

# $k$ -Inductive Neural Barrier Certificates for Unknown Nonlinear Dynamics

**Ben Wooding**

**Hongchao Zhang**

**Taylor T Johnson**

*Vanderbilt University, Nashville, TN, USA*

BEN.WOODING@VANDERBILT.EDU

HONGCHAO.ZHANG@VANDERBILT.EDU

TAYLOR.JOHNSON@VANDERBILT.EDU

**Abolfazl Lavaei**

*Newcastle University, Newcastle upon Tyne, UK*

ABOLFAZL.LAVAEI@NEWCASTLE.AC.UK

## Abstract

While conventional ( $k = 1$ ) discrete-time barrier certificate conditions impose strict safety constraints by requiring the function to be non-increasing at every step,  $k$ -inductive barrier certificates relax this by allowing a temporary increase—up to  $k - 1$  times, each within a threshold  $\epsilon$ —while maintaining overall safety, and improving flexibility. This paper leverages neural networks and constructs  $k$ -inductive neural barrier certificates ( $k$ -NBCs) for (partially) unknown nonlinear systems. While neural networks offer scalability in the design process, they lack formal guarantees, requiring additional approaches such as counterexample-guided inductive synthesis (CEGIS) with satisfiability modulo theories (SMT) for verification. However, the CEGIS-SMT framework requires knowledge of system dynamics, which is unavailable in practical settings. To address this, we leverage the generalization of the Willems *et al.*’s fundamental lemma, using a *single state trajectory*, to construct a data-driven representation of (partially) unknown models for SMT verification without sacrificing accuracy. Additionally, CEGIS-SMT further removes the constraint of restricting barrier certificates to specific function classes, such as sum-of-squares, enabling greater flexibility in their design. We validate our approach on three nonlinear case studies with (partially) unknown dynamics.

**Keywords:** Data-driven verification, neuro-symbolic AI, barrier certificate, safety, formal methods

## 1. Introduction

Neural barrier certificates (NBCs) are growing in popularity as a technique to provide safety over dynamical systems (Peruffo et al., 2021; Prajna and Jadbabaie, 2004; Qin et al., 2021). NBCs are *neuro-symbolic* (Garcez and Lamb, 2023), integrating neural approaches (neural networks, deep learning, *etc.*) with symbolic approaches (logic, rules, explicit reasoning, *etc.*). One approach to design NBCs is a two-step closed-loop hierarchy: (i) leveraging a neural network as a *function approximator* to generate a candidate that satisfies desired system properties using a collection of data samples, and (ii) verification that the candidate satisfies the property across the whole state space using a *known model* and satisfiability modulo theories (SMT) solvers (De Moura and Bjørner, 2008; Gao et al., 2013). If the second step fails, counterexamples refine the candidate NBC from the first phase, and the loop iterates until a candidate passes the second phase or the process is manually terminated. This loop is known as *counterexample-guided inductive synthesis* (CEGIS).

A key strength of designing NBCs with neural networks is their ability to generate *general nonlinear candidates*, unlike the more common *sum-of-squares* approach, which is restricted to polynomial functions (Prajna et al., 2002; Parrilo and Thomas, 2020). Moreover, verifying an NBC

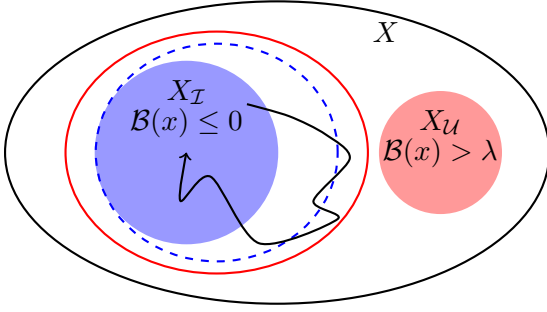


Figure 1: Safety illustration: The black trajectory, starting in the initial region  $X_I \subset X$ , may temporarily increase (crossing the blue-dashed 0-level set) as long as it decreases within  $k$  steps and never exceeds the red level set  $\lambda = (k - 1)\epsilon$ . This ensures it does not reach the unsafe region  $X_U \subset X$ , where  $X_I \cap X_U = \emptyset$ .

candidate using SMT solvers is computationally more efficient than directly searching for a solution via SMT. However, a limitation of SMT-based NBCs is their *dependence on prior knowledge of system dynamics* to guarantee validity on unsampled points.

Data-driven methods offer a promising alternative to deriving safety certificates when the system model is unavailable. Two widely used techniques are the *scenario approach* (Calafiore and Campi, 2006; Campi and Garatti, 2018; Kazemi et al., 2024) and the *single-trajectory approach*, which builds on Willems *et al.*’s fundamental lemma (Willems et al., 2005; De Persis and Tesi, 2019; Verhoek et al., 2021). The scenario approach gives probabilistic guarantees that strengthen with more data, while the single-trajectory approach leverages a *persistently excited* state trajectory to infer system behavior. In this work, we adopt the latter to extend the CEGIS-SMT framework for NBCs, removing the assumption of a known model *a priori*.

This work focuses on discrete-time dynamical systems and introduces a more general class of NBCs, termed  $k$ -inductive neural barrier certificates ( $k$ -NBCs), by leveraging  $k$ -induction (Anand et al., 2021). Specifically, we address system *safety*, ensuring that a system initialized in a predefined region never transitions into an unsafe zone. The  $k$ -induction framework relaxes the strict non-increasing constraint of barrier certificates (Prajna and Jadbabaie, 2004), allowing temporary increases over a finite horizon  $k$  before enforcing a decrease, as illustrated in Figure 1.

### 1.1. Contributions

Our contributions are twofold:

- We construct a data-driven closed-loop model of (partially) unknown system dynamics using the generalization of Willems *et al.*’s fundamental lemma. Since CEGIS-SMT requires a known model for verifying candidate neural barrier certificates, our approach eliminates the need for an *a priori* closed-form system representation, enabling verification with only a single state trajectory.
- We significantly relax the safety conditions with  $k$ -NBCs; removing the non-increasing requirement, and enabling *general* nonlinear barrier certificates for general nonlinear (partially) *unknown* dynamical systems.

### 1.2. Related Work

Prior work has explored the integration of data-driven methods with barrier certificates. Studies such as (Nejati et al., 2023; Salamati et al., 2024; Anand and Zamani, 2023) employ barrier certificates for verification using the scenario approach, while (Nejati et al., 2022; Samari et al., 2024) leverage Willems *et al.*’s fundamental lemma for controller synthesis. However, these approaches

do not incorporate  $k$ -induction or *neural* barrier certificates.  $k$ -inductive barrier certificates have been studied in (Murali et al., 2022) using the scenario and sum-of-squares approaches, but only for linear systems. The closest work to ours is (Wooding and Lavaei, 2024), which designs  $k$ -inductive control barrier certificates for nonlinear dynamics using sum-of-squares. However, that approach relies on *nonlinearity cancellation through control input*, making verification of nonlinear systems infeasible. Furthermore, this work extends  $k$ -NBCs beyond the quadratic barrier certificate constraints imposed in previous approaches, allowing greater flexibility in their design.

NBCs have been explored in (Yang et al., 2023), but in a non-formal manner, achieving “near-zero” constraint violations through reinforcement learning. In (Zhao et al., 2020; Zhang et al., 2023, 2024), NBCs are synthesized using SMT and nonlinear program solvers for controller design with the CEGIS approach. The works (Yang et al., 2023; Zhao et al., 2020; Kong et al., 2026) rely on a known system model and focus on conventional NBCs rather than the relaxed  $k$ -inductive version. In contrast, (Zhou et al., 2022; Hsieh et al., 2025) address black-box systems, eliminating the need for a model in neural approaches. However, (Zhou et al., 2022) assumes knowledge of Lipschitz constants, while (Hsieh et al., 2025) employs CEGIS without SMT solvers and establishes termination guarantees. Notably, both focus on stability rather than barrier certificates, as considered in this work. Other neural-certificate based works include (Debauche et al., 2025; Abate et al., 2020; Mathiesen et al., 2022) and the survey of Dawson et al. (2023). Several software tools have been developed in this area: FOSSIL (Abate et al., 2021; Edwards et al., 2024) for NBCs, PRoTECT (Wooding et al., 2025) for model-based barrier certificates, and TRUST (Gardner et al., 2025) for data-driven barrier certificates.

### 1.3. Organization

The paper is structured as follows: Section 2 defines the (partially) *unknown* system under consideration. Section 3 presents how the generalization of Willems *et al.*’s fundamental lemma is used to represent a single state trajectory as a system model. Section 4 introduces data-driven  $k$ -inductive barrier certificate conditions for nonlinear systems. Section 5 details the use of the CEGIS-SMT framework to generate and verify candidate  $k$ -NBCs for (partially) unknown systems. Section 6 demonstrates the results on a deliberately challenging *highly nonlinear* case study, followed by conclusions in Section 7. Due to space constraints, proofs and additional (physical) case studies are presented in the appendix.

### 1.4. Notation

We use  $\mathbb{R}$ ,  $\mathbb{R}_{\geq 0}$ ,  $\mathbb{R}_{> 0}$ ,  $\mathbb{N}$  and  $\mathbb{N}_{> 0}$  to denote the real numbers, non-negative real numbers, positive real numbers, and the non-negative and positive integers, respectively. We define an  $(n \times n)$  identity matrix by  $\mathbb{I}_n$ . For a state space  $X$ , we define  $X_{\mathcal{I}} \subset X$  and  $X_{\mathcal{U}} \subset X$  as the set of initial states and set of unsafe states, respectively, where  $X_{\mathcal{I}} \cap X_{\mathcal{U}} = \emptyset$ .

## 2. Problem Formulation

**Definition 1** A discrete-time nonlinear system (dt-NS) in this work is described by

$$\Sigma : x^+ = f(x), \quad (1)$$

where  $x^+$  represents the state variables at the next time step, i.e.,  $x^+ := x(k+1)$ ,  $k \in \mathbb{N}$ ,  $x \in X$  is a system state with  $X \subset \mathbb{R}^n$  being the state set, and  $f : X \rightarrow X$  is an unknown transition map.

For the sake of simplicity, one can reformulate the system in (1) as

$$\Sigma : x^+ = A\mathfrak{D}(x), \quad (2)$$

where  $A \in \mathbb{R}^{n \times N}$  is a constant matrix, and  $\mathfrak{D}(x) \in \mathbb{R}^N$  is a vector of nonlinear terms in state  $x$ . We denote by  $x_{x_0}(k)$  the state trajectory of  $\Sigma$  at time  $k \in \mathbb{N}$  starting from an initial condition  $x_0 = x(0)$ .

While we assume that the matrix  $A$  is *unknown*, which reflects practical scenarios, a dictionary  $\mathfrak{D}$  is assumed to be available, constructed to adequately capture the true dynamics by being *sufficiently comprehensive* to encompass all possible terms in the actual system, even if some irrelevant terms are included. We have thus far used the term (*partially*) *unknown* to emphasize the availability of an extensive dictionary together with unknown parameters; however, for simplicity of presentation, we henceforth use the term *unknown*, as the distinction should now be clear from the context. We also note that there are some related studies in the literature that allow  $\mathfrak{D}$  to exclude certain nonlinearities by assuming boundedness of the neglected terms with known bounds; see, e.g., (De Persis et al., 2023; Samari et al., 2026).

Existing works on neural barrier certificates often assume a *fully known* closed-form system model (1), which enables validation through SMT solvers. In the next section, we relax this assumption by adopting a data-driven approach.

### 3. Data-Driven Models

We collect state data from the *unknown* dt-NS over the time horizon  $[0, T]$ , where  $T \in \mathbb{N}_{>0}$  is the number of collected samples:

$$\mathcal{X}_{0,T} = [x(0), x(1), \dots, x(T-1)] \in \mathbb{R}^{n \times T}, \quad (3)$$

$$\mathcal{X}_{1,T} = [x(1), x(2), \dots, x(T)] \in \mathbb{R}^{n \times T}. \quad (4)$$

We also consider the following data-driven representation of the extensive dictionary of nonlinear functions:

$$\mathcal{D}_{0,T} = [\mathfrak{D}(x(0)), \mathfrak{D}(x(1)), \dots, \mathfrak{D}(x(T-1))], \quad (5)$$

where  $\mathcal{D}_{0,T} \in \mathbb{R}^{N \times T}$ . While the data is collected noise-free, the current approach can be extended to noisy data by building upon the results of (Guo et al., 2021).

We first consider the following lemma for the dt-NS system, to obtain a data-based representation of a closed-loop dt-NS (with the continuous-time equivalent by Nejati et al. (2022)).

**Lemma 2** *Let  $Q(x)$  be a  $(T \times N)$  matrix such that*

$$\mathbb{I}_N = \mathcal{D}_{0,T}Q(x), \quad (6)$$

*where  $\mathcal{D}_{0,T}$  is an  $(N \times T)$  full row-rank matrix. The closed-loop system  $x^+ = A\mathfrak{D}(x)$  has the following data-based representation:*

$$x^+ = \mathcal{X}_{1,T}Q(x)\mathfrak{D}(x). \quad (7)$$

**Remark 3** To enforce  $\mathcal{D}_{0,T}$  to be full row-rank, the number of samples  $T$  should be at least  $N$  (De Persis et al., 2023) as a necessary condition. Since this matrix is derived based on sampled data, this rank condition is readily satisfiable.

Since the  $k$ -step evolution for dt-NS is required in our setting, we describe it using a recursive expression in the following theorem.

**Lemma 4** Under Lemma 2, the data-based evolution of a dt-NS after  $k$  steps is described by

$$x^{k+} = f_k(x) = \mathcal{X}_{1,T}Q(f_{k-1}(x))\mathfrak{D}(f_{k-1}(x)), \quad (8)$$

where  $f_1(x) = \mathcal{X}_{1,T}Q(x)\mathfrak{D}(x)$ .

#### 4. $k$ -Inductive Safety Barrier Certificates

We define safety, as illustrated in Figure 1, by the following:

**Definition 5 (Safety)** A dt-NS  $\Sigma$  is safe if all trajectories  $x_{x_0}(k)$  evolving from each state in the initial region  $x_0 \in X_{\mathcal{I}}$  never eventually reach the unsafe region  $X_{\mathcal{U}}$ , i.e.,  $x_{x_0}(k) \notin X_{\mathcal{U}}, \forall k \in \mathbb{N}$ .

We consider  $k$ -inductive safety barrier certificates ( $k$ -BCs) as defined in (Anand et al., 2021). Importantly, we **do not** impose any predefined structure on the  $k$ -BC  $\mathcal{B}(x)$ , such as assuming it to be a quadratic polynomial of the form  $x^\top Px$ , a limitation of prior data-driven work (Wooding and Lavaei, 2024).

##### 4.1. General $k$ -BC Conditions

Building on Lemma 2 and Lemma 4, we establish the  $k$ -BC conditions for data-driven safety guarantees of an unknown dt-NS in the following proposition, adapting the  $k$ -BC definition from (Anand et al., 2021).

**Proposition 6** Given a dt-NS  $\Sigma$  in Definition 1, and data-driven system evolutions (7) and (8), a function  $\mathcal{B} : X \rightarrow \mathbb{R}$  is a  $k$ -inductive barrier certificate ( $k$ -BC) for  $\Sigma$  if there exist  $k \in \mathbb{N}_{>0}$  and  $\epsilon \in \mathbb{R}_{\geq 0}$ , such that

$$\mathcal{B}(x) \leq 0, \quad \forall x \in X_{\mathcal{I}}, \quad (9a)$$

$$\mathcal{B}(x) > (k-1)\epsilon, \quad \forall x \in X_{\mathcal{U}}, \quad (9b)$$

$$\mathcal{B}(\mathcal{X}_{1,T}Q(x)\mathfrak{D}(x)) \leq \mathcal{B}(x) + \epsilon, \quad \forall x \in X \text{ where } \mathcal{B}(x) \leq (k-1)\epsilon, \quad (9c)$$

$$\mathcal{B}(\mathcal{X}_{1,T}Q(f_{k-1}(x))\mathfrak{D}(f_{k-1}(x))) \leq \mathcal{B}(x), \quad \forall x \in X \text{ where } \mathcal{B}(x) \leq 0, \quad (9d)$$

with  $f_1(x) = \mathcal{X}_{1,T}Q(x)\mathfrak{D}(x)$  and  $f_k(x) = \mathcal{X}_{1,T}Q(f_{k-1}(x))\mathfrak{D}(f_{k-1}(x))$ .

When  $k = 1$ , the conditions reduce to those of conventional barrier certificates, implying that  $k$ -BCs with  $k > 1$  are always less restrictive without compromising guarantees. Condition (9c) relaxes the one-step evolution requirement from strict non-negativity to allowing an increase of up to  $\epsilon$ . This relaxation is balanced by condition (9d), which enforces strict non-negativity at  $k$  future

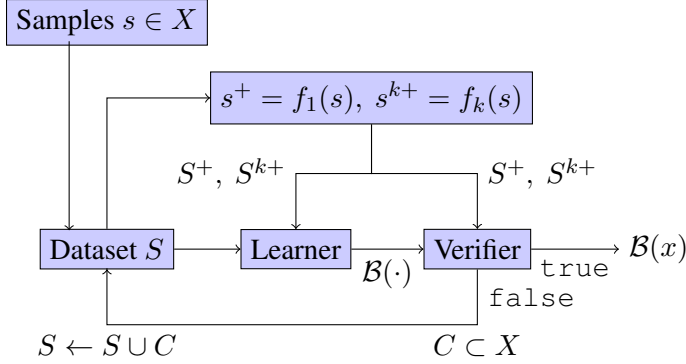


Figure 2: The CEGIS procedure for verification with  $k$ -NBCs. Data points  $s \in S$  are sampled from the state space  $X$ . The *Learner* then generates a candidate  $k$ -NBC  $\mathcal{B}(\cdot)$  based on the dataset triple  $(S, S^+, S^{k+})$ . The *Verifier* checks if the candidate  $k$ -NBC is valid for all states  $x \in X$ . If  $\mathcal{B}(\cdot)$  is invalid, any counterexamples  $c \in C$  are added to the dataset and the *Learner* generates a new candidate. The process is repeated until a valid  $k$ -NBC  $\mathcal{B}(x)$  is found, or the process is manually terminated.

time steps. The term  $(k-1)\epsilon$  in (9b) plays a crucial role in preventing excessive relaxation, ensuring the system trajectory does not reach the unsafe region.

For completeness, we present the following theorem, adapted from (Anand et al., 2021, Theorem 3.2), providing safety guarantees for discrete-time dynamical systems by leveraging the notion of  $k$ -BCs from Proposition 6.

**Theorem 7** *Given a dt-NS  $\Sigma$  and some  $k \in \mathbb{N}$ , suppose  $\mathcal{B}(x)$  is a  $k$ -BC for  $\Sigma$  as in Proposition 6. Then, the state trajectory  $x_{x_0}(k)$  remains safe (i.e.,  $x_{x_0}(k) \notin X_{\mathcal{U}}, \forall k$ ) for any  $x_0 \in X_{\mathcal{I}}$  with data-based dynamic representation satisfying conditions (9c) and (9d).*

Note that conditions (9c)-(9d) are not enforced over all  $x \in X$ , as Theorem 7 ensures forward invariance, system trajectories outside this set can be ignored. The  $k = 2$  special case has (9c)-(9d) both quantified for  $x \in X$  where  $\mathcal{B}(x) \leq 0$ . Sum-of-squares approaches often impose the evolution condition on all states for computational simplicity, whereas SMT solvers enable this additional relaxation. We also present the theory applied to the less-general linear systems in Appendix B, showing simpler conditions can be utilized.

## 5. $k$ -Inductive Neural Barrier Certificates

We now present the main contribution of this work. We adapt the CEGIS-SMT framework to generate  $k$ -inductive neural safety barrier certificates ( $k$ -NBCs) that relax the assumption of knowing a closed-form model of  $\Sigma$ . We highlight again that to our knowledge, this is the first work designing and verifying NBCs using CEGIS-SMT entirely from data.

### 5.1. CEGIS Procedure

We first describe the CEGIS procedure as visualized in Figure 2. At a high level, CEGIS has two main *neuro-symbolic* components: a *learner* (neural component) and a *verifier* (symbolic component). The learner generates a candidate  $k$ -NBC based on the sampled data points. Then the verifier checks if the candidate  $k$ -NBC is valid for the whole state space. If the candidate  $k$ -NBC is invalid, counterexamples  $C$  are added to the dataset and the learner generates a new candidate. This process

is repeated until a valid  $k$ -NBC is found, or the process is manually terminated. We notate  $\mathcal{B}(\cdot)$  for a candidate  $k$ -NBC and  $\mathcal{B}(x)$  for a verified  $k$ -NBC.

**Samples and Dataset.** The state space  $X$  is sampled for some positive integer  $m$  number of data points  $s$ . The collection of these data points forms the dataset  $S$ .

**System Evolution.** To verify the conditions of Proposition 6, it is necessary to evaluate the 1-step and  $k$ -step evolutions of the system. For each data point  $s \in S$ , we acquire the state evolutions  $s^+$  and  $s^{k+}$  by evolving the system for 1 or  $k$  steps from initial state  $s$ , respectively. Collectively these datasets form a dataset triple  $(S, S^+, S^{k+})$  that is used by the learner.

**Learner.** The learner trains a neural network to approximate the  $k$ -NBC function. When the loss function is optimally minimized, the resulting function serves as a candidate  $k$ -NBC. The candidate  $k$ -NBC  $\mathcal{B}(\cdot)$  is trained to satisfy the conditions of Proposition 6 for the dataset triple  $(S, S^+, S^{k+})$ . Although the candidate satisfies the conditions of Proposition 6 for all  $s \in S$ , it is not necessarily valid for all states  $x \in X$  and therefore requires verification.

**Verifier.** The verifier formally checks if the candidate  $\mathcal{B}(\cdot)$  violates the conditions of Proposition 6 within the whole domain  $X$ . If a violation is detected, the verifier generates a set of counterexamples  $C \subset X$  that includes at least one counterexample  $c$ . The set  $C$  is then appended to the original dataset  $S$  and System Evolution and Learner steps are repeated, encouraging the learner to generate a new candidate  $k$ -NBC for verification. This loop continues indefinitely until it is either manually terminated or the verifier confirms that no counterexamples exist, thereby proving the candidate  $k$ -NBC is valid across the entire state space, ensuring  $\mathcal{B}(\cdot) = \mathcal{B}(x)$ .

Verifying the  $k$ -CBC candidate over the entire state space  $X$  using SMT solvers has required knowledge of the closed-form underlying dynamical system (1). By leveraging Lemma 2 and Lemma 4, we relax this requirement and instead use the model (7), which is derived solely from a single data trajectory.

## 5.2. Training the Neural Network

A neural network functions as an approximator by processing inputs through a series of weighted operations, biases, and activation functions to generate the desired output. The first main component of the CEGIS-SMT procedure is the *learner*, which assigns weights and biases to the neural network according to a given loss function. The learner designs a candidate  $k$ -CBC using the hyper-parameters  $d$  and  $h_1, \dots, h_d$ , representing the depth and width of the neural network, where every connection between nodes has a weight and a bias. The learner optimizes weights and biases using backpropagation, via the Adam optimizer (Kingma and Ba, 2014), over the dataset triple  $(S, S^+, S^{k+})$ . It refines its training whenever the verifier includes new counterexamples.

The training procedure minimizes a loss function that is designed to satisfy the four conditions of Proposition 6. Conditions (9a) and (9b) apply only to specific subsets of the state space. Therefore, the relevant data points should be filtered accordingly, denoted as  $S_{\mathcal{I}}$  and  $S_{\mathcal{U}}$ , respectively. The loss function  $\mathcal{L}$  to minimize is then constructed as:

$$\mathcal{L} = \mathcal{L}_{\mathcal{I}} + \mathcal{L}_{\mathcal{U}} + \mathcal{L}_1 + \mathcal{L}_k, \quad (10)$$

where:

$$\begin{aligned}\mathcal{L}_{\mathcal{I}} &= \frac{1}{|S_{\mathcal{I}}|} \sum_{s \in S_{\mathcal{I}}} \text{ReLU}(\mathcal{B}(s) + \eta_1), \\ \mathcal{L}_{\mathcal{U}} &= \frac{1}{|S_{\mathcal{U}}|} \sum_{s \in S_{\mathcal{U}}} \text{ReLU}(-\mathcal{B}(s) + (k-1)\epsilon + \eta_2), \\ \mathcal{L}_1 &= \frac{1}{|S|} \sum_{s \in S} \text{ReLU}(\mathcal{B}(\mathcal{X}_{1,T}Q(s)\mathfrak{D}(s)) - \mathcal{B}(s) - \epsilon + \eta_3), \\ \mathcal{L}_k &= \frac{1}{|S|} \sum_{s \in S} \text{ReLU}(\mathcal{B}(\mathcal{X}_{1,T}Q(f_{k-1}(s))\mathfrak{D}(f_{k-1}(s))) - \mathcal{B}(s) + \eta_4),\end{aligned}$$

with the function  $\text{ReLU}(\cdot)$  and  $\eta_i \geq 0$ ,  $i = \{1, \dots, 4\}$ , are tuning parameter that may increase the conservativeness of the desired function. Iterations of backpropagation are then performed until the value of the loss function is sufficiently small (ideally zero). Note that the loss functions  $\mathcal{L}_1$  and  $\mathcal{L}_k$  are applied across the entire state space. This is necessary because, until the function is generated, the states satisfying  $\mathcal{B}(x) \leq 0$  cannot be determined.

The loss function may not always converge to zero. As a practical strategy, training under more conservative conditions can lead to a function that meets the requirements of Proposition 6, even if the loss does not reach zero.

### 5.3. Verifying the Candidate

The second component of the CEGIS procedure is the *verifier*, which verifies a candidate is a valid  $k$ -NBC for all states  $x$  in the state space, rather than only the sampled states  $s$  as trained. Although a closed-form model of (1) is unavailable, the single-trajectory approach gives a data-driven model (7) that serves as a suitable substitute for the true system, given that the rank condition ensures persistently excited data. Since the structure, weights, biases, and activation functions of the neural network are known, a symbolic expression of the candidate  $k$ -NBC  $\mathcal{B}(\cdot)$  can be extracted.

The goal of the SMT solver is to verify if this candidate is a valid  $k$ -NBC for the full state space, and if not to generate appropriate counterexamples. This check is performed by negating the conditions of Proposition 6 and trying to solve for any solution. As the conditions are negated, a solution to the SMT solver is a generated counterexample, which can be passed to the learner to find a new candidate. If no counterexample can be found, the  $k$ -NBC  $\mathcal{B}(x)$  is verified over the whole state space.

In natural language, we check if for a given state space  $X$ , any of the conditions in Proposition 6 are violated. Using implication, and negating every condition, the conditions can be checked based on the regions where they apply. The two evolution conditions can be combined as they refer to the same set of states. The logical formula we check to find a counterexample can be written formally as follows:

$$\text{cex} = (x \in X) \wedge (\neg \mathbf{V}_I \vee \neg \mathbf{V}_U \vee \neg \mathbf{V}_{e1} \vee \neg \mathbf{V}_{e2}), \quad (11)$$

where

$$\begin{aligned}\mathbf{V}_I &:= x \in X_I \implies \mathcal{B}(x) \leq 0, & \mathbf{V}_U &:= x \in X_U \implies \mathcal{B}(x) > (k-1)\epsilon, \\ \mathbf{V}_{e1} &:= \mathcal{B}(x) \leq (k-1)\epsilon \implies (\mathcal{B}(\mathcal{X}_{1,T}Q(x)\mathfrak{D}(x))) \leq \mathcal{B}(x) + \epsilon \\ \mathbf{V}_{e2} &:= \mathcal{B}(x) \leq 0 \implies (\mathcal{B}(\mathcal{X}_{1,T}Q(f_{k-1}(x))\mathfrak{D}(f_{k-1}(x)))) \leq \mathcal{B}(x).\end{aligned}$$

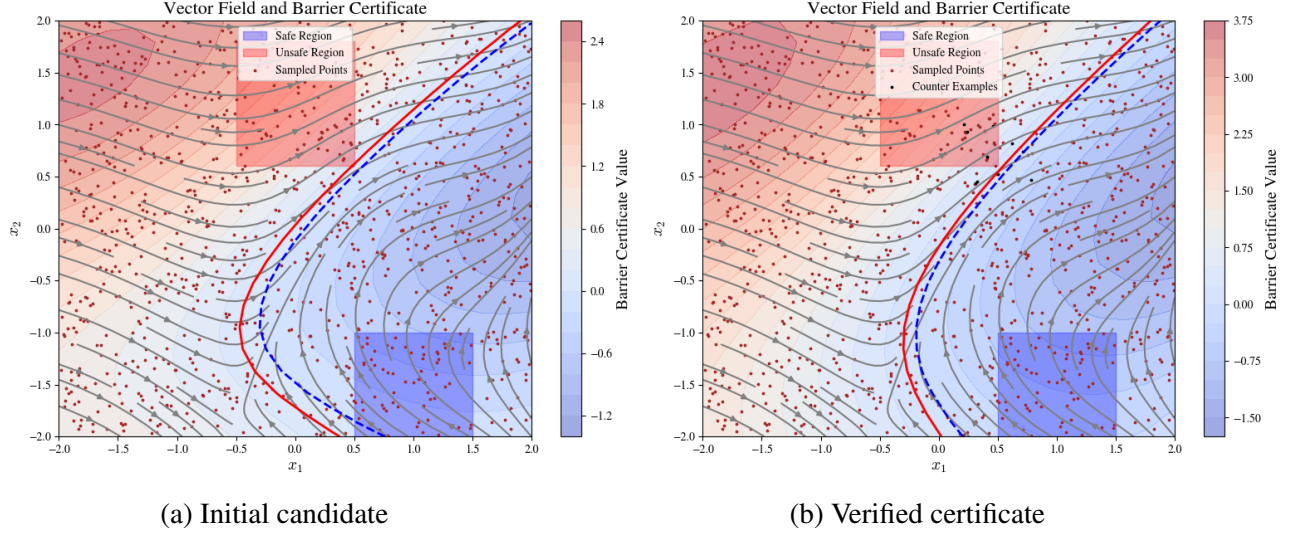


Figure 3:  $k$ -NBC results for the *highly nonlinear* system. (a) *Candidate*  $k$ -NBC showing violations where  $X_I$  and  $X_U$  intersect the level sets. Sampled points are depicted in brown. (b) *Verified*  $k$ -NBC after retraining with counterexamples (black points). Both show the zero level set in dashed blue, unsafe level set in red, and true vector field in gray.

If the above logical formula is satisfied, the SMT solver returns `sat` along with the *identified counterexample*. If the solver returns `unsat`, *no counterexample exists*, confirming that the  $k$ -NBC is valid for all  $x \in X$ .

## 6. Simulation Results

We demonstrate the efficacy of our approach through three case studies: a *nonlinear polynomial* system, a *physical pendulum* system, and a *highly nonlinear* system. Due to space constraints, only the highly nonlinear system is presented in the main body of the paper as it is the most complex. Both the nonlinear polynomial system and the physical pendulum system are presented in Appendix C.

### 6.1. Highly Nonlinear System

We present our results for the following academic system that is designed to demonstrate the generality of our approach.

**System Description.** We consider a discrete-time highly nonlinear system as

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \Delta t \begin{bmatrix} x_2 + e^{-x_1} + \sin^2(x_1) \\ x_1 - \sin^2(x_1) + \cos^2(x_1) \end{bmatrix},$$

where the sampling time  $\Delta t$  is 0.1 seconds. The model is assumed to be *unknown*, but an extensive dictionary is available:  $\mathcal{D}(x) = [x_1, x_2, e^{-x_1}, e^{-x_2}, \sin^2(x_1), \cos^2(x_1)]$ . We consider a safety specification over the state space  $X = [-2, 2]^2$  with an initial region  $X_I = [0.5, 1.5] \times [-2, -1]$  and unsafe region  $X_U = [-0.5, 0.5] \times [0.6, 1.8]$ . As our dictionary has 6 terms, we consider a single state trajectory of length 7, from initial state  $x_0 = [0.5, -1]$ . Using the datasets  $\mathcal{X}_{0,T}$ ,  $\mathcal{X}_{1,T}$ , and  $\mathcal{D}_{0,T}$ , we solve (6), obtaining the data-driven model  $x^+ = \mathcal{X}_{1,T}Q(x)\mathcal{D}(x)$ .

**Initial Candidate.** We aim to find a  $k$ -NBC for this system, with  $k = 2$  and  $\epsilon = 0.1$ . We fix the parameter  $\eta_2 = 0.001$ . We sample 1000 data points from the data-driven model to form our dataset triple  $(S, S^+, S^{k+})$ . We also consider a neural network with two inputs  $x_1, x_2$  (one per dimension), one hidden layer containing four nodes, and one output which is the evaluation of our candidate  $k$ -NBC. Each node  $y$  evaluates the formula  $y = g(b + \sum_{i=1}^2 x_i w_i)$ , where, for each node,  $b$  represents the bias,  $w_i$  are the weights, and the activation function  $g(\cdot)$  is  $\sin(\cdot)$  for two hidden nodes and  $\cos(\cdot)$  for the other two hidden nodes. For training, backpropagation is performed using the Adam optimizer (Kingma and Ba, 2014), running for 1000 epochs with a learning rate of 0.1. Verification with dReal (Gao et al., 2013) identified a counterexample, with  $\delta$ -bound set to 0.001. As seen in Figure 3(a), the candidate  $k$ -NBC is invalid, as parts of the region  $X_{\mathcal{I}}$  have barrier values above the blue-dashed zero level set. Additionally, the lower right corner of region  $X_{\mathcal{U}}$  has  $k$ -NBC value below the red level set.

**Retraining.** From the range of the counterexample, 20 points within a 0.1 radius are appended to the dataset for retraining. The model is retrained for another 1000 epochs with a reduced learning rate of 0.05. On this iteration, the SMT solver *did not* return a counterexample — verifying the  $k$ -NBC  $\mathcal{B}(x)$  as valid for all  $x$ , as seen in Figure 3(b). All results were generated and verified using the data-driven model. Allowing for rounding, the  $k$ -NBC is designed as the following nonlinear non-polynomial function:

$$\begin{aligned} \mathcal{B}(x) = & -0.55 \sin(0.54x_1 - 1.32x_2 + 1.14) - 1.35 \sin(0.58x_1 - 0.47x_2 + 0.29) \\ & + 0.65 \cos(0.72x_1 - 0.06x_2 + 1.40) + 0.12 \cos(0.80x_1 - 0.05x_2 + 1.31) + 0.99. \end{aligned}$$

**Remark 8** *It naturally follows that using a larger neural network with a broader range of activation functions can further enhance the nonlinearity of  $\mathcal{B}(x)$ .*

**Conventional Barrier Certificate Solution.** We investigate whether the solved  $\mathcal{B}(x)$  satisfies the conventional barrier certificate conditions (where  $k = 1$  and  $\epsilon = 0$ ). However, dReal identifies a counterexample within the range  $[0.6389, 0.6391] \times [-2, -1.9995]$ . Consequently, only by applying the  $k$ -inductive conditions from Proposition 6 could we obtain a valid relaxed solution.

As a general observation, across various simulations, we found that the  $k$ -NBCs require fewer iterations than conventional NBCs and, in several cases, could identify a valid  $k$ -NBC on the first attempt.

## 7. Conclusion

We proposed a data-driven method for constructing  $k$ -inductive neural barrier certificates and verify them using the neuro-symbolic CEGIS-SMT framework, without requiring an *a priori* known model of the underlying system dynamics. By leveraging persistently excited data, a single state trajectory was used to derive a data-driven model of the dynamics, ensuring validity for verification with the SMT solver. Additionally,  $k$ -induction was employed as a relaxation of conventional discrete-time barrier certificate conditions, and our experiments demonstrated that it can find certificates that conventional barrier certificates fail to design. Neural approaches enabled the discovery of *general* nonlinear barrier certificates for discrete-time systems with *unknown* general nonlinear dynamics. While our paper broadens the applicability of CEGIS-SMT, the existing scalability limitations of the framework remain; nevertheless, the pipeline can be extended by replacing the SMT solver with alternative verification tools (e.g., scenario approach). Future work can consider  $t$ -barriers, a challenging continuous-time analogy to  $k$ -NBCs requiring derivative relaxations (Bak, 2018).

## Acknowledgments

The material presented in this paper is based upon work supported by the National Science Foundation (NSF) through grant number 2220401 and the Defense Advanced Research Projects Agency (DARPA) under contract number FA8750-23-C-0518. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA or NSF. Early stages of this work were supported by an EPSRC Doctoral Prize Research Fellowship.

## References

- Alessandro Abate, Daniele Ahmed, Mirco Giacobbe, and Andrea Peruffo. Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2020.
- 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.
- Mahathi Anand and Majid Zamani. Formally verified neural network control barrier certificates for unknown systems. *IFAC-PapersOnLine*, 56(2):2431–2436, 2023.
- Mahathi Anand, Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. Safety verification of dynamical systems via k-inductive barrier certificates. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 1314–1320, 2021. doi: 10.1109/CDC45484.2021.9682889.
- Stanley Bak. t-barrier certificates: a continuous analogy to k-induction. *IFAC-PapersOnLine*, 51(16):145–150, 2018.
- Giuseppe C. Calafiore and Marco C. Campi. The scenario approach to robust control design. *IEEE Transactions on automatic control*, 51(5):742–753, 2006.
- Marco C. Campi and Simone Garatti. *Introduction to the scenario approach*. SIAM, 2018.
- Charles Dawson, Sicun Gao, and Chuchu Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 39(3):1749–1767, 2023.
- Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.
- Claudio De Persis and Pietro Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924, 2019.
- Claudio De Persis, Monica Rotulo, and Pietro Tesi. Learning controllers from data via approximate nonlinearity cancellation. *IEEE Transactions on Automatic Control*, 68(10):6082–6097, 2023.

- Virginie Debauche, Alec Edwards, Raphael M Jungers, and Alessandro Abate. Formal Synthesis of Lyapunov Stability Certificates for Linear Switched Systems using ReLU Neural Networks. In *International Conference on Neuro-symbolic Systems*, pages 352–364. PMLR, 2025.
- 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.
- Sicun Gao, Soonho Kong, and Edmund M Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *International conference on automated deduction*, pages 208–214. Springer, 2013.
- Artur d’Avila Garcez and Luis C Lamb. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.
- Jamie Gardner, Ben Wooding, Amy Nejati, and Abolfazl Lavaei. TRUST: Stability and Safety ContRoller Synthesis for Unknown Dynamical Models Using a Single Trajectory. In *Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–16, 2025.
- Meichen Guo, Claudio De Persis, and Pietro Tesi. Data-driven stabilization of nonlinear polynomial systems with noisy data. *IEEE Transactions on Automatic Control*, 67(8):4210–4217, 2021.
- Chiao Hsieh, Masaki Waga, and Kohei Suenaga. Certifying Lyapunov Stability of Black-Box Nonlinear Systems via Counterexample Guided Synthesis. In *Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control*, pages 1–11, 2025.
- Milad Kazemi, Rupak Majumdar, Mahmoud Salamati, Sadegh Soudjani, and Ben Wooding. Data-driven abstraction-based control synthesis. *Nonlinear Analysis: Hybrid Systems*, 52:101467, 2024.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Chun-Wei Kong, Sebastian Escobar, Ibon Gracia, Jay McMahon, and Morteza Lahijanian. Training with Hard Constraints: Learning Neural Certificates and Controllers for SDEs. *arXiv preprint arXiv:2602.23526*, 2026.
- Frederik Baymler Mathiesen, Simeon C Calvert, and Luca Laurenti. Safety certification for stochastic systems via neural barrier functions. *IEEE Control Systems Letters*, 7:973–978, 2022.
- Vishnu Murali, Ashutosh Trivedi, and Majid Zamani. A scenario approach for synthesizing k-inductive barrier certificates. *IEEE Control Systems Letters*, 6:3247–3252, 2022.
- Ameneh Nejati, Bingzhuo Zhong, Marco Caccamo, and Majid Zamani. Data-driven controller synthesis of unknown nonlinear polynomial systems via control barrier certificates. In *Learning for Dynamics and Control Conference*, pages 763–776. PMLR, 2022.
- Ameneh Nejati, Abolfazl Lavaei, Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. Formal verification of unknown discrete-and continuous-time systems: A data-driven approach. *IEEE Transactions on Automatic Control*, 68(5):3011–3024, 2023.

- Pablo A Parrilo and Rekha R Thomas. *Sum of Squares: Theory and Applications*, volume 77. American Mathematical Soc., 2020.
- Andrea Peruffo, Daniele Ahmed, and Alessandro Abate. Automated and formal synthesis of neural barrier certificates for dynamical models. In *International conference on tools and algorithms for the construction and analysis of systems*, pages 370–388. Springer, 2021.
- Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *International workshop on hybrid systems: Computation and control*, pages 477–492. Springer, 2004.
- Stephen Prajna, Antonis Papachristodoulou, and Pablo A Parrilo. Introducing SOSTOOLS: A general purpose sum of squares programming solver. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 1, pages 741–746. IEEE, 2002.
- Zengyi Qin, Kaiqing Zhang, Yuxiao Chen, Jingkai Chen, and Chuchu Fan. Learning safe multi-agent control with decentralized neural barrier certificates. In *International Conference on Learning Representations*, 2021.
- Ali Salamati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven verification and synthesis of stochastic systems via barrier certificates. *Automatica*, 159:111323, 2024.
- Behrad Samari, Omid Akbarzadeh, Mahdieh Zaker, and Abolfazl Lavaei. From a single trajectory to safety controller synthesis of discrete-time nonlinear polynomial systems. *IEEE Control Systems Letters*, 2024.
- Behrad Samari, Henrik Sandberg, Karl H Johansson, and Abolfazl Lavaei. Data-driven model order reduction for continuous-and discrete-time nonlinear systems. *arXiv:2507.18131v2*, 2026.
- Chris Verhoek, Roland Tóth, Sofie Haesaert, and Anne Koch. Fundamental lemma for data-driven analysis of linear parameter-varying systems. In *2021 60th IEEE conference on decision and control (CDC)*, pages 5040–5046. IEEE, 2021.
- Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.
- Ben Wooding and Abolfazl Lavaei. Learning  $k$ -inductive control barrier certificates for unknown nonlinear dynamics beyond polynomials. *arXiv preprint arXiv:2412.07232*, 2024.
- Ben Wooding, Viacheslav Horbanov, and Abolfazl Lavaei. PRoTECT: Parallelized construction of safety barrier certificates for nonlinear polynomial systems. In *International Colloquium on Theoretical Aspects of Computing*, pages 448–458. Springer, 2025.
- 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.
- 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.

Hongchao Zhang, Zhizhen Qin, Sicun Gao, and Andrew Clark. SEEV: Synthesis with Efficient Exact Verification for ReLU Neural Barrier Functions. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 101367–101392. Curran Associates, Inc., 2024. doi: 10.52202/079017-3214.

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.

Ruikun Zhou, Thanin Quartz, Hans De Sterck, and Jun Liu. Neural Lyapunov control of unknown nonlinear systems with stability guarantees. *Advances in Neural Information Processing Systems*, 35:29113–29125, 2022.

## Appendix A. Proofs

### A.1. Proof for Lemma 2

The closed-loop dt-NS can be written as

$$A\mathfrak{D}(x) = A \overbrace{\mathcal{D}_{0,T}Q(x)}^{\mathbb{I}_N} \mathfrak{D}(x) = \mathcal{X}_{1,T}Q(x)\mathfrak{D}(x),$$

with  $\mathcal{X}_{1,T} = A\mathcal{D}_{0,T}$  as in (5). Therefore,  $x^+ = \mathcal{X}_{1,T}Q(x)\mathfrak{D}(x)$  is the data-based representation of the closed loop dt-NS, completing the proof. ■

### A.2. Proof for Lemma 4

Since  $x^{k+}$  is obtained by applying  $k$  iterations of  $A\mathfrak{D}(x)$ , it follows that

$$\begin{aligned} x^+ &= A\mathfrak{D}(x) = \mathcal{X}_{1,T}Q(x)\mathfrak{D}(x) = f_1(x), \\ x^{2+} &= \mathcal{X}_{1,T}Q(x^+)\mathfrak{D}(x^+) = \mathcal{X}_{1,T}Q(f_1(x))\mathfrak{D}(f_1(x)) = f_2(x), \end{aligned}$$

and by induction, we have

$$x^{k+} = \mathcal{X}_{1,T}Q(f_{k-1}(x))\mathfrak{D}(f_{k-1}(x)) = f_k(x),$$

completing the proof. ■

### A.3. Proof for Proposition 6

Proposition 6 builds upon the definition of  $k$ -NBCs found in (Anand et al., 2021):

**Definition 9** ( $k$ -BC (Anand et al., 2021)) *We say that a function  $\mathcal{B} : X \rightarrow \mathbb{R}_{\geq 0}$  is a  $k$ -inductive barrier certificate for the system  $\Sigma$  with respect to a set of initial states  $X_{\mathcal{I}} \subseteq X$  and a set of unsafe states  $X_{\mathcal{U}} \subseteq X$ , if there exist  $k \in \mathbb{N}_{>0}$  and  $\epsilon \in \mathbb{R}_{\geq 0}$ , such that following conditions hold:*

$$\begin{aligned} \mathcal{B}(x) &\leq 0, & \forall x \in X_{\mathcal{I}}, \\ \mathcal{B}(x) &> (k-1)\epsilon, & \forall x \in X_{\mathcal{U}}, \\ \mathcal{B}(f_1(x)) - \mathcal{B}(x) &\leq \epsilon, & \forall x \in X, \\ \mathcal{B}(f_k(x)) - \mathcal{B}(x) &\leq 0, & \forall x \in X. \end{aligned} \tag{12a}$$

$$\tag{12b}$$

Substituting the result of Lemma 2 into (12a) and Lemma 4 into (12b) gives conservative certificate conditions. To achieve our further relaxed conditions, we enforce (12a) and (12b) when  $\mathcal{B}(x) \leq (k-1)\epsilon$  and  $\mathcal{B}(x) \leq 0$ , respectively — enforcing the condition to hold only for states reachable from the initial set  $X_{\mathcal{I}}$ . ■

#### A.4. Proof for Theorem 7

By condition (9a) and (9b), the initial and unsafe regions are distinctly separated by two level sets of  $\mathcal{B}(x)$ , with the initial condition level set (9a) of lower value than the unsafe region's level set.

Condition (9d) enforces the system to be strictly decreasing in  $\mathcal{B}(x)$  value after  $k$  steps, while a one step transition may increase by a value of  $\epsilon$  (9c). The maximal increase in  $\mathcal{B}(x)$  value as the system evolves is therefore  $k-1$  increases by a value of  $\epsilon$ . As the unsafe region value is at least  $(k-1)\epsilon$ , and the largest initial value is 0, a  $k$ -BC certificate always guarantees safety. ■

## Appendix B. Linear Systems

We separate the discrete-time linear system (dt-LS) class from the more general dt-NS class presented in the main paper. For dt-LS, the  $k$ -step evolution of the system has a simple closed-form solution for any  $k$ . We consider the following lemma for the dt-LS case, *i.e.*  $\mathfrak{D}(x) = x$ , to obtain a data-based representation of a closed-loop dt-LS.

**Lemma 10** *Let matrix  $Q$  be a  $(T \times n)$  matrix such that*

$$\mathbb{I}_n = \mathcal{X}_{0,T}Q, \quad (13)$$

*where  $\mathcal{X}_{0,T}$  is an  $(n \times T)$  full row-rank matrix. The closed-loop system  $x^+ = Ax$  has the following data-based representation:*

$$x^+ = \mathcal{X}_{1,T}Qx, \text{ equivalently, } A = \mathcal{X}_{1,T}Q.$$

*Additionally, the data-based evolution of the system after  $k$  steps is*

$$x^{k+} = (\mathcal{X}_1Q)^k x. \quad (14)$$

**Proof** The proof is straightforward following the proofs of Lemma 2 and Lemma 4. ■

**Remark 11** *Similar to the dt-NS case, to enforce  $\mathcal{X}_{0,T}$  to be full row-rank, the number of samples  $T$  should be at least  $n$ .*

For dt-LSs, the  $k$ -BC conditions leverage Lemma 10, yielding a simple closed-form expression for the last two conditions.

**Proposition 12** *Given a dt-LS  $\Sigma$ , a function  $\mathcal{B} : X \rightarrow \mathbb{R}$  is a  $k$ -inductive barrier certificate ( $k$ -BC) for  $\Sigma$  if there exist  $k \in \mathbb{N}_{>0}$ ,  $\epsilon \in \mathbb{R}_{\geq 0}$ , such that*

$$\mathcal{B}(x) \leq 0, \quad \forall x \in X_{\mathcal{I}}, \quad (15a)$$

$$\mathcal{B}(x) > (k-1)\epsilon, \quad \forall x \in X_{\mathcal{U}}, \quad (15b)$$

$$\mathcal{B}(\mathcal{X}_1Qx) \leq \mathcal{B}(x) + \epsilon, \quad \forall x \in X \text{ where } \mathcal{B}(x) \leq (k-1)\epsilon, \quad (15c)$$

$$\mathcal{B}((\mathcal{X}_1Q)^k x) \leq \mathcal{B}(x) \quad \forall x \in X \text{ where } \mathcal{B}(x) \leq 0. \quad (15d)$$

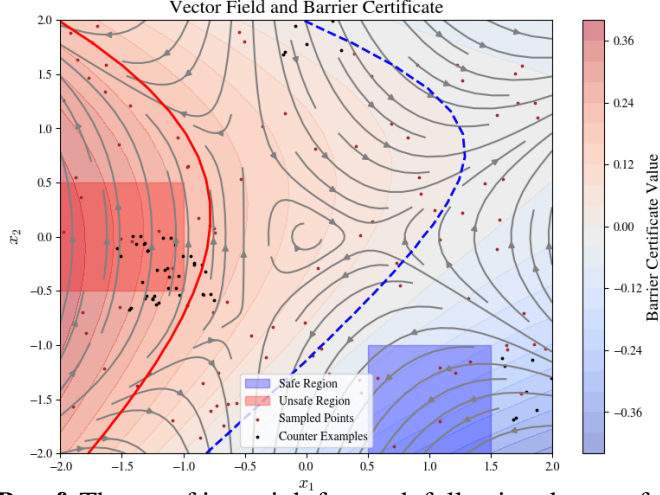


Figure 4: *Verified*  $k$ -NBC for the polynomial system, with the zero level set shown in dashed blue and the unsafe level set in red. Original sampled points are depicted in brown, while counterexamples are in black. The  $k$ -NBC was identified in the 7th iteration. The contour plot represents barrier certificate values across all states, and the true vector field is illustrated in gray.

**Proof** The proof is straightforward, following the proof of Proposition 6. ■

## Appendix C. Additional Case Studies

### C.1. Polynomial System

**System Description.** We first consider a discrete-time polynomial system as

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \Delta t \begin{bmatrix} x_2 + 2x_1x_2 \\ -x_1 + 2x_1^2 - 2x_2^2 \end{bmatrix},$$

where the sampling time  $\Delta t$  is 0.1 seconds. The model is assumed to be *unknown*, but an extensive dictionary of terms is available:  $\mathfrak{D}(x) = [x_1, x_2, x_1x_2, x_1^2, x_2^2]$ . This dictionary is constructed based on a first-principles understanding of the system, including all monomials up to degree 2. We consider a safety specification over the state space  $X = [-2, 2]^2$  with an initial region  $X_{\mathcal{I}} = [0.5, 1.5] \times [-2, -1]$  and unsafe region  $X_{\mathcal{U}} = [-2, -1] \times [-0.5, 0.5]$ . As our dictionary has 5 terms, we consider a single state trajectory of length 6, from initial state  $x_0 = [0.5, -2]$ . Using the datasets  $\mathcal{X}_0$ ,  $\mathcal{X}_1$ , and  $\mathcal{D}_0$ , we solve (6). We now obtain the data-driven model  $x^+ = \mathcal{X}_1 Q(x) \mathfrak{D}(x)$ .

**Initial Candidate.** We aim to find a  $k$ -NBC for this system, with  $k = 3$  and  $\epsilon = 0.1$ . For conservatism, the parameters are set to  $\eta_1 = 0.1$  and  $\eta_2 = 0.001$ . We sample 100 data points from the data-driven model to form our dataset triple  $(S, S^+, S^{k+})$ . We also consider a neural network with two inputs  $x_1, x_2$  (one per dimensions), 1 hidden layer containing 2 nodes, and one output which is the evaluation of our candidate  $k$ -NBC. Each node  $y$  evaluates the formula

$$y = (b + \sum_{i=1}^2 x_i w_i)^2, \quad (16)$$

where, for each node:  $b$  represents the bias,  $w_i$  are the weights, and the activation function is quadratic. For training, backpropagation is performed using the Adam optimizer (Kingma and Ba, 2014), running for 1000 epochs with a learning rate of 0.1. Verification with Z3 (De Moura and Bjørner, 2008) identified a counterexample at  $[0, \frac{15}{8}]$ .

**Retraining.** Prompted by this counterexample, it along with 20 points within a 0.1 radius are added to the dataset. The model is then retrained for another 1000 epochs with a reduced learning rate of 0.05. This loop was repeated for each counterexample until iteration 7 where the SMT solver Z3 returned `unsat` verifying the  $k$ -NBC  $\mathcal{B}(x)$  as valid for all  $x$ , as seen in Figure 4. In the figure, the true vector field is plotted but all results were generated and verified using the data-driven model. Allowing for rounding, the  $k$ -NBC is designed as

$$\mathcal{B}(x) = 0.02x_1^2 + 0.02x_1x_2 - 0.12x_1 - 0.04x_2^2 + 0.04x_2 + 0.10$$

which is not sum-of-squares, since its quadratic form is not positive semidefinite.

**Conventional Barrier Certificate Solution.** We investigate whether  $k$ -NBC satisfies the conventional barrier certificate conditions (where  $k = 1$  and  $\epsilon = 0$ ). However, Z3 identifies a counterexample at  $[0, \frac{255}{128}]$ .

## C.2. Pendulum System

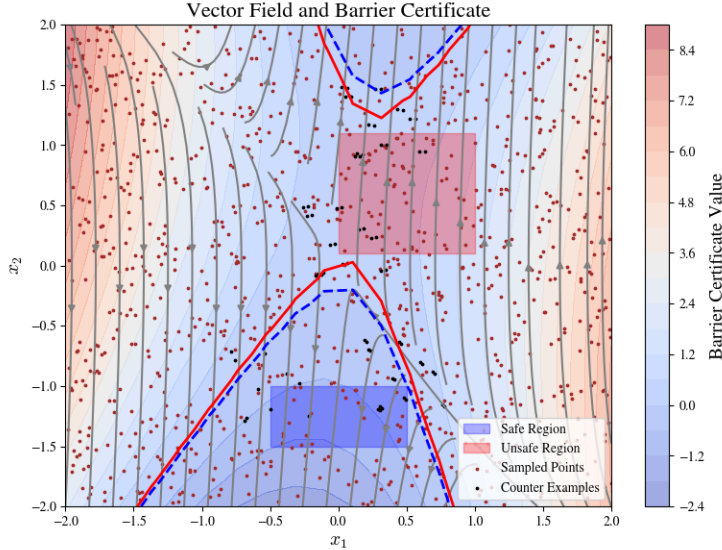


Figure 5: *Verified  $k$ -NBC for the pendulum system*, with the zero level set in dashed blue and the unsafe level set in red. Original sampled points are shown in brown, and counterexamples in black. The  $k$ -NBC was obtained after one retraining. The contour plot illustrates barrier certificate values across all states, while the true vector field is depicted in gray.

**System Description.** For our second case study, we consider a controlled pendulum system as

$$\begin{bmatrix} x_1^+ \\ x_2^+ \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \Delta t \begin{bmatrix} x_2 \\ \sin(x_1)g + \frac{b}{m}x_2 + \frac{-gmx_1 + bx_2}{ml} \end{bmatrix},$$

where  $x_1$  is the pendulum angle,  $x_2$  is the angular velocity,  $\Delta t = 0.1$  is the sampling time,  $g = 9.81$  is the gravitational acceleration,  $m = 1$  is the mass of the pendulum,  $l = 0.1$  is the pendulum's length, and  $b = 1.0$  is the damping coefficient.

Again, we consider the system to be *unknown*, but we have an extensive dictionary of terms  $\mathcal{D}(x) = [x_1, x_2, \sin(x_1), \cos(x_1)]$ . We consider a safety specification with regions  $X = [-2, 2]^2$ ,  $X_I = [-0.5, 0.5] \times [-1.5, -1]$  and  $X_U = [0, 1] \times [0.1, 1.1]$ . As the extensive dictionary has 4 terms, we take a single state trajectory of length 5 from initial state  $[0.5, -1.5]$ .

**Initial Candidate.** To find a  $k$ -NBC, we set  $k = 2$ ,  $\epsilon = 0.1$ ,  $\eta_1 = 0.1$ , and  $\eta_2 = 0.001$ . We sample 1000 data points to form our dataset triple  $(S, S^+, S^{k+})$ . We consider a neural network

with two inputs  $x_1, x_2$  (one per dimensions), 1 hidden layer containing 32 nodes with the same description as (16), and one output which is the evaluation of our candidate  $k$ -NBC. Using the Adam optimizer, we run 1000 epochs with learning rate 0.1 and verify candidates using the dReal (Gao et al., 2013) with  $\delta$ -bound set to 0.001. A counterexample is found on the first iteration in the range  $[0.4811, 0.4813] \times [-1.0002, -1.0000]$ .

**Retraining.** From the range of the counterexample, 10 additional points are added to the dataset for retraining of another 1000 epochs with learning rate 0.05. The process is repeated until iteration 8 where the SMT solver returns no counterexample, verifying the  $k$ -NBC valid for all  $x$ , as seen in Figure 5. The verified  $k$ -NBC  $\mathcal{B}(x)$  is omitted here due to its length.

**Conventional Barrier Certificate Solution.** We investigate whether  $k$ -NBC satisfies the conventional barrier certificate conditions (where  $k = 1$  and  $\epsilon = 0$ ). However, dReal identifies a counterexample in the range  $[0.282, 0.284] \times [-0.438, -0.436]$ .