0, \chi)$. Then if $\max_{\mathbf{w} \in \bar{\mathbf{w}}} \|\phi^{\mathbf{w}}(\mathbf{x}_0)\| + L\|\mathbf{x} - \mathbf{x}_0\| \leq \gamma^*$ for all $\mathbf{x} \in \chi$, we have $\forall \mathbf{x} \in \chi . \pi^{\bar{\mathbf{w}}}(\mathbf{x}) \subset \mathcal{I}(\mathbf{x})$.*

Then we are ready to present the safe weight set generation algorithm for BNNs (SWB for short) in Algorithm 1. Line 9 calls a subroutine *local_L()* to compute the Lipschitz constants as specified by Proposition 4. The estimation of Lipschitz constants for DNNs has been addressed in (Fazlyab et al. 2019; Ruan, Huang, and Kwiatkowska 2018;

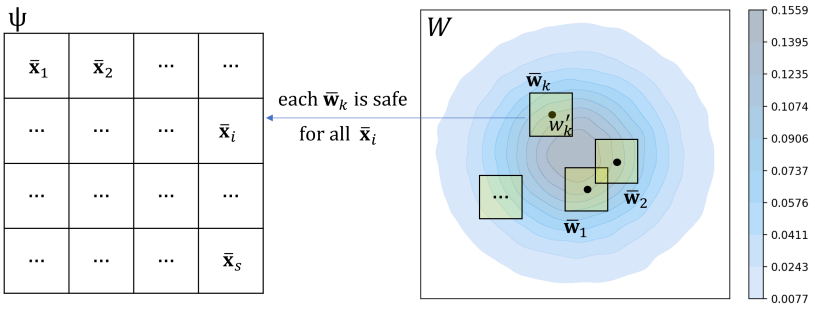

Figure 2: Safe weight set construction based on partition. The right part depicts small subsets $\bar{\mathbf{w}}_i$ by sampling $w'_i \sim \mathcal{N}(\mathbf{w}|\mu, \Sigma)$, and the left depicts the partition of system domain Ψ.

Algorithm 1: Safe weight set construction for BNN controllers (SWB)

Input: system domain Ψ, BNN policy $\pi^{\mathbf{w}}$, weight posterior $\mathbf{w} \sim \mathcal{N}(\cdot|\mu, \Sigma)$, safe control set $[p-\gamma^*, p+\gamma^*]$, number of partitions s, number of samples N, $\lambda \geq 0$, partition size δ;
Output: safe weight set $\tilde{W}$; probability measure p_{safe};

1: split Ψ into $\{\bar{\mathbf{x}}_1, ..., \bar{\mathbf{x}}_s\}$ with partition size δ;
2: $\tilde{W} \leftarrow \emptyset$;
3: **for** $k = 1 \rightarrow N$ **do**
4: $\quad w' \sim \mathcal{N}(.|\mu, \Sigma)$; $\quad \bar{\mathbf{w}} \leftarrow [w' - \lambda\Sigma, w' + \lambda\Sigma]$;
5: $\quad isSafe \leftarrow True$;
6: $\quad$ **for** $i = 1 \rightarrow s$ **do**
7: $\quad\quad \mathbf{x}_i \leftarrow (\bar{\mathbf{x}}_i^L + \bar{\mathbf{x}}_i^U)/2$;
8: $\quad\quad \|\phi(\mathbf{x}_i)\| \leftarrow \max_{\mathbf{w} \in \bar{\mathbf{w}}} \|\pi^{\mathbf{w}}(\mathbf{x}_i) - p(\mathbf{x}_i)\|$;
9: $\quad\quad L \leftarrow local\_L(\mathbf{x}_i, \delta, \bar{\mathbf{w}}, p)$;
10: $\quad\quad$ **if** $\|\phi(\mathbf{x}_i)\| + L\sqrt{n_{\mathbf{x}}}\delta > \gamma^*$ **then**
11: $\quad\quad\quad isSafe \leftarrow False$;
12: $\quad\quad\quad$ **break**;
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad$ **if** *isSafe* **then**
16: $\quad\quad \tilde{W} \leftarrow \tilde{W} \cup \bar{\mathbf{w}}$;
17: $\quad$ **end if**
18: **end for**
19: $\tilde{W} \leftarrow removeDuplicates(\tilde{W})$;
20: $p_{safe} \leftarrow computeProbability(\tilde{W})$;
21: **return** $\tilde{W}, p_{safe}$;

Szegedy et al. 2014). Here we extend the result in (Avant and Morgansen 2021) to Lipschitz constant estimation for BNNs. The $n_{\mathbf{x}}$ in Line 10 denotes the system dimension. Line 19 is based on (Wicker et al. 2020). In Line 20, we use the error function (erf) (Andrews 1998) to compute the returned probability measure. The correctness of Algorithm 1 can be stated as the following theorem.

Theorem 2 *The $\tilde{W}$ computed by Algorithm 1 is a safe weight set as per Problem 1.*

Experiments

In this part, we evaluate our approach on the piecewise-linear benchmarks from (Lechner et al. 2021) and make a comparison with the results reported therein. Furthermore, we demonstrate the effectiveness of our approach on some nonlinear benchmarks. All our experiments are carried out on a virtual machine equipped with a 64GB RAM, an Intel(R) Core(TM) i9-10900K CPU, and an NVIDIA GeForce RTX 3090 super GPU. More details of the experiments can be found at https://github.com/Estellaly/Safety-Verification-of-Bnn-Controllers.

For each safe control task, the BNN controller is obtained as follows. For benchmarks from (Lechner et al. 2021), we use their publicly available BNNs at Github. For the nonlinear benchmarks, we train BNN controllers in two steps: firstly, we apply Model Predictive Control (MPC) to generate the MPC controller and simulate the systems' trajectories to construct a dataset of state-action pairs; subsequently, Variational Inference is applied to train a BNN controller consisting of a hidden layer with 16 ReLU units with Gaussian distribution prior on all its weights, utilizing the data generated by MPC. Actually, our method doesn't rely on the MPC controller synthesis and the BNN training, which only provides BNN inputs to our algorithm.

For each BNN controller, we first compute the safe control set in the form of $p(\mathbf{x}) + [-\gamma, \gamma]$, where $p(\mathbf{x})$ is computed through a least square programming and γ is computed with barrier certificate generation via an encoded SOS relaxation programming using the Matlab package PENBMI (Kocvara, Stingl, and GbR 2005). Then we apply Algorithm 1, named SWB, to compute the safe weight set $\tilde{W}$ and its probability measure, denoted by P_{SWB}. Since the construction by Algorithm 1 is incremental, our generated $\tilde{W}$ consists of a series of disjoint high-dimensional intervals. It is statistically reasonable to find one such interval, denoted by S_μ, that contains the mean weight of the BNN's posterior, because the safe control set is computed around a polynomial approximation $p(\mathbf{x})$ to the posterior's mean, and the sampled weights (up to 5000 points) in Algorithm 1 also have largest probability density around the posterior's mean.

Table 1 shows the comparison results between our approach (SWB) and the *Bootstrapping* method in (Lechner et al. 2021). We conduct experiments on three benchmarks provided in (Lechner et al. 2021). For each benchmark, we verify two BNN controllers reported in (Lechner et al. 2021): one with Bayesian weights from the second layer

Benchmark	γ	deg $p(\mathbf{x})$	deg $B(\mathbf{x})$	verified radius		probability		
				r_{S_μ}	r_{B}	P_{SWB}	P_{S_μ}	P_{B}
Unstable LDS	0.5	2	2	2.0σ	2σ	0.93	0.45	0.45
Unstable LDS (all)	0.4	2	2	2.3σ	0.5σ	0.90	0.12	3.6e-41
Pendulum	0.8	3	2	2.4σ	2σ	0.96	0.76	0.45
Pendulum (all)	1.8	3	2	2.4σ	1.5σ	0.82	0.20	9.0e-7
Collision avoid.	0.2	2	2	3.4σ	2.0σ	0.99	0.99	0.09
Collision avoid. (all)	0.5	2	2	3.0σ	1.5σ	0.95	0.71	7.0e-11

Table 1: Comparison between our approach and (Lechner et al. 2021) on the same benchmarks

Benchmark	$n_{\mathbf{x}}$	γ	deg $p(\mathbf{x})$	deg $B(\mathbf{x})$	r_{S_μ}	P_{S_μ}	P_{SWB}
Dubin's Car (Zhao et al. 2021)	2	0.3	2	2	2.6σ	0.54	0.96
Oscillator (Zhu et al. 2019)	2	0.1	2	2	3.0σ	0.84	0.98
Academic 3D (Deshmukh et al. 2019)	3	1.6	2	4	2.7σ	0.57	0.96
Bicycle Steering (Deshmukh et al. 2019)	3	0.7	2	2	3.0σ	0.80	0.96
Chesi 3 (Andrews 1998)	4	0.1	2	2	2.9σ	0.70	0.96
LALO20 (Laub and Loomis 1998)	7	0.1	2	2	2.9σ	0.58	0.95

Table 2: Performance of our approach on a series of nonlinear benchmarks

(with $\mathcal{N}(0, 0.1)$ prior) and one with Bayesian weights in all layers (with $\mathcal{N}(0, 0.05)$ prior) which has been marked by 'all' in the Table 1. deg $p(\mathbf{x})$ refers to the degree of the computed polynomial $p(\mathbf{x})$ for approximating the controller instantiated at the posterior's mean μ, γ denotes the computed maximal radius for the safe control set, and deg $B(\mathbf{x})$ refers to the degree of the generated barrier certificate.

The comparison in Table 1 is made in two aspects, i.e. the interval radius as well as the probability measure of the generated safe weight set $\tilde{W}$. Since the computed $\tilde{W}$ is not a connected set, we extract a connected interval $S_\mu \subset \tilde{W}$ containing the posterior's mean μ and denote its radius as r_{S_μ}. r_{B} denotes the radius of verified safe weight set reported in (Lechner et al. 2021). σ means standard deviation of the weights' posterior distribution. P_{SWB}, P_{S_μ} and P_{B} represent the probability measure of our obtained safe weight set $\tilde{W}$, the extracted interval S_μ, and the safe weight set by Bootstrapping in (Lechner et al. 2021), respectively. We can see that all our verified safe weight subsets S_μ are not narrower than the $\tilde{W}$ of (Lechner et al. 2021) by comparison between r_{S_μ} and r_{B}. For benchmark *Collision avoid. (all)*, our verified safe weight subset S_μ has a radius twice as large as (Lechner et al. 2021). Especially for *Unstable LDS (all)*, our verified subset has width even more than four times as large as (Lechner et al. 2021). By comparison between P_{SWB} and P_{B}, our verified safe weight set $\tilde{W}$ has consistently higher probability measure. The least result for P_{SWB} is 82% whereas the greatest for P_{B} is 45%. For *Unstable LDS (all)*, the reported safe weight set in (Lechner et al. 2021) has an extremely small probability measure of $3.6 \cdot 10^{-41}$ due to the exponential decreasing rate of probability w.r.t. the number of BNN's weights for small safe sets. Thus we can conclude that our method can synthesize larger safe sets for guaranteeing safety while maintaining most of the expressiveness of the original BNNs.

To demonstrate the scalability of our method, we further conduct evaluation on a series of nonlinear control systems and the results are shown in Table 2. The origins of these 6 examples are provided in the first column, $n_{\mathbf{x}}$ denotes the number of state variables, and the meanings of the other columns are the same as Table 1. Our computed safe weight sets have probability measure at least 95% even for a large-scale nonlinear system with dimension 7, which shows the good scalability of our approach.

Conclusion

In this paper, we study the safety verification problem of nonlinear control systems with BNN controllers, and present a methodology for computing a safe weight set such that as long as the BNN controller is always applied with weights sampled from the safe weight set, the controlled system is guaranteed to be safe. We construct a safe control set as a bridge to help synthesize the safe weight set. Based on the notion of barrier certificate for continuous system safety verification, we formalize the safe control set generation problem into a polynomial optimization problem which is efficient to solve. We present an incremental construction of safe weight set with reference to the safe control set, and design an algorithm with a BNN range inclusion verification module based on local Lipschitz constant estimation for BNNs. Extensive experiment results consistently demonstrate the effectiveness and scalability of our approach.

Despite that our computed safe weight sets have been verified with high probability measures, the efficiency of the safe weight set construction process needs to be improved because of the complexity caused by state space division. How to find a valid division for a trade-off between the success rate of verification and the time efficiency is an interesting topic for our future work.

Acknowledgments

This work was supported in part by the National Key Research and Development Project, China under Grant 2022YFA1005100, in part by the National Natural Science Foundation of China under Grants (No. 61902325, No. 12171159, No. 62272397, No. 62032019, No. 61732019), Shanghai Trusted Industry Internet Software Collaborative Innovation Center, "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (Grant No. 22510750100), and the Capacity Development Grant of Southwest University (No. SWU116007).

References

Anand, A.; Seel, K.; Gjærum, V.; Håkansson, A.; Robinson, H.; and Saad, A. 2021. Safe Learning for Control using Control Lyapunov Functions and Control Barrier Functions: A Review. *Procedia Computer Science*, 192: 3987–3997.

Andrews, L. C. 1998. *Special functions of mathematics for engineers*. Oxford University Press.

Avant, T.; and Morgansen, K. A. 2021. Analytical bounds on the local Lipschitz constants of ReLU networks. *CoRR*, abs/2104.14672.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight Uncertainty in Neural Networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, 1613–1622. JMLR.org.

Brooks, S.; Gelman, A.; Jones, G. L.; and Meng, X.-L., eds. 2011. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC.

Cardelli, L.; Kwiatkowska, M.; Laurenti, L.; Paoletti, N.; Patane, A.; and Wicker, M. 2019. Statistical Guarantees for the Robustness of Bayesian Neural Networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5693–5700. International Joint Conferences on Artificial Intelligence Organization.

Chen, X.; Ábrahám, E.; and Sankaranarayanan, S. 2012. Taylor Model Flowpipe Construction for Non-linear Hybrid Systems. In *2012 IEEE 33rd Real-Time Systems Symposium*, 183–192.

Cheng, R.; Murray, R. M.; and Burdick, J. W. 2021. Limits of Probabilistic Safety Guarantees when Considering Human Uncertainty. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 3182–3189.

Dawson, C.; Gao, S.; and Fan, C. 2022. Safe Control with Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction methods. *ArXiv*, abs/2202.11762.

Deshmukh, J. V.; Kapinski, J. P.; Yamaguchi, T.; and Prokhorov, D. 2019. Learning deep neural network controllers for dynamical systems with safety guarantees. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 1–7. IEEE.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking Deep Reinforcement Learning for Continuous Control. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 1329–1338. JMLR.org.

Everett, M. 2021. Neural Network Verification in Control. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 6326–6340.

Fazlyab, M.; Robey, A.; Hassani, H.; Morari, M.; and Pappas, G. 2019. Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Han, W.; Jasour, A.; and Williams, B. 2022. Non-Gaussian Risk Bounded Trajectory Optimization for Stochastic Nonlinear Systems in Uncertain Environments. In *2022 International Conference on Robotics and Automation (ICRA)*, 11044–11050.

Ivanov, R.; Weimer, J.; Alur, R.; Pappas, G. J.; and Lee, I. 2019. Verisig: verifying safety properties of hybrid systems with neural network controllers. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2019.*, 169–178.

Katz, G.; Barrett, C.; Dill, D. L.; Julian, K.; and Kochenderfer, M. J. 2017. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In Majumdar, R.; and Kunčak, V., eds., *Computer Aided Verification*, 97–117. Cham: Springer International Publishing.

Kocvara, M.; Stingl, M.; and GbR, P. 2005. PENBMI user's guide (version 2.0). *software manual, PENOPT GbR, Hauptstrasse A*, 31: 91338.

Laub, M. T.; and Loomis, W. F. 1998. A molecular network that produces spontaneous oscillations in excitable cells of Dictyostelium. *Molecular biology of the cell*, 9(12): 3521–3532.

Lechner, M.; Žikelić, D. o. e.; Chatterjee, K.; and Henzinger, T. 2021. Infinite Time Horizon Safety of Bayesian Neural Networks. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 10171–10185. Curran Associates, Inc.

Lee, K.; Wang, Z.; Vlahov, B.; Brar, H.; and Theodorou, E. A. 2019. Ensemble Bayesian Decision Making with Redundant Deep Perceptual Control Policies. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 831–837.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Loquercio, A.; Segu, M.; and Scaramuzza, D. 2020. A General Framework for Uncertainty Estimation in Deep Learning. *IEEE Robotics and Automation Letters*, 5(2): 3153–3160.

McAllister, R.; Gal, Y.; Kendall, A.; Van Der Wilk, M.; Shah, A.; Cipolla, R.; and Weller, A. 2017. Concrete

Problems for Autonomous Vehicle Safety: Advantages of Bayesian Deep Learning. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, 4745–4753. AAAI Press.

Michelmore, R.; Wicker, M.; Laurenti, L.; Cardelli, L.; Gal, Y.; and Kwiatkowska, M. 2020. Uncertainty Quantification with Statistical Guarantees in End-to-End Autonomous Driving Control. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 7344–7350.

Neal, R. M. 1996. *Bayesian Learning for Neural Networks*. Berlin, Heidelberg: Springer-Verlag.

Oliver Dürr, E. M., Beate Sick. 2020. *Probabilistic Deep Learning*. Manning Publications Co. Press.

Papachristodoulou, A.; Anderson, J.; Valmórbida, G.; Prajna, S.; Seiler, P. J.; and Parrilo, P. A. 2021. SOSTOOLS Version 4.00 Sum of Squares Optimization Toolbox for MATLAB. *ArXiv*, abs/1310.4716.

Prajna, S.; Jadbabaie, A.; and Pappas, G. J. 2007. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8): 1415–1429.

Ruan, W.; Huang, X.; and Kwiatkowska, M. 2018. Reachability Analysis of Deep Neural Networks with Provable Guarantees. In Lang, J., ed., *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2651–2659. ijcai.org.

Squires, E.; Pierpaoli, P.; and Egerstedt, M. 2018. Constructive Barrier Certificates with Applications to Fixed-Wing Aircraft Collision Avoidance. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 1656–1661.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Tran, H.-D.; Yang, X.; Manzanas Lopez, D.; Musau, P.; Nguyen, L. V.; Xiang, W.; Bak, S.; and Johnson, T. T. 2020. NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems. In *Computer Aided Verification*, 3–17. Springer International Publishing.

Wicker, M.; Laurenti, L.; Patane, A.; and Kwiatkowska, M. 2020. Probabilistic Safety for Bayesian Neural Networks. In Peters, J.; and Sontag, D., eds., *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, 1198–1207. PMLR.

Wicker, M.; Laurenti, L.; Patane, A.; Paoletti, N.; Abate, A.; and Kwiatkowska, M. 2021. Certification of iterative predictions in Bayesian neural networks. In de Campos, C.; and Maathuis, M. H., eds., *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, 1713–1723. PMLR.

Yang, Z.; Zhang, Y.; Lin, W.; Zeng, X.; Tang, X.; Zeng, Z.; and Liu, Z. 2021. An Iterative Scheme of Safe Reinforcement Learning for Nonlinear Systems via Barrier Certificate Generation. In *Computer Aided Verification: 33rd International Conference, CAV 2021, Virtual Event, July 20–23, 2021, Proceedings, Part I*, 467–490. Berlin, Heidelberg: Springer-Verlag.

Zhao, H.; Zeng, X.; Chen, T.; Liu, Z.; and Woodcock, J. 2021. Learning safe neural network controllers with barrier certificates. *Formal Aspects of Computing*, 33(3): 437–455.

Zhu, H.; Xiong, Z.; Magill, S.; and Jagannathan, S. 2019. An inductive synthesis framework for verifiable reinforcement learning. *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*.