Formal Synthesis of Safe Kolmogorov-Arnold Network Controllers with Barrier Certificates

Xiongqi Zhang , **Ning Lv** , **Wang Lin**[*] , **Zuohua Ding**
School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China
{xqiii_zhang, lvning_0826}@163.com, linwang@zstu.edu.cn, zouhuading@hotmail.com

Abstract

Control barrier certificate generation is an efficient and powerful technique for the safe control of cyber-physical systems. Feed-forward neural networks (FNNs) are commonly used to synthesize control barrier certificates and safe controllers, but they struggle to effectively address the challenges posed by high-dimensional complex systems. In this paper, we propose a novel method for generating control barrier certificates and controllers using Kolmogorov-Arnold Networks (KANs). Specifically, it utilizes KANs to replace FNNs as the template of control barrier certificates and contrllers. Since KAN has learnable activation functions, it can efficiently improve the representation power. Then, it leverages the pruning and symbolization properties of KANs, which significantly simplify the network structure, allowing for more efficient formal verification of the simplified candidate KAN control barrier certificates and controllers using Satisfiability Modulo Theories. We implement the tool KAN4CBC, and evaluate its performance over a set of benchmarks. The experimental results demonstrate that our method addresses the issues of system dimension expansion and improved solution efficiency.

1 Introduction

Controller design and synthesis is a fundamental challenge in control theory, particularly for complex and nonlinear systems. With the development of deep learning, neural networks have become widely used in controlling various cyber-physical systems, such as unmanned aerial vehicles, autonomous vehicles, and smart transportation [Andrychowicz *et al.*, 2020; Ding and Tomlin, 2010]. It is important to note that many of these systems are safety-critical, meaning the controller must ensure that the system never enter dangerous or undesirable states during operation. However, synthesizing safe controllers is inherently complex and presents a significant challenge.

[*]Corresponding authors.

Control barrier certificate (CBC) generation is an efficient and powerful technique often used for the formal synthesis of safe controllers. The conditions of control barrier certificates impose certain constraints on the controllers. When a control barrier certificate is identified, it divides the model's state space into two regions. This division ensures that any system trajectory starting from a given initial set lies on one side of the control barrier certificate, and cannot reach a given unsafe set on the other side. Therefore, we formally prove the safety of controlled dynamical systems by constructing a control barrier certificate, the existence of which is sufficient to guarantee the safety of the controller.

Sum-of-squares (SOS) programming is a traditional method for synthesizing polynomial barrier certificates [Prajna, 2006; Legat *et al.*, 2020]. But this method lacks scalability and expressiveness. Based on the universal approximation theorem [Leshno *et al.*, 1993], neural networks have the capability to approximate any arbitrary function, making them suitable for representing barrier certificates. Zhao et al. [2020] first synthesized barrier certificates via neural network training and verification. FOSSIL [Abate *et al.*, 2021] is a tool for the automated formal synthesis of barrier certificates, using neural networks as templates and Satisfiability Modulo Theories (SMT) solvers as verification tools. However, traditional feed-forward neural networks (FNNs) often use fixed activation functions, which can significantly impact the efficiency of synthesizing barrier certificates. Moreover, for controlled continuous dynamical systems, most methods tend to perform well on low-dimensional systems but struggle to effectively address the challenges posed by high-dimensional complex systems. This is mainly because high-dimensional systems require more complex neural network structures to design controllers and control barrier certificates, which causes general SMT verification methods to fail.

Liu et al. [2024] proposed Kolmogorov-Arnold Networks (KANs) as a promising alternative to multi-layer Perceptrons. In contrast to traditional models, KANs use activation functions on the connections between nodes, which can also learn and adapt during the training process. This architectural breakthrough enables KANs to better capture complex, nonlinear relationships by directly optimizing univariate functions. In addition, KANs have pruning properties that can remove unimportant edges and nodes from the network, significantly reducing its complexity. To address the issue of general SMT verification failure caused by neural controllers and neural control barrier certificates in high-dimensional systems, in this paper, we propose a new framework using KANs instead of FNNs to formally synthesize safe controllers and control barrier certificates. Specifically, our proposed framework consists of two parts. In the first part, it uses KANs to generate candidate control barrier certificates and controllers. The B-spline-based learnable activation function enhances the network's representation power and effectively addresses the issues of low success rates encountered when using fixed activation functions in FNNs for safe controller synthesis. In the second part, it prunes the candidate barrier certificate and controller networks, allowing the networks to more efficiently extract important features. Then, symbolization is applied to the networks, resulting in simpler and more human-readable expressions. Finally, it checks whether the candidates satisfy all the requirements of control barrier certificates through the SMT solver dReal. Overall, the main contributions of this work are described as follows:

- We introduce a new type of control barrier certificates and controllers represented as KANs, which uses learnable activation functions to enhance representation power.
- We leverage the pruning and network symbolization of KANs, improving both the efficiency and success rate of SMT verification in high-dimensional systems.
- We implement our tool KAN4CBC and conduct a detailed experimental evaluation on a set of benchmarks, demonstrating that our approach is more effective in generating safe controllers than the state-of-the-art tools.

2 Related Work

There are two main methods for analyzing and verifying continuous dynamical systems: reachability computation and barrier certificate synthesis. Reachability relies on over- and under-approximations, but due to the lack of closed-form solutions for ODEs and the computational cost of numerical methods, its scalability is limited. The concept of barrier certificates was first introduced by Prajna [2006], dividing the state space into safe and unsafe regions. This approach provides a more efficient and accurate proof of safety, particularly for nonlinear and uncertain systems, and is especially effective for verifying safety over an infinite time horizon.

In the field of control, researchers have explored various forms of barrier certificates to accommodate different types of dynamical systems. Many studies [Ahmadi and Majumdar, 2016; She *et al.*, 2013] have employed SOS relaxation and semidefinite programming methods to generate polynomial barrier certificates. These approaches transform the barrier certificate synthesis problem into constraints expressed as linear or bilinear matrix inequalities, which are typically solved through numerical optimization. Kong et al. [2014] introduced a novel exponential condition barrier certificate that maintains convexity while reducing conservativeness, providing a more precise and effective approach for safety verification of semialgebraic hybrid systems. Zeng et al. [2016] proposed a new verification condition and a novel computational method combining sampling-based relaxation with least-squares and quadratic programming (LS-QP) alternating projection to find Darboux-type barrier certificates, enhancing the verification of nonlinear hybrid systems. Sogokon et al. [2018] introduced a multi-dimensional framework using vector barrier certificates, thereby extending their applicability.

The emergence of neural networks has provided new possibilities for synthesizing barrier certificates. Researchers have explored learning neural network certificate functions and achieved encouraging results [Zhang *et al.*, 2023b; Lindemann *et al.*, 2021]. However, to ensure deterministic guarantees, these methods must include a verification procedure to confirm the validity of the certificate. Some contributions have employed counterexample-guided Inductive Synthesis (CEGIS) procedure [Liu *et al.*, 2023], where a learner trains a neural network to meet barrier certificate requirements over a finite set of samples, and a verifier either proves validity of the barrier certificate or provides counterexamples using an SMT solver [Kapinski *et al.*, 2014; Chang *et al.*, 2019]. However, the use of fixed activation functions can significantly impact the efficiency and success rate of generating barrier certificates. To address the non-differentiability issue caused by ReLU, Zhang et al. [2023a] proposed a piecewise linear optimization method combined with Interval Bound Propagation for verification. Hu et al. [2024] proposed using symbolic derivative boundary propagation to verify ReLU-based neural CBCs. However, these methods are not universally applicable and become inefficient for high-dimensional systems. Therefore, in this paper, we propose improvements to the network by using KAN, which not only addresses the issues caused by fixed activation functions but also ensures that SMT verification remains efficient in high-dimensional systems.

3 Preliminaries

This section introduces the basic concepts used in the paper.

3.1 Continuous Dynamical System

Consider a controlled continuous dynamical system S:

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

where $\mathbf{x} \subseteq \mathbb{R}^n$ represents the state of the system, $\dot{\mathbf{x}}$ denotes the derivative of $\mathbf{x}$ with respect to time t, $\mathbf{u} \subseteq \mathbb{R}^m$ means a control input, $\mathbf{f} = (f_1, f_2, \cdots, f_n)^T$ is the vector field on the state space $\Omega \subseteq \mathbb{R}^n$. We define a feedback controller $\mathbf{u} = \mathbf{h}(\mathbf{x})$, which leads to the autonomous closed-loop dynamics given by:

$$\dot{\mathbf{x}} = \mathbf{f}_{\mathbf{h}}(\mathbf{x}) := \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x})). \tag{2}$$

Let $X \subseteq \mathbb{R}^n$ be a compact set denoting the state space of the system, and $X_I, X_U \subseteq X$ represent the set of initial states and unsafe states, respectively.

We assume that $\mathbf{f}$ and $\mathbf{h}$ are Lipschitz continuous, ensuring the existence of a time trajectory $\xi_{\mathbf{h}}^{\mathbf{x}_0}(t)$, where $\xi_{\mathbf{h}}^{\mathbf{x}_0}(t)$ starting from $\mathbf{x_0}$ denotes the value of the state $\mathbf{x}$ at time $t > 0$.

In this paper, we focus on the problem of safe controller synthesis, specifically designing a safe controller $\mathbf{u} = \mathbf{h}(\mathbf{x})$ that guarantees the controlled system (2) remains safe.

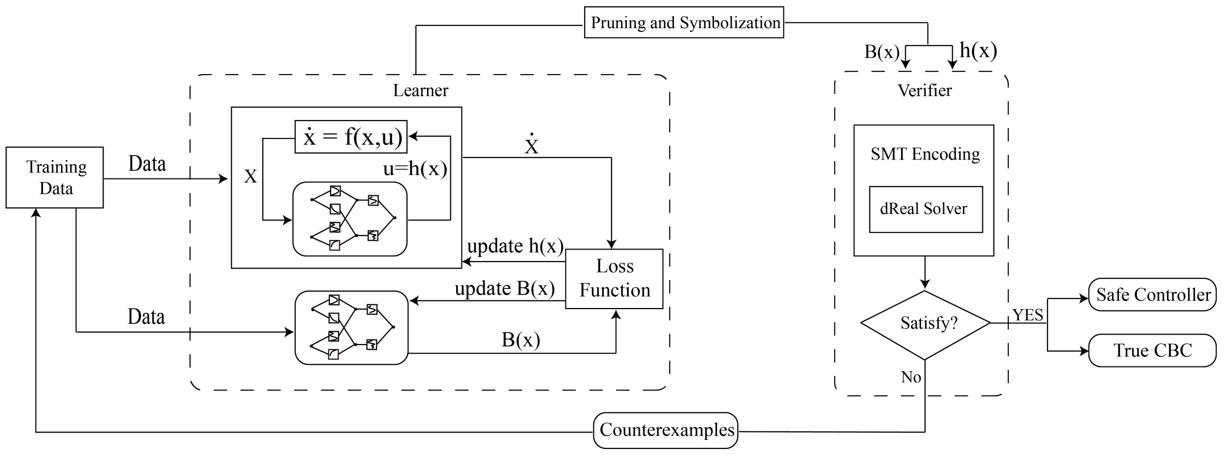

Figure 1: The framework of safe controller synthesis

Definition 1 (safety)**.** *For a controlled continuous dynamical system, a given initial region $X_I \subseteq X$ and a given unsafe region $X_U \subseteq X$, the controller $\mathbf{u} = \mathbf{h}(\mathbf{x})$ is a safe one if the following condition holds:*

$$\xi_{\mathbf{h}}^{\mathbf{x}_0}(t) \in X \Rightarrow \xi_{\mathbf{h}}^{\mathbf{x}_0}(t) \notin X_U, \forall \mathbf{x}_0 \in X_I, \forall t \geq 0. \quad (3)$$

3.2 Control Barrier Certificate

Verifying the safety of the controller requires analyzing the behavior of system (2) over an infinite time horizon. Following [28], we aim to find a control barrier certificate as proof of safety, with the existence of a control barrier certificate implying the safety of the controller. For a controlled continuous dynamical system (2), given the sets X, X_U and X_I, if there exists a continuous real-valued function $B(\mathbf{x}) : X \to \mathbb{R}$ satisfying:

$$\begin{aligned} &B(\mathbf{x}) \leq 0 \quad \forall \mathbf{x} \in X_I \\ &B(\mathbf{x}) > 0 \quad \forall \mathbf{x} \in X_U \\ &\dot{B}(\mathbf{x}) < 0 \quad \forall \mathbf{x} \in X \quad s.t. B(\mathbf{x}) = 0, \end{aligned} \quad (4)$$

then $B(\mathbf{x})$ is a control barrier certificate, and the safety of the system is guaranteed. Here, $\dot{B}(\mathbf{x})$ is the Lie derivative of B with respect to a vector field $\mathbf{f}$, defined as follows:

$$\dot{B}(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial B}{\partial x_i} \frac{dx_i}{dt} = \sum_{i=1}^{n} \frac{\partial B}{\partial x_i} f_i(\mathbf{x}, \mathbf{u}). \quad (5)$$

Consider a trajectory $\xi_{\mathbf{h}}^{\mathbf{x}_0}(t)$ and observe the evolution of $B(\xi_{\mathbf{h}}^{\mathbf{x}_0}(t))$ along this trajectory. The initial condition ensures $B(\mathbf{x}_0) \leq 0$, and the final condition requires $B(\xi_{\mathbf{h}}^{\mathbf{x}_0}(t))$ to decrease alone the trajectory $\xi_{\mathbf{h}}^{\mathbf{x}_0}(t)$. As a result, under inputs provided by $\mathbf{h}$, such a trajectory $\xi_{\mathbf{h}}^{\mathbf{x}_0}(t)$ is prevented from entering the unsafe region X_U, where $B(\mathbf{x}) > 0$, so the safety of the controller is guaranteed.

3.3 Problem Statement

Previous works for constructing control barrier certificates and safe controllers often represent them in the form of FNNs. In high-dimensional systems, as the number of layers and nodes in the network increases, the expressions become quite complex, which significantly reduces the efficiency of general SMT methods for verification. To effectively address the challenges of high-dimensional complex systems, we propose using KANs instead of FNNs because they have learnable activation functions, which provide enhanced scalability and expressive power. Additionally, their pruning and symbolization properties can generate simple expressions, making SMT verification more efficient.

Problem 1. *Given a controlled continuous dynamical system S, the problem is to synthesize control barrier certificates and safe controllers using KANs to ensure the safety of the system, achieving dimensional scalability and improved verification efficiency.*

4 Method

In this section, we introduce a novel framework for synthesizing KAN controllers, which can guarantee the safety of the continuous dynamical systems. The overview of the methodology is depicted in Figure 1. The complete process can be divided into the following two stages.

- We devise a Learner component to synthesize candidate control barrier certificates and controllers. This component trains two KANs on a sampled dataset and learns the networks' activation functions by a special loss function.
- We design a Verifier component to formally verify the validity of the candidate KAN control barrier certificates and controllers. This component utilizes pruning and symbolization to simplify the candidate networks, and then uses the SMT solver to verify their validity.

4.1 Kolmogorov-Arnold Network (KAN)

Kolmogorov-Arnold Network (KAN) is a novel network architecture inspired by the Kolmogorov-Arnold representation theorem [Kolmogorov, 1961].

Theorem 1 (Kolmogorov-Arnold representation theorem)**.** *For any continuous function f mapping from $[0, 1]^n$ to the* *real numbers $\mathbb{R}$, there exists a set of continuous functions $\phi_{i,j}$ (where i = 1, 2, . . . , 2n + 1 and j = 1, 2, . . . , n + 1) such that*

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{2n+1} \phi_i(\sum_{j=1}^{n} \phi_{i,j}(\mathbf{x}_j)). \tag{6}$$

According to Equation (6), KAN can be represented as a nested combination of two layers of univariate functions. In matrix form, KAN is defined as:

$$\mathrm{KAN}(\mathbf{x}) = \Phi_{out} \circ \Phi_{in} \circ \mathbf{x}, \tag{7}$$

where Φ_{in} is a matrix composed of univariate functions, represented as:

$$\Phi_{in} = \begin{pmatrix} \phi_{1,1}(\cdot) & \cdots & \phi_{1,n}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{2n+1,1}(\cdot) & \cdots & \phi_{2n+1,n}(\cdot) \end{pmatrix}. \tag{8}$$

and Φ_{out} is a row vector of univariate functions:

$$\Phi_{out} = (\phi_1(\cdot) \quad \ldots \quad \phi_{2n+1}(\cdot)). \tag{9}$$

So the KAN in Equation (7) are simply compositions of two KAN layers. In practical applications, we can achieve deeper KAN by simply stacking more KAN layers. Thus, a L-Layer KAN can be constructed as follows:

$$\mathrm{KAN}(\mathbf{x}) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)\mathbf{x}, \tag{10}$$

where Φ_l is the matrix of activation functions corresponding to the l^{th} KAN layer. A simple KAN with two input neurons and one output neuron is illustrated in Figure 2.

The shape of a KAN can be represented as an integer array $[n_0, n_1, \cdots, n_L]$, where n_i denotes the number of nodes in the i-th layer. Let $x_{l,i}$ denote the i^{th} neuron in the l^{th} layer. There are $n_l n_{l+1}$ activation functions between the l-th layer and the $(l+1)$-th layer. Let $\phi_{l,i,j}$ denote the activation function that connects $x_{l,i}$ and $x_{l+1,j}$. The value of $x_{l+1,j}$ can be calculated as follows:

$$x_{l+1,j} = \sum_{i=1}^{n_l} \phi_{l,i,j}(x_{l,i}) \qquad j = 1, \cdots, n_{l+1}. \tag{11}$$

Different from FNNs, KAN employs learnable activation functions $\phi(\mathbf{x})$ at the network's edges:

$$\phi(\mathbf{x}) = \mathbf{w}_b b(\mathbf{x}) + \mathbf{w}_s \mathrm{spline}(\mathbf{x}), \tag{12}$$

which is the sum of the basis function $b(\mathbf{x})$:

$$b(\mathbf{x}) = \mathrm{silu}(\mathbf{x}) = \frac{\mathbf{x}}{1 + e^{\mathbf{x}}}, \tag{13}$$

and the spline function $\mathrm{spline}(\mathbf{x})$:

$$\mathrm{spline}(\mathbf{x}) = \sum_i c_i B_i(\mathbf{x}), \tag{14}$$

where c_i is trainable, and $B_i(\mathbf{x})$ are B-spline basis functions, and $\mathbf{w}_b$ and $\mathbf{w}_s$ are trainable.

By observing the network structure of KAN, it can fit complex nonlinear relationships with fewer parameters using B-spline functions. This is because B-splines represent global functions through local control, reducing the need for a large number of parameters compared to FNN. By flexibly adjusting the degree k and grid size g, which determine the number of B-splines, KAN can adapt to complex patterns in different data and systems, ensuring the model has strong expressiveness while avoiding overfitting. While maintaining the model's representation power, a smaller grid size can significantly reduce computational complexity, allowing KAN to efficiently handle large-scale datasets. As training progresses, the grid size can also be dynamically adjusted to improve model accuracy without the need to retrain the entire network.

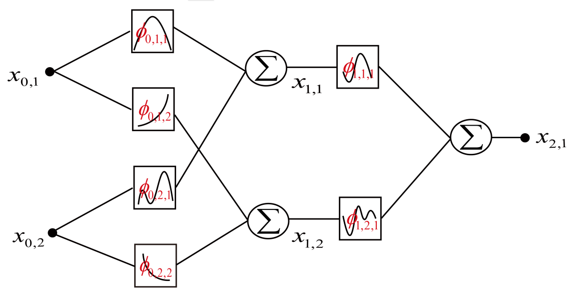

Figure 2: A 2-Layer KAN with shape [2, 2, 1].

Therefore, it is reasonable to use KANs as the networks for synthesizing control barrier certificates and controllers. For the controlled dynamical system S, we use two KANs to separately train a control barrier certificate and a controller. The input dimension of both networks matches the system's dimension, with the output dimension of the barrier certificate network set to 1, and the output dimension of the controller network corresponding to the control input dimension.

4.2 Training of the KAN Barrier and Controller

In this subsection, we will introduce the method for synthesizing KAN control barrier certificates and controllers. Based on the control barrier certificate conditions, we design a loss function for the training dataset. During the training process, the B-spline functions in KANs are updated by minimizing the loss function value of sampled data points.

Training Dataset Construction. Without loss of generality, consider a state set X. We divide X into a finite number of cells $X_1, X_2, \ldots, X_N$ by choosing a discretization parameter ϵ. From each of these cells, we then select sample points $\mathbf{x}_i \in X_i$ such that $\|\mathbf{x} - \mathbf{x}_i\| \leq \epsilon$, for all $\mathbf{x} \in X_i$. Let D denote the set of all these sampled points. Using this method, we can construct the dataset D_I, D_U and D_D, with each batch being sampled from X_I, X_U and X.

Loss Function. We formally define the loss function below:

$$\begin{aligned} L_B = &\sum_{\mathbf{x} \in D_I} \max\{B(\mathbf{x}) + \tau_I\} + \sum_{\mathbf{x} \in D_U} \max\{-B(\mathbf{x}) + \tau_U\} \\ &+ \sum_{\mathbf{x} \in D_D, B(\mathbf{x})=0} \max\{\dot{B}(\mathbf{x}) + \tau_D\}, \end{aligned} \tag{15}$$

where τ_I, τ_U, τ_D serve as offsets that are included to enhance the numerical stability during training. Note that the terms in

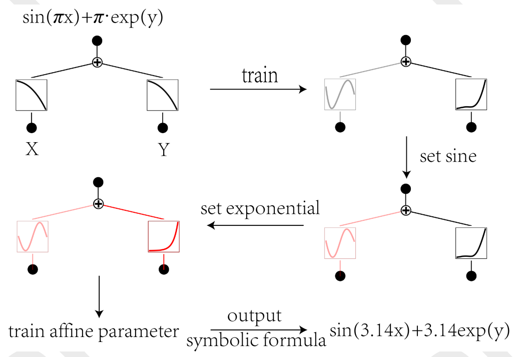

Figure 3: An example of how to do network symbolization.

(15) correspond to the three control barrier certificate conditions, respectively.

To generate a KAN control barrier certificate candidate $B(\mathbf{x})$ and a KAN controller $h(\mathbf{x})$, we employ gradient descent techniques to minimize L_B. When the loss decreases to 0, it indicates that the learned KANs can serve as a candidate control barrier certificate and a candidate controller, respectively.

4.3 KAN Barrier and Controller Verification

In this subsection, we will introduce SMT solving for formally verifying candidate control barrier certificates and controllers represented by KANs, as those generated through gradient descent techniques do not provide formal safety guarantees. Before SMT verification, we will prune and symbolize the networks to accelerate the verification process.

Pruning and Symbolization. To facilitate verification, we prune the trained networks into smaller subnetworks. We perform sparsification [Liu *et al.*, 2024] at the node level, and for the i-th neuron in the l-th layer, we define its incoming and outgoing scores as:

$$I_{l,i} = \max_k(|\phi_{l-1,k,i}|_1),\ O_{l,i} = \max_j(|\phi_{l+1,i,j}|_1). \quad (16)$$

If both the incoming and outgoing scores are greater than the threshold, the node is retained, and the remaining unimportant nodes are pruned. We take node $x_{1,1}$ in Figure 2 as an example, its incoming scores $I_{1,1} = max\{|\phi_{0,1,1}|, |\phi_{0,2,1}|\}$ and outgoing scores $O_{1,1} = |\phi_{1,1,1}|$. If the score of either $I_{1,1}$ or $O_{1,1}$ is less than the threshold, the node, along with its connected edges and activation function, will be removed.

Because B-spline functions are generated recursively, they cannot be formally expressed with explicit expressions, making verification challenging. Since the essence of KAN is to approximate arbitrary functions through the superposition of multiple nonlinear functions, we can transform the activation functions from numerical forms of B-spline to symbolic forms (such as sin, tanh, etc.). In Figure 3, we show the symbolization process through a regression task $f(x, y) = sin(\pi x) + \pi exp(y)$. A one-layer KAN with shape [2,1] is initialized and trained using a dataset $(x_i, y_i, f_i)\ where\ i = 1, 2, \ldots, n$. Then, the activation functions are set to sin and exp, respectively. At this stage, an accurate symbolic formula cannot be obtained because the input and output of the

Algorithm 1 Learning Formally Verified CBC and controller

```
Input: System $S = (\mathbf{f}, X_D, X_I, X_U)$, a discretization parameter $\epsilon$
Output:control barrier certificate $B$ and a safe controller $h$
function LEARNER($S, \epsilon$)
  repeat
    $D \leftarrow$ SampleData($S, \epsilon$)
    compute loss $L_B$, update KANs
  until $L_B = 0$
  return candidate KAN barrier $B$ and controller $h$
end function

function VERIFIER($B, h, iter$)
  if $iter = 1$ then
    $B, h \leftarrow$ Pruning and Symbolification($B, h$)
  end if
  $B, h \leftarrow$ encoding_dReal($B, h$)
  $Cex \leftarrow$ verify_dReal($B, h$)
  if $Cex$ = None then
    return True
  else
    $D \leftarrow D \cup Cex$
  end if
end function

function CEGIS($S$)
  initialise KAN $B$ and $h$, $S, \epsilon$
  repeat
    $iter \leftarrow iter + 1$
    $B, h \leftarrow$ LEARNER($S, \epsilon$)
    $Flag \leftarrow$ VERIFIER($B, h, iter$)
  until $Flag$ = True
  return $B, h$
end function
```

activation functions may have shifts and scalings. Therefore, after further fitting, the coefficient for x and exp is both set to 3.14. The final output symbolic formula $f(x, y) = sin(3.14x) + 3.14exp(y)$ closely matches the expression of the regression task.

In this way, we simplify candidate barrier certificates and controllers through pruning and symbolization for SMT verification.

SMT Verification. We design a counterexample-guided framework based on the SMT solver for verification, which is designed to find states that violate the barrier conditions in (4). To achieve this, we formulate the negation of the three conditions and verify the conditions are unsatisfiable, as follows:

$$\begin{aligned} &\mathbf{x} \in X_I \wedge B(\mathbf{x}) > 0 \\ &\mathbf{x} \in X_U \wedge B(\mathbf{x}) \leq 0 \\ &\mathbf{x} \in X_D \wedge \dot{B}(\mathbf{x}) \geq 0 \wedge B(\mathbf{x}) = 0. \end{aligned} \quad (17)$$

We choose dReal as our SMT verifier, which ensures the correctness of its unsat decisions. Therefore, when dReal returns unsat for the given formula (17), it confirms that no solution exists within the specified precision, and the simplified candidate control barrier certificate $B(\mathbf{x})$ and controller $h(\mathbf{x})$ are valid. Otherwise, it means that dReal has found a counterexample that violates the safety properties of the control barrier certificate.

During the verification process, dReal can only find one counterexample at a time. To improve the efficiency of verification, points around the counterexample will be sampled and added to the dataset to help the learner refine the control barrier certificate and controller. We repeat this process of refinement and verification until no counterexamples are found.

Our main algorithm is illustrated in Algorithm 1.

5 Experiments

We have implemented a tool named KAN4CBC using PyTorch platform for synthesizing KAN control barrier certificates and safe controlllers. We compare our tool against the latest version of FOSSIL2.0 [Edwards *et al.*, 2024] and nncontroller [Zhao *et al.*, 2021]. To compare and illustrate the performance differences between the automatic learning of activation functions and the manual selection of activation functions, we choose sigmoid function as the activation function for FOSSIL2.0, bent-relu function for nncontroller. For each case, we ensure that the number of sampling points is consistent between the three tools. Besides, we use AdamW as the optimization algorithm whose learning rate is 0.001 with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$, and the loss function in Equation (15) with $\tau_I = \tau_U = \tau_D = 0$. We set the hyper-parameter grid size in KAN to 5, and the degree k to 3. The timeout is set to 1 hour, meaning that if a real control barrier certificate and safe controller are not synthesized within 1 hour, the attempt is considered a failure. All experiments are performed on a machine running Ubuntu 24.04 with AMD Ryzen 5 5600G and 16GB memory.

5.1 Case Studies

We evaluate the effectiveness of our tool with examples that include both polynomial and non-polynomial cases. The Ex1 is a 2-dimensional inverted pendulum system [Zhu *et al.*, 2019], the Ex2 is a 3-dimensional ACAS Xu system [Clavière *et al.*, 2021], where a controller is used to control the flight angle of the airplane to prevent collision. The Ex3 is a unicycle system [Yang *et al.*, 2023], where a controller is used to control the unicycle's speed and the angle. The Ex4 [Chesi, 2004], Ex5 [Sassi and Sankaranarayanan, 2015], Ex6 [Zeng *et al.*, 2016], Ex7 [Klipp *et al.*, 2005] and Ex8 [Klipp *et al.*, 2005] are systems with four, five, six, seven, and nine dimensions, respectively. And the Ex1 to Ex3 include non-polynomial terms in the dynamics.

Example 1 (Inverted Pendulum)**.** *The continuous dynamical system is as follows:*

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}x_2\\ 9.8sin(x_1)+u\end{bmatrix}.$$

The domain is $X = \{\mathbf{x} \in \mathbb{R}^2 \mid -\frac{\pi}{4} \le x_1 \le \frac{\pi}{4}, -\frac{\pi}{4} \le x_2 \le \frac{\pi}{4}\}$. Our goal is to generate a safe controller that ensures all trajectories starting from the initial region $X_I = \{\mathbf{x} \in \mathbb{R}^2 \mid -\frac{\pi}{15} \le x_1 \le \frac{\pi}{15}, -\frac{\pi}{15} \le x_2 \le \frac{\pi}{15}\}$ will never enter the unsafe region X_U, which is the complement of $\{\mathbf{x} \mid -\frac{\pi}{6} \le x_1 \le \frac{\pi}{6}, -\frac{\pi}{6} \le x_2 \le \frac{\pi}{6}\}$ in X.

We sample 16 points from the initial region, 256 points from unsafe region, and 1024 points from domain region. Using a two-layer KAN with shape [2, 5, 1], KAN4CBC successfully synthesized the KAN barrier certificate and the safety KAN controller, as illustrated in Figure 4, in which initial regions X_I and unsafe sets X_U are represented in green and red respectively, and the blue line outlines the level curve $B(\mathbf{x}) = 0$. The expressions for the KAN control barrier certificate $B(\mathbf{x})$ and the safe controller $h(\mathbf{x})$ are as follows:

$$\begin{aligned}B(\mathbf{x}) = {} & 0.726 \cdot (0.038 - x_1)^2 - 0.63 \cdot (-x_2 - 0.491)^3\\ & - 0.468 \cdot (0.49 \cdot x_2 - 0.522 \cdot (-x_1 - 0.32)^4 + 1)^2\\ & + 0.275,\end{aligned}$$

$$\begin{aligned}h(\mathbf{x}) = {} & 26.963 \cdot (0.331 - x_2)^2 + 9.01 \cdot (0.345 - x_2)^2\\ & + 0.911 \cdot (1 - 0.989 \cdot x_1)^2 + 4.147 \cdot (1 - 0.984 \cdot x_1)^2\\ & - 9.656.\end{aligned}$$

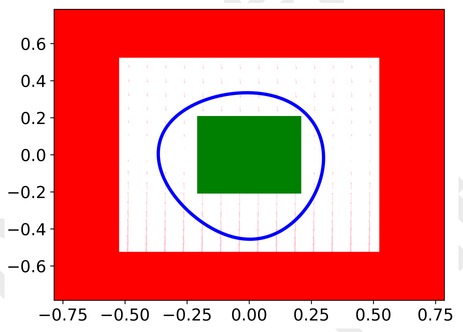

Figure 4: The learned barrier and controller for Example 1

Example 2 (ACAS Xu system)**.** *The continuous dynamical system is as follows:*

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}-0.1sin(\theta)\\ 0.1cos(\theta)-0.1\\ -u\end{bmatrix}.$$

The domain is $X = \{\mathbf{x} \in \mathbb{R}^3 \mid -1 \le x_1 \le 1, -1 \le x_2 \le 1, 0 \le \theta \le \pi\}$. Our goal is to generate a safe controller that ensures all trajectories starting from the initial region $X_I = \{\mathbf{x} \in \mathbb{R}^3 \mid x_1^2 + x_2^2 \le 0.25, 0 \le \theta \le \pi\}$ will never enter the unsafe region $X_U = \{\mathbf{x} \in \mathbb{R}^3 \mid 0.5 \le x_1 \le 0.75, 0.5 \le x_2 \le 1, 0 \le \theta \le \pi\}$.

We sample 125 points each from the initial region and the unsafe region, and 1000 points from the domain region. Using a two-layer KAN with shape [3, 5, 1], KAN4CBC successfully synthesized the KAN barrier certificate and safety KAN controller, as illustrated in Figure 5, in which the zero level set of learned barrier certificate (the yellow surface) separates X_U (the red cuboid) from simulated trajectories starting from X_I (the green ellipsoid). The expressions for the

cases	shape	FOSSIL 2.0				nncontroller				KAN4CBC			
		iter	T$_l$	T$_v$	T$_{sum}$	iter	T$_l$	T$_v$	T$_{sum}$	iter	T$_l$	T$_v$	T$_{sum}$
Ex1	[2,5,1]	41	61.10	0.09	**61.19**	$\times$	$\times$	$\times$	$\times$	1	199.41	0.13	199.54
Ex2	[3,5,1]	41	63.67	31.87	95.54	8	14.02	70.17	**84.19**	3	130.08	1.44	131.52
Ex3	[4,5,1]	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	50	1206.08	14.70	**1220.77**
Ex4	[4,3,1]	10	20.55	152.69	173.23	$\times$	$\times$	$\times$	$\times$	2	149.86	1.21	**151.07**
Ex5	[5,2,1]	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	187	2566.12	39.51	**2605.63**
Ex6	[6,2,1]	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	2	368.33	45.84	**414.17**
Ex7	[7,2,1]	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	2	112.48	8.84	**121.32**
Ex8	[9,1,1]	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	2	1363.12	0.21	**1363.33**

Table 1: Comparative results of the solution in different cases.

KAN control barrier certificate $B(\mathbf{x})$ and the safe controller $h(\mathbf{x})$ are as follows:

$$\begin{aligned}B(\mathbf{x}) = {} & 1.265 \cdot x_1 - 0.072 \cdot x_3 - 0.634(0.18 \cdot x_2 + 0.096 \\ & \cdot x_3 + 0.654 \cdot (0.045 - x_1)^2 - 1)^2 + 0.014cos(8.623 \\ & \cdot x_2 + 2.267) - 0.235,\end{aligned}$$

$$\begin{aligned}h(\mathbf{x}) = {} & -0.28 \cdot x_1 - 1.01 \cdot x_3 + 0.658 \cdot (1 - 0.997 \cdot x_2)^2 \\ & - 0.212 \cdot cos(0.119 \cdot x_1 + 1.485 \cdot x_3 + 1.899 \cdot (1 \\ & - x_2)^2 - 9.11) + 0.436.\end{aligned}$$

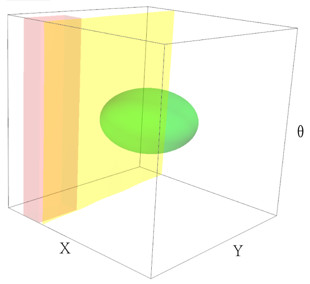

Figure 5: The learned barrier and controller for Example 2

5.2 Results Analysis

Table 1 summarizes the results of our KAN4CBC compared to FOSSIL2.0 and nncontroller in eight different cases. The *shape* represents the KAN structures corresponding to the control barrier certificate and safe controller. To compare the performance between learnable activation functions and fixed activation function, the number of layers and nodes per layer in the neural network within two other methods are kept consistent with those in KAN. The $iter$, T_l, T_v, T_{sum} represent the number of iterations, training time, verification time and the total time required for the method to generate a safe controller in the corresponding case, respectively. And the symbol $\times$ indicates the synthesis of the safe controller failed within one hour.

From Table 1, we conclude that our tool KAN4CBC can successfully synthesize safe controllers for all cases. In contrast, FOSSIL2.0 can only synthesize safe controllers for three cases, and nncontroller succeeds in just one case. Our tool is more scalable because KAN has learnable activation functions, which enhance its expressive power compared to neural networks, enabling it to synthesize safe controllers with fewer network nodes. However, the other two tools lack this capability.

Considering efficiency, our tool generally outperforms in the majority of cases because it controls network complexity through its pruning feature, achieving higher computational efficiency, especially in verification. On average, KAN4CBC has a longer training time per iteration, but a shorter verification time, and the total number of iterations is fewer than FOSSIL2.0 and nncontroller, so its total synthesis time ends up being shorter than two other methods. Only for Ex1 and Ex2 do FOSSIL2.0 and nncontroller synthesize safe controllers faster than our tool, because training neural networks is easier than training KAN's B-spline functions for low-dimensional simple systems. Our method outperforms FOSSIL2.0 and nncontroller on high-dimensional examples, because traditional safe controller synthesis using neural networks requires complex structures for high-dimensional systems, resulting in lengthy expressions that hinder SMT solving. In contrast, the KAN achieves strong expressiveness with fewer layers and nodes, and its pruning and network symbolization significantly improve SMT solving efficiency.

In summary, Table 1 indicates that our method successfully synthesizes safe controllers to ensure the system's safety across all eight benchmark experiments, but FOSSIL2.0 and nncontroller fail. Besides, KAN4CBC requires fewer iterations and less time compared to FOSSIL2.0 and nncontroller in most cases. The results demonstrate that our method exhibits superior efficiency and versatility in synthesizing safe controllers in high-dimensional systems.

6 Conclusion

In this paper, we have proposed a novel and efficient method for formally synthesizing control barrier certificates and controllers to verify the safety of controlled continuous dynamical systems. Our synthesis process consists of two stages: First, it generated candidate control barrier certificates and controllers using KAN which can improve representation power. Second, it simplified the candidate barrier certificates and controller networks through pruning, and then output the candidate expressions by symbolizing the networks which makes SMT verification more efficient and overcomes the limitation of SMT verification failure when applying neural barrier certificates to high-dimensional systems. We demonstrated the effectiveness of our tool using several case studies.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62272416 and Grant 62132014; 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.

References

[Abate *et al.*, 2021] Alessandro Abate, Daniele Ahmed, Alec Edwards, Mirco Giacobbe, and Andrea Peruffo. Fossil: a software tool for the formal synthesis of lyapunov functions and barrier certificates using neural networks. In *Proceedings of the 24th international conference on hybrid systems: computation and control*, pages 1–11, 2021.

[Ahmadi and Majumdar, 2016] Amir Ali Ahmadi and Anirudha Majumdar. Some applications of polynomial optimization in operations research and real-time decision making. *Optimization Letters*, 10:709–729, 2016.

[Andrychowicz *et al.*, 2020] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[Chang *et al.*, 2019] Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

[Chesi, 2004] Graziano Chesi. Computing output feedback controllers to enlarge the domain of attraction in polynomial systems. *IEEE Transactions on Automatic Control*, 49(10):1846–1853, 2004.

[Clavière *et al.*, 2021] Arthur Clavière, Eric Asselin, Christophe Garion, and Claire Pagetti. Safety verification of neural network controlled systems. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 47–54. IEEE, 2021.

[Ding and Tomlin, 2010] Jerry Ding and Claire J Tomlin. Robust reach-avoid controller synthesis for switched nonlinear systems. In *49th IEEE conference on decision and control (CDC)*, pages 6481–6486. IEEE, 2010.

[Edwards *et al.*, 2024] Alec Edwards, Andrea Peruffo, and Alessandro Abate. 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*, pages 1–10, 2024.

[Hu *et al.*, 2024] Hanjiang Hu, Yujie Yang, Tianhao Wei, and Changliu Liu. Verification of neural control barrier functions with symbolic derivative bounds propagation. *arXiv preprint arXiv:2410.16281*, 2024.

[Kapinski *et al.*, 2014] James Kapinski, Jyotirmoy V Deshmukh, Sriram Sankaranarayanan, and Nikos Arechiga. Simulation-guided lyapunov analysis for hybrid dynamical systems. In *Proceedings of the 17th international conference on Hybrid systems: computation and control*, pages 133–142, 2014.

[Klipp *et al.*, 2005] Edda Klipp, Ralf Herwig, Axel Kowald, Christoph Wierling, and Hans Lehrach. *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons, 2005.

[Kolmogorov, 1961] Andreĭ Kolmogorov. *On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables*. American Mathematical Society, 1961.

[Kong *et al.*, 2014] Hui Kong, Xiaoyu Song, Dong Han, Ming Gu, and Jiaguang Sun. A new barrier certificate for safety verification of hybrid systems. *The Computer Journal*, 57(7):1033–1045, 2014.

[Legat *et al.*, 2020] Benoît Legat, Paulo Tabuada, and Raphaël M Jungers. Sum-of-squares methods for controlled invariant sets with applications to model-predictive control. *Nonlinear Analysis: Hybrid Systems*, 36:100858, 2020.

[Leshno *et al.*, 1993] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[Lindemann *et al.*, 2021] Lars Lindemann, Haimin Hu, Alexander Robey, Hanwen Zhang, Dimos Dimarogonas, Stephen Tu, and Nikolai Matni. Learning hybrid control barrier functions from data. In *Conference on robot learning*, pages 1351–1370. PMLR, 2021.

[Liu *et al.*, 2023] Simin Liu, Changliu Liu, and John Dolan. Safe control under input limits with neural control barrier functions. In *Conference on Robot Learning*, pages 1970–1980. PMLR, 2023.

[Liu *et al.*, 2024] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.

[Prajna, 2006] Stephen Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117–126, 2006.

[Sassi and Sankaranarayanan, 2015] Mohamed Amin Ben Sassi and Sriram Sankaranarayanan. Stabilization of polynomial dynamical systems using linear programming based on bernstein polynomials. *arXiv preprint arXiv:1501.04578*, 2015.

[She *et al.*, 2013] Zhikun She, Haoyang Li, Bai Xue, Zhiming Zheng, and Bican Xia. Discovering polynomial lyapunov functions for continuous dynamical systems. *Journal of Symbolic Computation*, 58:41–63, 2013.

[Sogokon *et al.*, 2018] Andrew Sogokon, Khalil Ghorbal, Yong Kiam Tan, and André Platzer. Vector barrier certificates and comparison systems. In *International Symposium on Formal Methods*, pages 418–437. Springer, 2018.

[Yang *et al.*, 2023] Yujie Yang, Yuxuan Jiang, Yichen Liu, Jianyu Chen, and Shengbo Eben Li. Model-free safe reinforcement learning through neural barrier certificate. *IEEE Robotics and Automation Letters*, 8(3):1295–1302, 2023.

[Zeng *et al.*, 2016] Xia Zeng, Wang Lin, Zhengfeng Yang, Xin Chen, and Lilei Wang. Darboux-type barrier certificates for safety verification of nonlinear hybrid systems. In *Proceedings of the 13th International Conference on Embedded Software*, pages 1–10, 2016.

[Zhang *et al.*, 2023a] Hongchao Zhang, Junlin Wu, Yevgeniy Vorobeychik, and Andrew Clark. Exact verification of relu neural control barrier functions. *Advances in neural information processing systems*, 36:5685–5705, 2023.

[Zhang *et al.*, 2023b] Songyuan Zhang, Yumeng Xiu, Guannan Qu, and Chuchu Fan. Compositional neural certificates for networked dynamical systems. In *Learning for Dynamics and Control Conference*, pages 272–285. PMLR, 2023.

[Zhao *et al.*, 2020] Hengjun Zhao, Xia Zeng, Taolue Chen, and Zhiming Liu. Synthesizing barrier certificates using neural networks. In *Proceedings of the 23rd international conference on hybrid systems: Computation and control*, pages 1–11, 2020.

[Zhao *et al.*, 2021] Hengjun Zhao, Xia Zeng, Taolue Chen, Zhiming Liu, and Jim Woodcock. Learning safe neural network controllers with barrier certificates. *Formal Aspects of Computing*, 33:437–455, 2021.

[Zhu *et al.*, 2019] He Zhu, Zikang Xiong, Stephen Magill, and Suresh Jagannathan. An inductive synthesis framework for verifiable reinforcement learning. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, pages 686–701, 2019.