mathematics

MDPI

Article

A Learner-Refiner Framework for Barrier Certificate Generation

Deng Chen, Wang Lin * and Zuohua Ding

School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; chendnb@163.com (D.C.); zuohuading@zstu.edu.cn (Z.D.)
* Correspondence: linwang@zstu.edu.cn

Abstract: Barrier certificate is a powerful tool for verifying they safety property of dynamical systems. In this paper, we introduce an innovative learner–refiner framework for synthesizing polynomial barrier certificates. The framework comprises a *learner* and a *refiner*, which work inductively to generate barrier certificates. More specifically, the *learner* trains barrier certificate candidates represented by feedforward neural networks with polynomial activations, while the *refiner* utilizes sums of squares (SOS) programming to either validate the candidates or recover valid barrier certificates. Our framework achieves great efficiency via supervised learning, and it ensures formal soundness using SOS-based verification. We implement the LR4BC tool, and we perform a comprehensive experimental evaluation using several benchmarks. The results demonstrate that our tool not only successfully synthesizes polynomial barrier certificates undetected via the SOS-based tool PRoTECT but also achieves a significant speedup in efficiency compared to the neural network-based tool FOSSIL 2.0.

Keywords: formal verification; barrier certificate; neural network learning; sums of squares programming

MSC: 68N30

check for updates

Academic Editor: Antanas Cenys

Received: 19 January 2025
Revised: 22 February 2025
Accepted: 24 February 2025
Published: 4 March 2025

Citation: Chen, D.; Lin, W.; Ding, Z. A Learner-Refiner Framework for Barrier Certificate Generation. *Mathematics* **2025**, *13*, 848. https://doi.org/10.3390/math13050848

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Many research studies have been dedicated to safety verification, aiming to achieve the reliability of cyber–physical systems (CPSs). Safety verification aims to determine whether a system will evolve into states that violate the safety property. However, the inherent complexity of CPSs remains a persistent challenge.

Recently, barrier certificate-based techniques have been extensively researched for the safety verification of dynamical systems, which are indispensable components of hybrid systems and are important in formal modeling for CPSs. Since the zero-level set of a barrier certificate can separate the reachable states of the dynamical system from the unsafe region, its existence ensures the safety of the system.

Generating barrier certificates entails significant computational requirements. Several symbolic computation methods, such as quantifier elimination [1] and Grönber basis computation [2], have been employed to address the challenge of quantification, facilitating barrier certificate verification. Conversely, symbolic approaches struggle with the difficulty of scalability.

Convex relaxation methods can achieve superior computational efficiency, but they yield more conservative results. Rather than directly handling quantified constraints, techniques like linear relaxation and sums of squares (SOS) relaxation are adopted to transform them into conservative ones, formulated as linear (or bilinear) matrix inequalities. These

transformed constraints can then be efficiently tackled using numerical approaches, including interval computation, semidefinite programming, and linear programming [3–6]. Nonetheless, maintaining a balance between computational efficiency and the expressiveness of barrier certificates is crucial.

Generally, methods for synthesizing neural barrier candidates utilize a counterexample-guided inductive synthesis framework that constitutes a learner and a verifier to generate neural barrier certificates [7–9]. In the proposed framework, the learner trains a neural candidate that satisfies the barrier conditions over a training dataset, while the verifier validates the candidate's correctness using such SMT solvers [7,10], MILP solvers [11–13], or CROWN [14]. For the CEGIS approach, its effectiveness and practicality are largely contingent on the scalability of satisfiability checking, and the complexity of the neural network architecture poses a challenge in verifying the accuracy of neural barrier candidates.

In this paper, we present a novel approach to generating polynomial barrier certificates that can ensure the safety of dynamical systems. The proposed framework operates as an iterative process with a *learner* and a *refiner* working together to generate polynomial barrier certificates. Concretely, the *learner* trains a barrier certificate candidate expressed by feedforward neural networks with polynomial activations, while the *refiner* utilizes sums of squares programming to either validate the candidates or recover real barrier certificates, thereby circumventing the time-consuming neural network verification. Our approach achieves great efficiency via supervised learning, and it ensures formal soundness using SOS-based verification. We implement the tool *LR4BC*, and we perform a comprehensive experimental evaluation using several benchmarks. The experimental results indicate the efficiency and effectiveness of our tool.

The structure of the remainder of this paper is as follows. In Section 2, we simply offer basic preliminaries. Section 3 discusses the training and refinement processes for polynomial barrier certificates, as aligned with Algorithm 1. In Section 4, we provide an experimental evaluation of our method. Section 5 presents related work, and Section 6 presents conclusions.

2. Related Work

2.1. Polynomial Barrier Certificate Generation

In pioneering work, Prajna et al. introduced safety verification grounded in barrier certificates, as detailed in [15,16]. They leveraged Putinar's Positivstellensatz theory to design an SOS framework for generating a barrier certificate. A barrier certificate based on the Lyapunov function was introduced by Kapinski et al. [17]. In their work, Kong et al. [18,19] introduced an exponential criterion specifically designed for generating barrier certificates suited to semi-algebraic hybrid systems. Sloth et al. [20] introduced a unique barrier certificate that is tailored to a specific type of hybrid system consisting of several interlinked parts. Dai et al. [21] investigated ways to generally loosen barrier certificate requirements without sacrificing convexity. Tuncali et al. [22] used a validation procedure in conjunction with a simulation-guided approach to ascertain barrier certificates. Zeng et al. [23] introduced Darboux-type barrier certificates that delineate an algebraic curve, ensuring that system trajectories do not escape once they cross it.

2.2. Neural Barrier Certificate Synthesis

The advent of neural networks has invigorated the synthesis of barrier certificates. Neural network verification is included in decision-making procedures, and barrier certifications may be provided in more flexible ways than just set shapes. As we know, Zhao and colleagues [24] presented a pioneering contribution that introduced neural networks as barrier certifications. To validate dynamical systems, neural networks with Bent-ReLU

activation are used. After identifying candidate networks, they approximate the Bent-ReLU function with piece-wise lines and then verify barrier certificate conditions using the non-linear SMT isat3 solver. In the CounterExample-Guided Inductive Synthesis (CEGIS) methodology [9,25], a learner teaches neural networks to fulfill the conditions of barrier certificates across a sample dataset, with these conditions articulated through a loss function. Leveraging the expressive prowess of neural networks, these certificates can capture more complex, non-linear safety constraints that are difficult to represent using traditional methodologies.

In order to convert counterexample generation into convex linear matrix inequality (LMI) feasibility problems that can be solved using SOS programming, Zhao et al. [26] created neural barrier certificates with a unique structure and used a particular form. The MIP-based approach converts the neural barrier certificate synthesis problem into a series of MIP problems, which are subsequently validated using numerical optimization solvers [12,13]. In order to handle high-dimensional continuous systems and shorten the time required to synthesize barrier certificates, Xia et al. [27] employed parallel learning approaches. Additionally, in complex tasks, reinforcement learning algorithms are often employed to learn neural barrier certificates in order to achieve near-zero constraint violations within accurate environment models [28].

3. Preliminaries

Notations. Let $\mathbb{R}$ be real number fields. $\mathbb{R}[\mathbf{x}]$ denotes the ring of polynomials with coefficients in $\mathbb{R}$ over variables $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, and $\mathbb{R}[\mathbf{x}]^n$ represents the n-dimensional polynomial ring vector. Moreover, $\Sigma[\mathbf{x}]$ denotes the space of SOS polynomials.

3.1. Safety Verification and Barrier Certificates

Consider the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in D \subseteq \mathbb{R}^n$ denotes the system states, and $\mathbf{f} : \Psi \times U \to \mathbb{R}^n$ represents a locally continuous vector field. Let $\mathbf{f}$ be a Lipschitz function; then, for each starting state $\mathbf{x}_0 \in \Psi$ at time $t = 0$, the system has a unique solution, $\mathbf{x}(t, \mathbf{x}_0)$, in Ψ.

Given an initial region, $\Theta \subset D$, and an unsafe region, $X_u \subset D$, the system (1) is *safe* if there exists no trajectory starting from Θ that may progress into any state indicated by X_u. The concept of *barrier certificates* is crucial for safety verification.

Theorem 1 ([16]). *Consider system (1); if there exists a real-valued function, $B : D \to \mathbb{R}$, such that*

(i) $B(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in \Theta$,

(ii) $B(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in X_u$,

(iii) $B(\mathbf{x}) = 0 \Rightarrow \mathcal{L}_f B(\mathbf{x}) > 0 \quad \forall \mathbf{x} \in D$,

where $\mathcal{L}_f B(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial B}{\partial x_i} \cdot f_i(\mathbf{x})$, then $B(\mathbf{x})$ is called a barrier certificate that ensures the safety property of the system.

In this paper, the domain D, the initial region Θ, and the unsafe region X_u are all assumed to be the basic semi-algebraic sets:

$$\begin{cases} D: & = \{\mathbf{x} \in \mathbb{R}^n \,|\, \psi(\mathbf{x}) \geq 0\}, \\ \Theta: & = \{\mathbf{x} \in \mathbb{R}^n \,|\, \theta(\mathbf{x}) \geq 0\}, \\ X_u: & = \{\mathbf{x} \in \mathbb{R}^n \,|\, \zeta(\mathbf{x}) \geq 0\}, \end{cases}$$

with $\psi(\mathbf{x}), \theta(\mathbf{x}), \zeta(\mathbf{x})$ being polynomial vectors, and the inequalities are satisfied in an entry-wise manner.

3.2. Neural Networks

Typically, a feed-forward neural network, characterized by its layered architecture, comprises multiple interconnected neurons arranged into one input layer, one output layer, and several intermediary hidden layers. Every neuron within a layer accepts inputs from neurons in the preceding layer and transmits processed signals to neurons in the subsequent layer. Subscript 0 and L respectively represent the input and output layers, and l serves as an index for variables specific to each layer, l. Furthermore, W_l denotes the weight matrix that connects layer $l-1$ to layer l, while $\mathbf{b}_l$ signifies the bias vector for layer l, and σ_l serves as the activation for layer l.

Let $\mathbf{x}_0$ be the values of neurons in the input layer. For hidden layers l with $1 \leq l \leq L$, $\mathbf{z}_l$ and $\mathbf{x}_l$ respectively denote the pre-activation and post-activation values of the neurons in layer l. The output layer's neuron values are denoted as $\mathbf{x}_L$. The equations below delineate the forward propagation of signals from the input layer to the layer L:

$$\begin{cases} \mathbf{z}_l = W_l \mathbf{x}_{l-1} + \mathbf{b}_l, & l = 1, 2, \ldots, L-1 \\ \mathbf{x}_l = \sigma_l(\mathbf{z}_l), & l = 1, 2, \ldots, L-1 \\ \mathbf{x}_L = W_L \mathbf{x}_{L-1} + \mathbf{b}_L. \end{cases} \tag{2}$$

3.3. Sums of Squares Programming

Assume that $\mathcal{T} = \{\mathbf{x} \in \mathbb{R}^n \mid p_1(\mathbf{x}) \geq 0, \ldots, p_m(\mathbf{x}) \geq 0\}$, with $p_i \in \mathbb{R}[\mathbf{x}], 1 \leq i \leq m$, is a basic semi-algebraic set; the polynomial set

$$M(p_1, \ldots, p_m) := \{\vartheta_0 + \sum_{i=1}^{m} \vartheta_i p_i \mid \vartheta_i \in \Sigma[\mathbf{x}],\ 0 \leq i \leq m\}$$

is called the quadratic module, and it is yielded via $\{p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})\}$.

Theorem 2 ([29]). *Let $M(p_1, \ldots, p_m)$ be Archimedean. If a polynomial, $f(\mathbf{x})$, is strictly positive over $\mathcal{T}$, then $f(\mathbf{x})$ can be rewritten as*

$$f(\mathbf{x}) = \vartheta_0(\mathbf{x}) + \sum_{i=1}^{m} \vartheta_i(\mathbf{x}) p_i(\mathbf{x}), \tag{3}$$

with $\vartheta_i \in \Sigma[\mathbf{x}]$ for each $i = 0, 1, \cdots, m$.

According to Theorem 2, the expression (3) establishes a sufficient and necessary criterion for polynomial positivity over a tight semi-algebraic set.

4. Learning and Refinement of Barrier Certificates

In the following, we suggest a novel framework, which operates through an interactive process involving a *learner* and a *refiner*, for the generation of polynomial barrier certificates. The *learner* trains a neural candidate for barrier certificates by integrating the barrier conditions into the neural network's loss function. When the loss converges to zero, the returned candidate certificate satisfies the barrier conditions on the training data. The *refiner* utilizes an SOS programming approach to expedite the verification process for the trained neural candidate, thereby circumventing the time-consuming neural network verification.

4.1. Training of Polynomial Barrier Candidates

The subsequent section devotes its attention primarily to the *learner* component, demonstrating the application of supervised learning techniques in training polynomial barrier certificates. These certificates are crafted using templates consisting of feedforward

neural networks of a specific configuration, comprising an initial layer for inputs and a final layer for outputs, along with $L-1$ hidden layers.

Specifically, system (1) features a state vector, $\mathbf{x}$; the dimensionality of this vector aligns precisely with the count of input neurons. Additionally, the neural network features a single output neuron, indicating that its output yields a numerical outcome without directionality.

Activation function. We adopt polynomials as legal activation functions. As introduced in [9], a γ-order polynomial activation function is defined as $\sigma(x) = x^{\gamma}$. This choice is necessitated by the requirement for the neural network to generate outputs that are interpretable and whose analytical forms can be easily manipulated by the verifier. It is evident that neural networks with a feed-forward architecture utilizing polynomial activations are fundamentally polynomials and are thus continuously differentiable.

Training data generation. The mesh construction method is used for sampling. Consider the initial set Θ. The bounds of Θ are defined by the vectors $\mathbf{l}, \mathbf{u} \in \mathbb{R}^n$, such that Θ is contained within the interval $[\mathbf{l}, \mathbf{u}]$. To begin with, we sample uniformly within $[\mathbf{l}, \mathbf{u}]$ using a uniform mesh size, δ. We then apply a filter to exclude states that do not meet the original barrier condition. Therefore, the training dataset D_I from Θ is generated. And the datasets $\mathcal{D}_U$ and $\mathcal{D}_X$ from the unsafe set X_U and the domain D can be constructed following the same procedure.

It should be noted that the datasets' cardinalities increase exponentially with the state space's dimension, which makes neural barrier certificate training extremely time-consuming. Using heterogeneous granularity, rather than uniform granularity, to discretize various state sets is one possible solution to the complexity bottleneck.

Loss function. To enhance the likelihood of satisfaction with barrier conditions, we introduce the following barrier loss:

$$\begin{aligned}\mathcal{L}_{total} = &\sum_{\mathbf{x}\in\mathcal{D}_I} \max(-B(\mathbf{x}) + \epsilon, 0) + \sum_{\mathbf{x}\in\mathcal{D}_U} \max(B(\mathbf{x}) - \epsilon, 0) \\ &+ \sum_{\mathbf{x}\in\mathcal{D}_X} \max(-\mathcal{L}_{\mathbf{f}} B(\mathbf{x}) + \epsilon, 0)\end{aligned}$$

where ϵ is a small non-negative constant. In the loss, the terms are directly linked to barrier conditions in order to measure the extent of violation for the barrier conditions. To train a candidate barrier certificate, we use backward propagation to update neural parameters by minimizing $\mathcal{L}_{total}$.

4.2. Refining of Polynomial Candidates

The candidate barrier certificates cannot be formally guaranteed in the learning stage; thus, it is important to confirm the correctness of the candidates. In [7,9], the dReal SMT solver is utilized to validate the candidate or generate counterexamples. Concretely, verifying a candidate barrier certificate involves determining the satisfiability of the following formulas:

$$\begin{cases} \forall \mathbf{x} \in \Theta. & B(\mathbf{x}) \geq 0 \\ \forall \mathbf{x} \in \Psi. & B(\mathbf{x}) = 0 \Rightarrow \mathcal{L}_f B(\mathbf{x}) > 0 \\ \forall \mathbf{x} \in X_u. & B(\mathbf{x}) < 0. \end{cases} \tag{4}$$

Since a feed-forward neural network with polynomial activations can be rewritten as a polynomial, problem (4) belongs to the decidable theory of real closed fields. However, the high computational complexity caused by interval computation in SMT solving makes the verification of complex barrier candidates remain a challenge.

In fact, the barrier conditions for polynomial dynamical systems can be transformed into a unified form, that is, checking the positivity of a polynomial over a semi-algebraic set. Below, we propose a sufficient but tractable approach based on SOS relaxation to recover true barrier certificates for polynomial dynamical systems. Let us consider the

following initial condition: $B(\mathbf{x}) \geq 0,\ \forall \mathbf{x} \in \Theta$, where $\Theta := \{\mathbf{x} \in \mathbb{R}^n | \theta(\mathbf{x}) \geq 0\}$. With the trained barrier candidate $B(\mathbf{x})$, the initial condition is met if there exists a vector of the SOS polynomial $\sigma(\mathbf{x})$ that makes

$$B(\mathbf{x}) - \sigma(\mathbf{x})\theta(\mathbf{x})$$

SOS. Consequently, Putinar's Positivstellensatz can assist in converting the candidate barrier certificate verification into the SOS problem:

$$\left.\begin{array}{l} \text{find } \sigma(\mathbf{x}), \phi(\mathbf{x}), \nu(\mathbf{x}), \mu(\mathbf{x}) \\ \text{s.t. } B(\mathbf{x}) - \sigma(\mathbf{x})\theta(\mathbf{x}) \text{ is SOS} \\ \quad \left\langle \frac{\partial B}{\partial \mathbf{x}}(\mathbf{x}), \mathbf{f}(\mathbf{x}) \right\rangle - \phi(\mathbf{x})\psi(\mathbf{x}) - \nu(\mathbf{x})B(\mathbf{x}) - \epsilon \text{ is SOS} \\ \quad -B(\mathbf{x}) - \mu(\mathbf{x})\zeta(\mathbf{x}) - \epsilon \text{ is SOS}, \end{array}\right\} \tag{5}$$

which can be solved using SDP efficiently. If the feasibility of the problem (5) is established, the yielded $B(\mathbf{x})$ can be served as an appropriate barrier certificate.

Indeed, the constraints in (5) provide a sufficient condition for barrier conditions. When (5) is rendered infeasible, an iterative approach is introduced to recover the real barrier certificate from the candidate $B(\mathbf{x})$.

Step 1: Calculate the multiplies $\sigma(\mathbf{x}), \phi(\mathbf{x}), \nu(\mathbf{x})$, and $\mu(\mathbf{x})$. Fix the computed $B(\mathbf{x})$, and solve the following SOS problem to obtain $\sigma(\mathbf{x}), \phi(\mathbf{x}), \nu(\mathbf{x}), \mu(\mathbf{x})$ by maximizing ε:

$$\left.\begin{array}{l} \max\ \varepsilon \\ \text{s.t. } \overline{B}(\mathbf{x}) - \sigma(\mathbf{x})\theta(\mathbf{x}) \text{ is SOS} \\ \quad \left\langle \frac{\partial \overline{B}}{\partial \mathbf{x}}(\mathbf{x}), \mathbf{f}(\mathbf{x}) \right\rangle - \phi(\mathbf{x})\psi(\mathbf{x}) - \nu(\mathbf{x})\overline{B}(\mathbf{x}) - \varepsilon \text{ is SOS} \\ \quad -\overline{B}(\mathbf{x}) - \mu(\mathbf{x})\zeta(\mathbf{x}) - \varepsilon \text{ is SOS}, \end{array}\right\} \tag{6}$$

Step 2: Calculate the barrier candidate $B(\mathbf{x})$. Specify $\nu(\mathbf{x})$, and solve the following SOS problem to obtain $B(\mathbf{x})$:

$$\left.\begin{array}{l} \max\ \varepsilon \\ \text{s.t. } B(\mathbf{x}) - \sigma(\mathbf{x})\theta(\mathbf{x}) \text{ is SOS} \\ \quad \left\langle \frac{\partial B}{\partial \mathbf{x}}(\mathbf{x}), \mathbf{f}(\mathbf{x}) \right\rangle - \phi(\mathbf{x})\psi(\mathbf{x}) - \overline{\nu}(\mathbf{x})B(\mathbf{x}) - \varepsilon \text{ is SOS} \\ \quad -B(\mathbf{x}) - \mu(\mathbf{x})\zeta(\mathbf{x}) - \varepsilon \text{ is SOS}. \end{array}\right\} \tag{7}$$

The symbol *bar* signifies that the polynomials with the value were obtained from the preceding iteration; alternatively, for the first iteration, it represents the initial value. The final optimized value for $B(\mathbf{x})$ is used as the output, and the iterative process ends when $\varepsilon > 0$. Consequently, the system's safety property can be ensured by the presence of a solution to (7) with $\varepsilon > 0$. When (7) is infeasible, increasing the degrees for the polynomial activations can enhance the likelihood of computing $B(\mathbf{x})$.

4.3. The Learning Algorithm

Algorithm 1 outlines the iterative structure for synthesizing barrier certificates that ensure the safety of polynomial dynamical systems. The procedure accepts as input a dynamical system, $\mathbf{f}$, and the domain D, the unsafe region X_u, the initial region Θ, the maximum number of iteration *MaxIter*, the learning rate α, and the loss function parameters ϵ. It yields the real barrier certificates of a specified architecture or reports failure if it is unable to do so. The implementation process consists of the following two steps in a pass of the iteration.

(i) Generate training datasets $(\mathcal{D}_i, \mathcal{D}_U, \mathcal{D}_X)$ from (Θ, X_u, D). The learner continues to train neural candidates and adjust the parameters W and b until the barrier loss reaches zero in Lines (4–9).

(ii) From step (i), the refiner obtains a trained candidate and determines its validation. $B(\mathbf{x})$ qualifies as a valid barrier certificate, provided that problem (6) is possible; if not, (7)–(8) are solved iteratively until $\varepsilon > 0$, and the final optimized value for $B(\mathbf{x})$ is used as the output in Lines (11,15).

The interaction between the learner and the refiner proceeds iteratively until the iteration number *iter* reaches the threshold of maximum iterations, *MaxIter*.

Algorithm 1 Learner–Refiner Framework

Input: $\mathbf{f}$, Θ, D, X_u, $MaxIter$, ε, and α
Output: $B(\mathbf{x})$ or Failure
1: $iter \leftarrow 0$;
2: $(\mathcal{D}_I, \mathcal{D}_U, \mathcal{D}_X) \leftarrow$Sampling$(\Theta, D, X_u)$
3: Initialize NN_θ
4: **repeat**
5: $\quad B(\mathbf{x}) \leftarrow NN_\theta(\mathbf{x})$
6: $\quad$ Compute $\mathcal{L}_\theta$
7: $\quad W \leftarrow W + \alpha \nabla_W \mathcal{L}$
8: $\quad b \leftarrow b + \alpha \nabla_b \mathcal{L}$
9: **until** convergence
10: **if** (6) is feasible **then**
11: $\quad$ Return $B(\mathbf{x})$
12: **end if**
13: **while** $iter < maxIter$ **do**
14: $\quad$ **if** (8) is feasible and $\varepsilon \geq 0$ **then**
15: $\quad\quad$ Return $B(\mathbf{x})$
16: $\quad$ **end if**
17: **end while**

Note that, if $B(\mathbf{x})$ is the function produced via Algorithm 1, then $B(\mathbf{x})$ satisfies all the barrier certificate conditions in Theorem 1, and the safety of system (1) is guaranteed. More specifically, in Line 3, we represent $B(\mathbf{x})$ as a polynomial by neural network and train the candidate through gradient descent. In Line 10, we solve the feasibility of (6) to check the correctness of $B(\mathbf{x})$, and the algorithm returns a real barrier certificate if it is feasible. Otherwise, Lines 13–17 will repeat the refinement until Line 13 is satisfied, and the algorithm returns $B(\mathbf{x})$ as a real barrier certificate. Therefore, it is guaranteed that the algorithm is sound.

5. Experiments

Based on Algorithm 1, we implemented the *LR4BC* tool for barrier certificate generation using the PyTorch platform and SOSTOOLS [30]. Then, we compared it with the state-of-the-art neural BC synthesis tool FOSSIL [25] and the SOS-based BC generation tool PRoTECT [31] against a number of benchmarks from the literature In experiments, we started from a simpler network structure and increased the layer number until the desired results were obtained. We tried using polynomial activations of different degrees, and we incremented the degree if we failed to yield the desired results. All experiments were conducted on a machine with 64 GB of RAM, an Intel i9-10900K CPU, and an NVIDIA GeForce RTX 3080Ti GPU running Ubuntu. Additionally, we used AdamW as the optimization algorithm; its learning rate is 0.01, and its batch size is 500.

Example 1. *Consider the Van der Pol system [16]*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 - x_2 + \frac{1}{3}x_1^3 \end{bmatrix}$$

with domain $D = \{\mathbf{x} \in \mathbb{R}^2 | -2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2\}$. *We aim to ensure that all trajectories starting from* $\Theta = \{\mathbf{x} \in \mathbb{R}^2 | (x_1 - 1.5)^2 + x_2^2 \leq 0.25\}$ *cannot reach*

$$\mathcal{X}_u = \{\mathbf{x} \in \mathbb{R}^2 | (x_1 + 1)^2 + (x_2 + 1)^2 \leq 0.16\}.$$

We obtain a polynomial barrier certificate:

$$B(\mathbf{x}) = -0.4684x_1^2 + 0.1368x_1x_2 + 1.878x_1 - 1.504x_2^2 + 0.8993x_2 + 1.866.$$

From Figure 1, the zero level set of $B(\mathbf{x})$ *(the blue line) divides* $\mathcal{X}_u$ *(the red circle) from all trajectories from* Θ *(the green circle).*

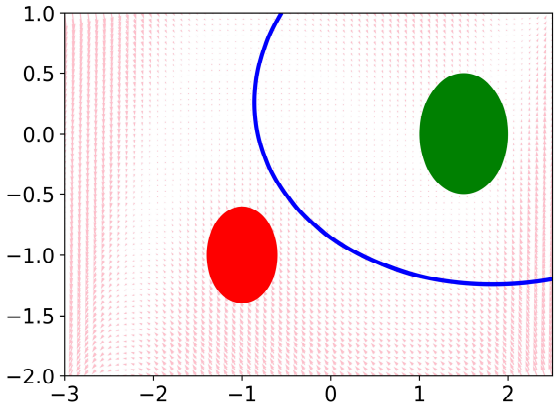


Figure 1. Phase portrait of the system in Example 1.

Example 2. *Consider the system [32]*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{1}{3}x_1^3 + x_1 - x_2 + \frac{7}{8} \\ \frac{2}{25}(x_1 - \frac{4}{5}x_2 + \frac{7}{10}) \end{bmatrix}$$

with the domain $D = \{\mathbf{x} \in \mathbb{R}^2 | -4 \leq x_1, x_2 \leq 4\}$. *We aim to ensure that all trajectories from* $\Theta = \{\mathbf{x} \in \mathbb{R}^2 | -1 \leq x_1 \leq 1, 1 \leq x_2 \leq 3\}$ *cannot enter the unsafe region*

$$\mathcal{X}_u = \{\mathbf{x} \in \mathbb{R}^2 | (x_1 + 2.25)^2 + (x_2 + 1.75)^2 \leq 1\}.$$

We obtain a polynomial barrier certificate:

$$B(\mathbf{x}) = -0.2087x_1^2 - 0.02166x_1x_2 + 0.8298x_1 - 0.08925x_2^2 + 3.833x_2 + 2.016.$$

From Figure 2, the zero level set of $B(\mathbf{x})$ *(the blue line) separates all trajectories from* Θ *(the green circle) with* $\mathcal{X}_u$ *(the red circle).*

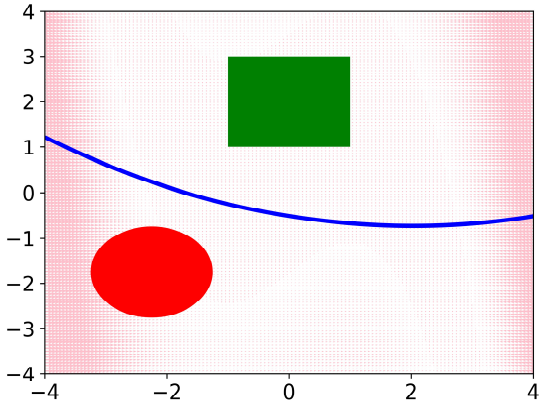


Figure 2. Phase portrait of the system in Example 2.

Example 3. *Consider the system [33]*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -x_1 + x_2 - x_3 \\ -x_1(x_3 + 1) - x_2 \\ -x_1 + 1.8x_1 - 4.8x_3 \end{bmatrix}$$

with domain $D = \{\mathbf{x} \in \mathbb{R}^3 | -3 \leq x_1, x_2, x_3 \leq 3\}$. *We aim to check that all trajectories from*

$$\Theta = \{\mathbf{x} \in \mathbb{R}^3 | (x_1 - 1.5)^2 + x_2^2 + x_3^2 \leq 0.0625\}$$

cannot enter the unsafe set

$$\mathcal{X}_u = \{\mathbf{x} \in \mathbb{R}^3 | (x_1 + 1)^2 + (x_2 + 1)^2 + (x_3 - 1)^2 \leq 0.64\}.$$

We finally obtain a barrier certificate:

$$\begin{aligned} \tilde{B}(\mathbf{x}) = &-0.7076x_1^2 + 0.09938x_1x_2 + 0.2374x_1x_3 + 0.913x_1 - 0.9915x_2^2 \\ &- 0.06812x_2x_3 - 0.2453x_2 - 1.148x_3^2 - 0.3766x_3 + 1.01. \end{aligned}$$

From Figure 3, the zero level set of $B(\mathbf{x})$ *(the yellow ball) divides* $\mathcal{X}_u$ *(the red ball) of all trajectories from* Θ *(the green ball).*

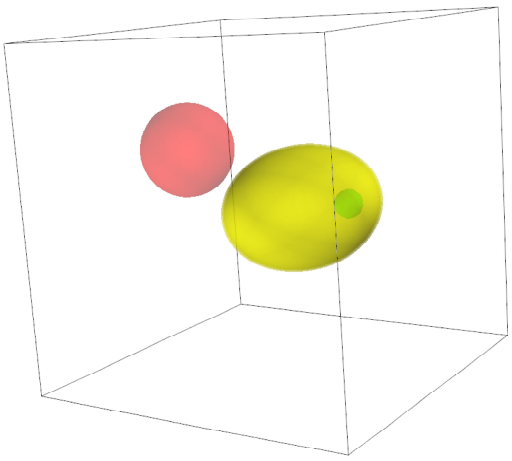


Figure 3. Phase portrait of the system in Example 3.

Example 4. *Consider the system [33]*

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} -x_1^3 + 4x_2^3 - 6x_3x_4 \\ -x_1 - x_2 + x_5^3 \\ x_1x_4 - x_3 + x_4x_6 \\ x_1x_3 + x_3x_6 - x_4^3 \\ -2x_2^3 - x_5 + x_6 \\ -3x_3x_4 - x_5^3 - x_6 \end{bmatrix}$$

with domain

$$D = \Big\{\mathbf{x} \in \mathbb{R}^6 | 0 \leq x_1, x_2, x_4, x_5, x_6 \leq 10, 2 \leq x_3 \leq 10\Big\}.$$

We aim to check that all trajectories from $\Theta = \{\mathbf{x} \in \mathbb{R}^6 | \sum_{i=1}^{6}(x_i - 3.05)^2 \leq 0.05^2\}$ *cannot reach*

$$\begin{aligned} \mathcal{X}_u = &\{\mathbf{x} \in \mathbb{R}^6 | (x_1 - 4.05)^2 + (x_2 - 4.15)^2 + (x_3 - 4.25)^2 \\ &+ (x_4 - 4.35)^2 + (x_5 - 4.45)^2 + (x_6 - 4.55)^2 \leq 0.05^2\}. \end{aligned}$$

We can obtain a polynomial barrier certificate:

$$\begin{aligned} B(\mathbf{x}) = &-0.1251x_1^2 - 0.01243x_1x_2 - 0.1408x_1x_3 + 0.04704x_1x_4 \\ &- 0.003081x_1x_5 - 0.03951x_1x_6 + 0.2585x_1 - 0.09061x_2^2 \\ &- 0.03143x_2x_3 - 0.01989x_2x_4 + 0.02753x_2x_5 + 1.028x_2 \\ &+ 0.00532x_2x_6 - 0.2757x_3^2 + 0.1363x_3x_4 - 0.04116x_3x_5 \\ &+ 0.0221x_3x_6 + 2.213x_3 - 0.4358x_4^2 - 0.001608x_5x_6 \\ &+ 0.03342x_4x_5 + 0.1373x_4x_6 + 1.444x_4 - 0.01464x_5^2 \\ &- 1.026x_5 - 0.08375x_6^2 - 1.471x_6 + 4.466, \end{aligned}$$

which guarantee the safety of the system.

Table 1 lists the performance evaluation of *LR4BC*, *PRoTECT*, and *FOSSIL* over the above examples. Here, n represents the dimension of the system state, $d_{\mathbf{f}}$ represents the maximal degree of the polynomials in $\mathbf{f}$, and $\mathcal{N}$ means the network structure used in *LR4BC* and *FOSSIL*. d represents the degree of the valid polynomial barrier certificates, and t is the computation time (in seconds) in *PRoTECT*. $Iters_1$ and $Iters_2$ represent the iteration number of CEGIS in *FOSSIL* and refinement in *LR4BC*, respectively. t_l, t_v, and t_r are the time (in seconds) of learning, verification, and refinement, respectively. Furthermore, $-$ means that the approach cannot yield polynomial barrier certificates with a degree less than 6.

Table 1. Algorithm performance on benchmarks.

EX	n	$d_{\mathbf{f}}$	$\mathcal{N}$	PRoTECT		Fossil 2.0				LR4BC			
				d	t	d	Iter_1	t_l	t_v	d	t_l	Iter_2	t_r
1	2	3	[2, 10, 1]	4	1.336	2	1	0.886	0.003	2	0.886	1	0.056
2	2	3	[2, 10, 1]	6	6.134	2	2	0.385	0.011	2	0.193	1	0.061
3	3	2	[3, 10, 1]	6	436.5	2	5	6.464	0.342	2	1.293	1	0.067
4	6	3	[6, 10, 1]	-	-	2	14	0.741	589.998	2	0.053	1	0.766

From Table 1, our tool *LR4BC* can synthesize valid polynomial barrier certificates for all examples, while *PRoTECT* cannot deal with C4. Furthermore, *LR4BC* can yield much simpler polynomial barrier certificates with lower degrees. Our *LR4BC* seems to provide a better solution capability than *PRoTECT*.

Consider the computational efficiency of *LR4BC* and *FOSSIL*. On average, our *LR4BC* needs about 0.844 seconds to synthesize a valid barrier certificate, while the *FOSSIL* tool takes more than 149 seconds, which is 170 times as long as our tool. Concretely, for Ex1, *FOSSIL* yields a barrier certificate faster than our *LR4BC*, but for Ex2, Ex3, and Ex4, our *LR4BC* performs much better than *FOSSIL* in efficiency. Especially for Ex4, our *LR4BC* manages to handle it in about 0.819 seconds with 1 iteration, while the *FOSSIL* tool needs 590.739 seconds with 14 iterations. The reason is that SOS programming enhances the efficiency of barrier certificate verification in high-dimensional systems.

6. Conclusions

This work has presented a new approach to generating polynomial barrier certificates, ensuring the safety of dynamical systems. The synthesis process includes two phases: training the neural barrier candidate and conducting the formal verification of that candidate. The learner develops a prospective model embodied by a neural network. The verifier uses SOS programming to expedite the identification of barrier certificates from candidate ones. The tool was deployed, and its performance was assessed across several benchmark examples. The testing outcomes demonstrate that our tool exhibits superior effectiveness and efficiency.

Author Contributions: Methodology, W.L.; Resources, Z.D.; Writing—original draft, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62272416, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020006, and in part by the Fundamental Research Funds of Zhejiang Sci-Tech University under Grant 24232090-Y.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gulwani, S.; Tiwari, A. Constraint-Based Approach for Analysis of Hybrid Systems. In Proceedings of the Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, NJ, USA, 7–14 July 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 190–203.
2. Sturm, T.; Tiwari, A. Verification and synthesis using real quantifier elimination. In Proceedings of the Symbolic and Algebraic Computation, International Symposium, ISSAC 2011, San Jose, CA, USA, 8–11 June 2011; pp. 329–336.
3. Djaballah, A.; Chapoutot, A.; Kieffer, M.; Bouissou, O. Construction of parametric barrier functions for dynamical systems using interval analysis. *Automatica* **2017**, *78*, 287–296. [CrossRef]
4. Yang, Z.; Wu, M.; Lin, W. An efficient framework for barrier certificate generation of uncertain nonlinear hybrid systems. *Nonlinear Anal. Hybrid Syst.* **2020**, *36*, 100837. [CrossRef]
5. Chen, X.; Peng, C.; Lin, W.; Yang, Z.; Zhang, Y.; Li, X. A Novel Approach for Solving the BMI Problem in Barrier Certificates Generation. In Proceedings of the Computer Aided Verification, Los Angeles, CA, USA, 19–24 July 2020; pp. 582–603.
6. Wang, Q.; Chen, M.; Xue, B.; Zhan, N.; Katoen, J.P. Synthesizing Invariant Barrier Certificates via Difference-of-Convex Programming. In Proceedings of the 33rd International Conference on Computer Aided Verification, CAV 2021, Virtual Event, 18–24 July 2021; pp. 443–466.
7. 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 Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control, Nashville, TN, USA, 19–21 May 2021; pp. 1–11.
8. Edwards, A.; Peruffo, A.; Abate, A. A General Verification Framework for Dynamical and Control Models via Certificate Synthesis. *arXiv* **2023**, arXiv:2309.06090.

9. Peruffo, A.; Ahmed, D.; Abate, A. Automated and formal synthesis of neural barrier certificates for dynamical models. In Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Luxembourg, 27 March–1 April 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 370–388.
10. Ahmed, D.; Peruffo, A.; Abate, A. Automated and sound synthesis of Lyapunov functions with SMT solvers. In Proceedings of the Tools and Algorithms for the Construction and Analysis of Systems: 26th International Conference, TACAS 2020, Dublin, Ireland, 25 April–1 May 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 97–114.
11. 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.* **2023**, *39*, 1749–1767. [CrossRef]
12. Zhao, Q.; Chen, X.; Zhang, Y.; Sha, M.; Yang, Z.; Lin, W.; Tang, E.; Chen, Q.; Li, X. Synthesizing ReLU neural networks with two hidden layers as barrier certificates for hybrid systems. In Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control, Nashville, TN, USA, 19–21 May 2021; pp. 1–11.
13. Zhao, Q.; Chen, X.; Zhao, Z.; Zhang, Y.; Tang, E.; Li, X. Verifying neural network controlled systems using neural networks. In Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control, Milan, Italy, 4–6 May 2022; pp. 1–11.
14. Xu, K.; Shi, Z.; Zhang, H.; Wang, Y.; Chang, K.W.; Huang, M.; Kailkhura, B.; Lin, X.; Hsieh, C.J. Automatic perturbation analysis for scalable certified robustness and beyond. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1129–1141.
15. Prajna, S.; Jadbabaie, A. Safety Verification of Hybrid Systems Using Barrier Certificates. In Proceedings of the Hybrid Systems: Computation and Control, Philadelphia, PA, USA, 25–27 March 2004; pp. 477–492.
16. Prajna, S.; Jadbabaie, A.; Pappas, G.J. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Trans. Autom. Control* **2007**, *52*, 1415–1429. [CrossRef]
17. Kapinski, J.; Deshmukh, J.V.; Sankaranarayanan, S.; Arechiga, N. Simulation-Guided Lyapunov Analysis for Hybrid Dynamical Systems. In Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control, Berlin, Germany, 15–17 April 2014; pp. 133–142.
18. Kong, H.; He, F.; Song, X.; Hung, W.; Gu, M. Exponential-Condition-Based Barrier Certificate Generation for Safety Verification of Hybrid Systems. In Proceedings of the 25th International Conference on Computer Aided Verification, CAV2013, Saint Petersburg, Russia, 13–19 July 2013; pp. 242–257.
19. Kong, H.; Song, X.; Han, D.; Gu, M.; Sun, J. A New Barrier Certificate for Safety Verification of Hybrid Systems. *Comput. J.* **2014**, *57*, 1033–1045. [CrossRef]
20. Sloth, C.; Pappas, G.; Wisniewski, R. Compositional safety analysis using barrier certificates. In Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control, HSCC'12, Beijing, China, 17–19 April 2012; pp. 15–24.
21. Dai, L.; Gan, T.; Xia, B.; Zhan, N. Barrier certificates revisited. *J. Symb. Comput.* **2017**, *80*, 62–86. [CrossRef]
22. Tuncali, C.E.; Kapinski, J.; Ito, H.; Deshmukh, J.V. Reasoning about Safety of Learning-Enabled Components in Autonomous Cyber-Physical Systems. In Proceedings of the 55th Annual Design Automation Conference, DAC 2018, San Francisco, CA, USA, 24–29 June 2018; pp. 30:1–30:6.
23. Zeng, X.; Lin, W.; Yang, Z.; Chen, X.; Wang, L. Darboux-Type Barrier Certificates for Safety Verification of Nonlinear Hybrid Systems. In Proceedings of the 13th International Conference on Embedded Software, Chengdu, China, 13–14 August 2016; pp. 11:1–11:10.
24. Zhao, H.; Zeng, X.; Chen, T.; Liu, Z. Synthesizing barrier certificates using neural networks. In Proceedings of the HSCC'20: 23rd ACM International Conference on Hybrid Systems: Computation and Control, Sydney, NSW, Australia, 21–24 April 2020; pp. 25:1–25:11.
25. Edwards, A.; Peruffo, A.; Abate, A. Fossil 2.0: Formal Certificate Synthesis for the Verification and Control of Dynamical Models. In Proceedings of the 27th ACM International Conference on Hybrid Systems: Computation and Control, Hong Kong, China, 14–16 May 2024; pp. 1–10.
26. Zhao, H.; Qi, N.; Ren, M.; Zeng, X.; Zeng, Z.; Yang, Z. Neural Barrier Certificates Synthesis of NN-Controlled Continuous Systems via Counterexample-Guided Learning. In Proceedings of the 61st ACM/IEEE Design Automation Conference, DAC 2024, San Francisco, CA, USA, 23–27 June 2024; pp. 70:1–70:6.
27. 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, San Francisco, CA, USA, 10–14 July 2022; pp. 1201–1206.
28. Yang, Y.; Jiang, Y.; Liu, Y.; Chen, J.; Li, S.E. Model-Free Safe Reinforcement Learning Through Neural Barrier Certificate. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1295–1302. [CrossRef]
29. Parrilo, P. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 2000.
30. Prajna, S.; Papachristodoulou, A.; Parrilo, P. SOSTOOLS: Sum of Squares Optimization Toolbox for MATLAB. 2002. Available online: http://www.cds.caltech.edu/sostools (accessed on 1 January 2025).

31. Wooding, B.; Horbanov, V.; Lavaei, A. PRoTECT: Parallelized Construction of Safety Barrier Certificates for Nonlinear Polynomial Systems. *arXiv* **2024**, arXiv:2404.14804.
32. Sogokon, A.; Ghorbal, K.; Johnson, T.T. Non-linear continuous systems for safety verification (benchmark proposal). In Proceedings of the Applied Verification for Continuous and Hybrid Systems Workshop (ARCH), Vienna, Austria, 12 April 2016.
33. Bouissou, O.; Chapoutot, A.; Djaballah, A.; Kieffer, M. Computation of parametric barrier functions for dynamical systems using interval analysis. In Proceedings of the IEEE 53rd Annual Conference on Decision and Control (CDC), Los Angeles, CA, USA, 15–17 December 2014; pp. 753–758.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.