mathematics

MDPI

Article

Verification of Approximate Initial-State Opacity for Control Systems via Neural Augmented Barrier Certificates

Shengpu Wang, Mi Ding, Wang Lin * and Yubo Jia *

School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China; wangshengpu0928@163.com (S.W.); houge1993@foxmail.com (M.D.)
* Correspondence: linwang@zstu.edu.cn (W.L.); jiayubo1964@163.com (Y.J.)

Abstract: In this paper, we propose an augmented barrier certificate-based method for formally verifying the approximate initial-state opacity property of discrete time control systems. The opacity verification problem is formulated as the safety verification of an augmented system and is then addressed by searching for augmented barrier certificates. A set of well-defined verification conditions is a prerequisite for successfully identifying augmented barrier certificates of a specific type. We first suggest a new type of augmented barrier certificate which produces a weaker sufficient condition for approximate initial-state opacity. Furthermore, we develop an algorithmic framework where a *learner* and a *verifier* interact to synthesize augmented barrier certificates in the form of neural networks. The *learner* trains neural certificates via the deep learning method, and the *verifier* solves several mixed integer linear programs to either ensure the validity of the candidate certificates or yield counterexamples, which are passed back to further guide the *learner*. The experimental results demonstrate that our approach is more scalable and effective than the existing sum of squares programming method.

Keywords: approximate initial-state opacity; barrier certificate; discrete-time control system; deep learning; mixed integer linear programming

MSC: 68N30

Citation: Wang, S.; Ding, M.; Lin, W.; Jia, Y. Verification of Approximate Initial-State Opacity for Control Systems via Neural Augmented Barrier Certificates. *Mathematics* **2022**, *10*, 2388. https://doi.org/10.3390/math10142388

Academic Editor: Armin Fügenschuh

Received: 29 May 2022
Accepted: 5 July 2022
Published: 7 July 2022

1. Introduction

1.1. Research Motivation

Cyber-physical systems (CPSs) consist of tightly coupled physical components and software systems and are deeply involved in safety-critical infrastructure, such as power plants, medical devices, and self-driving cars. In the last two decades, formal verification of the safety properties for cyber-physical systems has gained considerable attention [1,2]. However, security properties have not been investigated thoroughly for cyber-physical systems utill very recently [3,4]. Cyber-physical systems are security-critical, since the tight interaction between physical components and software systems may have the potential to release secret information and expose the system to malicious intruders. Therefore, ensuring the security of cyber-physical systems has become significantly important.

For cyber-physical systems, opacity verification is a vital requirement. Opacity is an information flow security property that captures whether or not the "secret" of the system can be revealed to an intruder that can infer the system's actual behavior based on the information flow [5,6]. Opacity has been widely investigated in the domain of discrete event systems [7–10]. For continuous state-space CPSs, the opacity verification can be performed by abstraction-based techniques in which continuous-space models are approximated by discrete ones [11,12].

The approximate initial-state opacity property requires the outside intruders to infer whether the initial state of the system is a secret one by observing the output trajectories of

the system [8]. A discretization-free barrier certificate-based approach for verifying the approximate initial-state opacity of discrete-time control systems was developed in [11]. The barrier certificate-based method is quite popular for the safety verification of CPSs [13–15], and in [11,12], an augmented barrier certificate is defined for an augmented system constructed by augmenting a control system with itself, while the initial and unsafe sets are designed to capture the secret and initial sets of the original system. Then, the existence of such barrier certificates can guarantee the approximate initial-state opacity of the original system. Moreover, a sum of squares (SOS) programming-based methodology is presented for the construction of the augmented barrier certificates. However, this methodology may suffer from scalability issues since it requires discretization of the state and input sets of the original system.

Motivated by the limitation of [11], in this work, we present a novel barrier certificate-based method for formally verifying the approximate initial-state opacity of discrete-time control systems. First, we suggest a new type of augmented barrier certificate which can produce a weaker sufficient condition for the approximate initial-state opacity. Intuitively, the augmented barrier certificates introduced in [11] guarantee that for any state run of the system starting from a secret state, all state runs starting from the non-secret state set have similar output trajectories. Therefore, the sufficient condition for the approximate initial-state opacity yielded from the augmented barrier certificates introduced in [11] is more conservative. Usually, a weaker condition for the augmented barrier certificates means that more augmented barrier certificates can be synthesized, as the expressiveness of the augmented barrier certificates is stronger.

We then suggest an algorithmic framework for synthesizing augmented barrier certificates in the form of neural networks. Our framework draws insight from the existing methods for learning the barrier certificates of hybrid systems [16–18] and consists of two modules. The *learner* trains a neural augmented barrier certificate that satisfies the barrier certificate conditions over a set of sampled data, and the *verifier* solves a mixed integer linear program (MILP) to either ensure the validity of the candidate augmented barrier certificate or yield counterexamples, which are passed back to further guide the *learner*. The loop is repeated until a trained candidate augmented barrier certificate is successfully verified. The main contributions of this paper are summarized as follows:

- We define a new type of augmented barrier certificate, which can produce a weaker sufficient condition for the approximate initial-state opacity.
- We present a framework for synthesizing neural augmented barrier certificates. In the framework, a counterexample guided procedure is adopted to speed up the construction of networks, and the verification of Rectified Linear Unit (ReLU) networks can be efficiently performed via an MILP.
- We carry out proof-of-concept case studies to empirically show the efficiency and practicability of the approach.

The structure of this paper is as follows. Section 1.2 introduces the related works. In Section 2, we give a less conservative sufficient condition with two relation functions to verify the approximate initial-state opacity, and then we train neural networks to synthesize the candidate neural barrier certificates and relation functions in Section 3.1 and convert the verification problem to a group of MILP optimization problems for validation in Section 3.2. In Section 3.3, we illustrate the barrier certificate and relation function synthesis algorithm. Section 4 provides an experimental evaluation of our algorithm with three examples, including linear, three-dimensional and non-polynomial examples. We conclude and introduce future work in Section 5.

1.2. Relate Work

Opacity is an important information flow property that was initially introduced in [5]. Specifically, the concept of language-based opacity was proposed in [19]. Aside from that, several state-based opacities were proposed, including initial-state opacity [20], K-step opacity [21], current-state opacity [22], and infinite-step opacity [23]. However,

the conditions of exact opacity could be too strong for the systems. Consequently, they need to be relaxed in some application scenarios. In [8], approximate opacity was introduced and verified by approximate opacity preserving the simulation relation. In [24,25], the authors synthesized a supervisor such that the closed-loop system was opaque.

The verification of opacity has also been widely studied in [26–28]. Four types of opacity properties have been verified: language-based opacity, initial-state opacity, current-state opacity, and initial-and-final-state opacity [7]. In [20], the verification problem of initial-state opacity to an initial-state estimation problem was translated, and this was addressed by the construction of initial-state estimators. Moreover, current-state and initial-state opacity problems in bounded Petri nets can be efficiently solved by adopting a compact representation of the reachability graph [29]. In [30], initial-state opacity for networks of interconnected control systems was verified. In [31], the verification of opacity for recursive tile systems is presented. Note that in [32], a finite abstraction-based technique was presented for verifying approximate opacity of the class of discrete-time stochastic systems, where the original control systems were approximated by discrete ones. For continuous state–space CPSs, a barrier certificate was used to verify the approximate initial-state opacity in [11,12]. However, this methodology may suffer from scalability issues since it requires discretization of the state and input sets of the original system.

Synthesizing augmented barrier certificates via deep learning methods is subject to further verification, which is related to formal verification of the neural networks, and much research focused on formal verification of the neural networks [33–35]. The problem of neural network verification is NP-hard [36], where a tool was implemented called Reluplex for validating neural networks with a ReLU activation function. In [37], a framework to verify deep neural networks based on Satisfiability Modulo Theory (SMT) was introduced. The verification of the properties of deep neural networks was finished based on the simplex method in [38]. The neural network with a ReLU activation function was verified by exploiting the dependencies between the ReLU nodes [39]. Furthermore, with the candidate network with ReLU, the non-linear SMT solver iSAT3 was used for verification [35]. In [17], candidate neural barrier certificates were synthesized with a specially structured network, which transformed the verification problems of neural networks into a group of mixed-integer linear programming problems and a mixed-integer quadratically constrained problem. Zhao et al. [40] provides an innovative method for synthesizing neural networks as barrier certificates, which can give safety guarantees for neural network-controlled systems.

2. An Augmented Barrier Certificate for Approximate Initial-State Opacity

We denote $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{Q}$, and $\mathbb{N}$ as real numbers, non-negative real numbers, rational numbers, and natural numbers, respectively, and appropriate subscripts are used to restrict the sets. The empty set is represented by $\emptyset$. Given a vector $\boldsymbol{x}$, we use $\mathbb{R}^n$ to represent an n-dimensional Euclidean space and $||\ \boldsymbol{x}\ ||$ to represent the Euclidean norm. For $a, b \in \mathbb{N}$, and $a \leq b$, we use $[a, b]$ to represent a closed interval. Given two sets A and B with $A \subset B$, we define the complement of A with respect to B as $B \backslash A = \{a \mid a \in B, a \notin A\}$. The Cartesian product of two sets A and B is defined as $A \times B = \{(a, b) \mid a \in A, b \in B\}$.

2.1. Approximate Initial-State Opacity for Discrete-Time Control Systems

In this subsection, we will introduce some preliminaries:

Definition 1 ([11]). *A discrete-time control system (dt-cs) is defined by a tuple* $\Pi = (\mathbb{X}, \mathbb{X}_0, \mathbb{X}_s, \mathbb{X}_{ns}, \mathbb{U}, f, \mathbb{Y}, h)$ *where the following are true:*

- $\mathbb{X}$, $\mathbb{U}$, *and* $\mathbb{Y}$ *are the state set, control input set, and output set, respectively;*
- $\mathbb{X}_0$, $\mathbb{X}_s$, *and* $\mathbb{X}_{ns}$ *are the initial state set, secret state set, and non-secret state set, respectively, where* $\mathbb{X}_0, \mathbb{X}_s, \mathbb{X}_{ns} \subseteq \mathbb{X}$, *and* $\mathbb{X}_s \bigcap \mathbb{X}_{ns} = \emptyset$;
- $f : \mathbb{X} \times \mathbb{U} \to \mathbb{X}$ *is the state transition function;*
- $h : \mathbb{X} \to \mathbb{Y}$ *is the output function.*

Intuitively, the discrete-time control system Π can be modeled by the following difference equations:

$$\Pi : \begin{cases} \boldsymbol{x}(t+1) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(t) = h(\boldsymbol{x}(t)) \end{cases} \tag{1}$$

where $\boldsymbol{x}$: $\mathbb{N} \to \mathbb{X}$, $\boldsymbol{u}$: $\mathbb{N} \to \mathbb{U}$, and $\boldsymbol{y}$: $\mathbb{N} \to \mathbb{Y}$ are the state signal, control input signal, and output signal, respectively. We use $\boldsymbol{x}_{x_0,\boldsymbol{u}}(k)$ to denote the state reached at time k of Π from the initial state $\boldsymbol{x}_0 \in \mathbb{X}_0$ under the input signal $\boldsymbol{u}$ and use $\boldsymbol{y}_{x_0,\boldsymbol{u}}(k) = h(\boldsymbol{x}_{x_0,\boldsymbol{u}}(k))$ to denote the output corresponding to the state $\boldsymbol{x}_{x_0,\boldsymbol{u}}(k)$.

Given a discrete-time control system Π, we want to verify the approximate initial-state opacity; that is, for each finite state run whose initial state is a secret state, there exists another finite state run starting from a non-secret state with a similar output trajectory. The following definition describes the approximate initial-state opacity of the system Π:

Definition 2 ([8]). *Consider a discrete-time control system $\Pi = (\mathbb{X}, \mathbb{X}_0, \mathbb{X}_s, \mathbb{X}_{ns}, \mathbb{U}, f, \mathbb{Y}, h)$ and a constant $\delta \in \mathbb{R}_{\geq 0}$. The system Π is δ-approximate initial-state opacity if for any $\boldsymbol{x}_0 \in \mathbb{X}_0 \bigcap \mathbb{X}_s$ and any finite state run $\boldsymbol{x}_{x_0,\boldsymbol{u}} = \{\boldsymbol{x}_0, \boldsymbol{x}_1 \ldots, \boldsymbol{x}_n\}$, there exists $\hat{\boldsymbol{x}}_0 \in \mathbb{X}_0 \bigcap \mathbb{X}_{ns}$ and a finite state run $\boldsymbol{x}_{\hat{x}_0,\hat{u}} = \{\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1 \ldots, \hat{\boldsymbol{x}}_n\}$ such that*

$$\max_{k \in [0;n]} \|(h(\boldsymbol{x}_k) - h(\hat{\boldsymbol{x}}_k)\| \leq \delta, \tag{2}$$

where the threshold parameter δ can be interpreted as either the measurement imprecision of the intruder or the security level the system can guarantee.

It is assumed that the secret of the system is not revealed initially; otherwise, it will violate δ-approximate initial-state opacity initially. Therefore, we assume that $\forall \boldsymbol{x}_0 \in \mathbb{X}_0 \bigcap \mathbb{X}_s$, where

$$\{\boldsymbol{x} \in \mathbb{X}_0 \mid \|h(\boldsymbol{x}) - h(\boldsymbol{x}_0)\| \leq \delta\} \subseteq \mathbb{X}_{ns}. \tag{3}$$

To verify the approximate initial-state opacity of discrete-time control systems, the authors of [11,12] achieved the goal with the help of barrier certificates. In detail, for a discrete-time control system Π, the approximate initial-state opacity verification problem can be cast into checking a safety property of the associated augmented system $\Pi \times \Pi$ using barrier certificates. More concretely, there is the augmented system

$$\begin{aligned} \Pi \times \Pi = (&\mathbb{X} \times \mathbb{X}, \mathbb{X}_0 \times \mathbb{X}_0, \mathbb{X}_s \times \mathbb{X}_s, \\ &\mathbb{X}_{ns} \times \mathbb{X}_{ns}, \mathbb{U} \times \mathbb{U}, f \times f, \mathbb{Y} \times \mathbb{Y}, h \times h), \end{aligned}$$

which can be regarded as the product of a dt-cs Π and itself. We denote a pair of states in $\Pi \times \Pi$ by $(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in \mathbb{X} \times \mathbb{X}$ and use $\Theta = \mathbb{X} \times \mathbb{X}$ to represent the state space of the augmented system. Given the initial state $(\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0)$ of the augmented system $\Pi \times \Pi$ and input run $(\boldsymbol{v}, \hat{\boldsymbol{v}})$, we denote the state trajectory of $\Pi \times \Pi$ with $(\boldsymbol{x}_{x_0,\boldsymbol{v}}, \hat{\boldsymbol{x}}_{x_0,\hat{\boldsymbol{v}}})$.

In order to leverage barrier certificates to verify the approximate initial-state opacity for a given discrete-time control system Π, the initial state set Θ_0 and the unsafe state set Θ_u are defined as follows:

$$\begin{aligned} \Theta_0 &= \{(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in (\mathbb{X}_0 \cap \mathbb{X}_s) \times (\mathbb{X}_0 \backslash \mathbb{X}_s) \mid \|h(\boldsymbol{x}) - h(\hat{\boldsymbol{x}})\| \leq \delta\}, \\ \Theta_u &= \{(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in \mathbb{X} \times \mathbb{X} \mid \|h(\boldsymbol{x}) - h(\hat{\boldsymbol{x}})\| > \delta\}. \end{aligned} \tag{4}$$

The following theorem introduces a sufficient condition for verifying the approximate initial-state opacity of discrete-time control systems via a concept of augmented barrier certificates:

Theorem 1. *Given a dt-cs Π, the associated augmented system $\Pi \times \Pi$ and the sets Θ_0, Θ_u in Equation (4), if there exists a function $B : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ such that*

$$\forall (\boldsymbol{x}, \hat{\boldsymbol{x}}) \in \Theta_0 \qquad\qquad B(\boldsymbol{x}, \hat{\boldsymbol{x}}) \leq \epsilon_1, \tag{5}$$

$$\forall(\boldsymbol{x},\hat{\boldsymbol{x}}) \in \Theta_u \qquad B(\boldsymbol{x},\hat{\boldsymbol{x}}) \geq \epsilon_2, \tag{6}$$

$$\begin{aligned} &\forall(\boldsymbol{x},\hat{\boldsymbol{x}}) \in \Theta, \forall \boldsymbol{u} \in \mathbb{U}, \exists \hat{\boldsymbol{u}} \in \mathbb{U} \\ &\qquad B(f(\boldsymbol{x},\boldsymbol{u}), f(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}})) - B(\boldsymbol{x},\hat{\boldsymbol{x}}) \leq 0, \end{aligned} \tag{7}$$

where the constants ϵ_1, $\epsilon_2 \in \mathbb{R}_{\geq 0}$, *and* $\epsilon_2 > \epsilon_1$ *are present, then the* δ*-approximate initial-state opacity of the dt-cs* Π *is guaranteed.*

The function $B(\boldsymbol{x},\hat{\boldsymbol{x}})$ in Theorem 1 is called an augmented barrier certificate of the augmented system $\Pi \times \Pi$. Intuitively, the augmented barrier certificate $B(\boldsymbol{x},\hat{\boldsymbol{x}})$ ensures that all trajectories of the system $\Pi \times \Pi$ originating from the initial state sets Θ_0 never reach the unsafe region Θ_u, and then the safety of the dt-cs Π is guaranteed.

In this paper, we focus on the nonlinear discrete-time control systems, where the state transition function and output function are represented by multivariate elementary functions, and the sets $\mathbb{X}$, $\mathbb{U}$, $\mathbb{Y}$, $\mathbb{X}_0$, $\mathbb{X}_s$, and $\mathbb{X}_{ns}$ are bounded polyhedrons each defined by linear inequalities over a variable $\boldsymbol{x}$. Concretely, multivariate elementary functions are expressed by the following grammar:

$$\begin{aligned} f, g ::= &cx | f | f^a | \ln(f) | \sin(f) | \cos(f) \\ &| e^f | f + g | f - g | f \times g | f / g, \end{aligned}$$

where $c \in \mathbb{R}$ is a real constant, $a \in \mathbb{Q}$ is a rational constant, and $\boldsymbol{x}$ can be any system variable.

2.2. A New Augmented Barrier Certificate for Approximate Initial-State Opacity

The augmented barrier certificates introduced in Theorem 1 guarantee that all trajectories of the augmented system $\Pi \times \Pi$ starting from the initial state set Θ_0 never reach the unsafe states in Θ_u. This means that for any state run of the discrete-time control system Π starting from a secret state $\boldsymbol{x}_0 \in \mathbb{X}_0 \cap \mathbb{X}_s$, all state runs starting from the non-secret state set $\mathbb{X}_0 \backslash \mathbb{X}_s$ have similar output trajectories. However, according to Definition 2, the δ-approximate initial-state opacity just requires that for each finite state run whose initial state is a secret state, there exists another finite state run starting from a non-secret state with a similar output trajectory. Clearly, the sufficient condition of the augmented barrier certificates introduced in Theorem 1 is more conservative, which may make the problem of δ-approximate initial-state opacity verification infeasible.

In the following, we introduce a less conservative augmented barrier certificate for the verification of approximate initial-state opacity:

Theorem 2. *Consider a dt-cs* Π *and the associated augmented system* $\Pi \times \Pi$. *If there exists a function* $B : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ *that satisfies the conditions in Equations (6) and (7) in Theorem 1 and moreover*

$$\begin{aligned} &\forall \boldsymbol{x} \in \mathbb{X}_0 \cap \mathbb{X}_s, \exists \hat{\boldsymbol{x}} \in \mathbb{X}_0 \backslash \mathbb{X}_s \\ &\qquad \|h(\boldsymbol{x}) - h(\hat{\boldsymbol{x}})\| \leq \delta \wedge B(\boldsymbol{x},\hat{\boldsymbol{x}}) \leq \epsilon_1. \end{aligned} \tag{8}$$

then the δ*-approximate initial-state opacity of the dt-cs* Π *is guaranteed.*

Proof. Consider an arbitrary initial secret state $\boldsymbol{x}_0$ and an input sequence $\boldsymbol{u}$, as well as the corresponding state sequence $\boldsymbol{x}_{\boldsymbol{x}_0,\boldsymbol{u}}$ in Π. First note Equations (3) and (8), where $\{\boldsymbol{x} \in \mathbb{X}_0 \mid \|h(\boldsymbol{x}) - h(\boldsymbol{x}_0)\| \leq \delta\} \subseteq \mathbb{X}_{ns}$. It follows that there exists an initial state $\hat{\boldsymbol{x}}_0 \in \mathbb{X}_0 \backslash \mathbb{X}_s$ for every initial state $\boldsymbol{x}_0 \in \mathbb{X}_0 \cap \mathbb{X}_s$ such that $\|h(\hat{\boldsymbol{x}}_0) - h(\boldsymbol{x}_0)\| \leq \delta$, based on Equation (8). Therefore, the system is initially satisfied for δ-approximate initial-state opacity. The existence of an augmented barrier certificate $B(\boldsymbol{x},\hat{\boldsymbol{x}})$ as described in Theorem 1 guarantees that for any $\boldsymbol{x}_0 \in \mathbb{X}_0 \cap \mathbb{X}_s$, there exists $\hat{\boldsymbol{x}}_0 \in \mathbb{X}_0 \backslash \mathbb{X}_s$ and a control policy $\hat{\boldsymbol{u}}$ such that any state sequence of

$\Pi \times \Pi$ starting from $(\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0)$ never reaches the unsafe region Θ_u under input run $(\boldsymbol{u}, \hat{\boldsymbol{u}})$. In other words, the following is true:

$$\max_{t\in[0;n]} \|(h(\boldsymbol{x}_{\boldsymbol{x}_0,\boldsymbol{u}}(t)) - h(\hat{\boldsymbol{x}}_{\hat{\boldsymbol{x}}_0,\hat{\boldsymbol{u}}}(t))\| \leq \delta \tag{9}$$

Since $\boldsymbol{x}_0$ and $\boldsymbol{x}_{\boldsymbol{x}_0,\boldsymbol{u}}$ are arbitrarily chosen, we conclude that Π is the δ-approximate initial-state opacity. □

Since the condition in Equation (8) is weaker than the initial condition in Theorem 1, Theorem 2 produces a less conservative sufficient condition for the δ-approximate initial-state opacity of the dt-cs Π. Usually, a weaker condition on augmented barrier certificates means that more augmented barrier certificates can be synthesized, as the associated expressiveness is stronger. Therefore, the problem of verifying the δ-approximate initial-state opacity for the dt-cs Π can be transformed into the problem of computing the relaxed augmented barrier certificates defined in Theorem 2.

However, the forall-exists quantifiers make the problem of computing the augmented barrier certificates given in Theorem 2 computationally difficult. From a computational point of view, the forall-exists statement in Equation (8) can be encoded as

$$\forall(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in \widetilde{\Theta}_0 \qquad B(\boldsymbol{x}, \hat{\boldsymbol{x}}) \leq \epsilon_1 \tag{10}$$

where $\widetilde{\Theta}_0$ is defined as follows:

$$\begin{aligned} \widetilde{\Theta}_0 = \{(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in (\mathbb{X}_0 \cap \mathbb{X}_s) \times (\mathbb{X}_0 \backslash \mathbb{X}_s) \mid \\ \hat{\boldsymbol{x}} = p(\boldsymbol{x}) \wedge \|h(\boldsymbol{x}) - h(\hat{\boldsymbol{x}})\| \leq \delta\}, \end{aligned} \tag{11}$$

by introducing a function p over $\boldsymbol{x}$ to describe the relation between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$. Similarly, we introduce another relation function q over $\boldsymbol{x}$, $\hat{\boldsymbol{x}}$, and $\boldsymbol{u}$ to encode the forall-exists statement:

Theorem 3. *Consider a dt-cs* Π *and the associated augmented system* $\Pi \times \Pi$. *If there exist functions* $B : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$, $p : \mathbb{X} \to \mathbb{R}$ *and* $q : \mathbb{X} \times \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ *which satisfy the conditions in Equations (6) and (10) and moreover*

$$\begin{aligned} &\forall(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in \Theta, \forall(\boldsymbol{u}, \hat{\boldsymbol{u}}) \in \mathbb{U} \times \mathbb{U}, \\ &\quad \hat{\boldsymbol{u}} = q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u}) \in \mathbb{U} \wedge B(f(\boldsymbol{x}, \boldsymbol{u}), f(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})) - B(\boldsymbol{x}, \hat{\boldsymbol{x}}) \leq 0. \end{aligned} \tag{12}$$

then the δ*-approximate initial-state opacity of the dt-cs* Π *is guaranteed.*

Now, the problem of verifying the δ-approximate initial-state opacity for the dt-cs Π can be transformed into the problem of computing the relaxed augmented barrier certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$ and the relation functions $p(\boldsymbol{x})$, $q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u})$ defined in Theorem 3.

3. Synthesis of an Augmented Barrier Certificate via Learning and Verification

In this section, we introduce an iterative framework to synthesize neural barrier certificates and relation functions to complete the verification of the approximate initial-state opacity of the dt-cs Π. As shown in Figure 1, the procedure is structured as an inductive loop between a *learner* and a *verifier*. At first, we present a less conservative augmented barrier certificate for the validation of approximate initial-state opacity. Then, the *learner* trains the certificate functions simultaneously over sampled data points from the initial set, the unsafe set, and the state space. After that, the trained candidates are formally verified by the *verifier* according to the conditions in Theorem 3. Due to the structure of ReLU neural networks, the verification problem can be transformed into a set of efficiently solvable MILP problems.

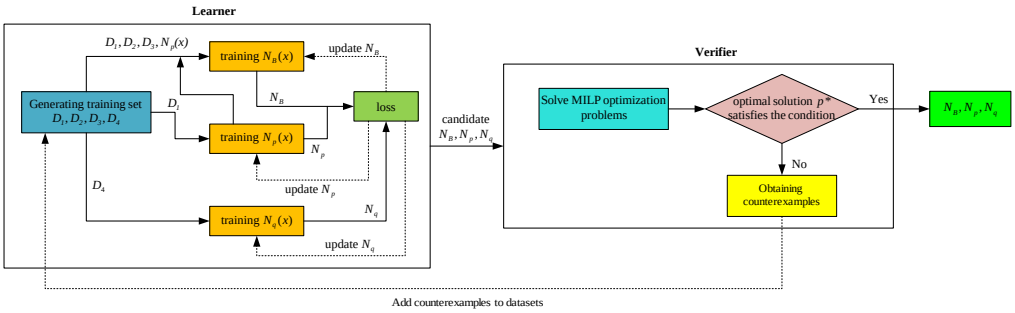

Figure 1. The framework for verifying approximate initial-state opacity.

3.1. Neural Certificate Training

In this subsection, we give the details for training the neural barrier certificates.

3.1.1. The Structure of Neural Networks

Consider a dt-cs Π as defined in Definition 1, where state $\boldsymbol{x}$ and control input $\boldsymbol{u}$ are of n and m dimensions, respectively. Now, we provide the specific structure of neural networks for certificate functions $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$, $p(\boldsymbol{x})$ and $q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u})$.

- One input layer, one output layer, and several hidden layers;
- The number of input neurons for B, p, and q is $2n$, n, and $2n + m$, respectively;
- Both B and q have one output neuron, while p has n output neurons;
- The ReLU function is the only legal activation function.

For the above feedforward neural network, a non-input layer neuron value is computed by the preceding layer neuron values, the layer weight matrix, and the bias vector. Let $\boldsymbol{x}_0$ denote the given neuron value of the input layer, $\boldsymbol{z}_k$ and $\boldsymbol{x}_k$ $(1 \leq k \leq \eta - 1)$ denote the neuron value of the hidden layer L_k before and after ReLU activation function, respectively, and $\boldsymbol{x}_\eta$ denote the neuron value of the output layer. Then, the forward propagation of the neural network is as follows:

$$\begin{cases} \boldsymbol{z}_k = W_k \boldsymbol{x}_{k-1} + \boldsymbol{b}_k, & k = 1, \ldots, \eta - 1 \\ \boldsymbol{x}_k = ReLU(\boldsymbol{z}_k), & k = 1, \ldots, \eta - 1 \\ \boldsymbol{x}_\eta = W_\eta \boldsymbol{x}_{\eta-1} + \boldsymbol{b}_\eta \end{cases} \tag{13}$$

where $\eta - 1$ is the number of hidden layers of the neural network. Intuitively, the output layer of neural network B and q has only one neuron, and hence their outputs are real numbers. As for the output of p, it is a vector and in the same form of the corresponding input vector.

According to Universal Approximation Theorem, we can approximate any complex rational function by the above neural network with the ReLU activation function, which makes it possible for us to synthesize the barrier certificate and relation functions. By restricting the number of hidden layers and the type of activation function, the verification problems of Equations (6), (10), and (12) can be transformed into a set of MILP problems. Then, the verification can be accomplished with the help of the optimization tool Gurobi.

3.1.2. Training Dataset Generation

The procedure for synthesizing neural networks takes a data-driven approach. Here, we discuss how to generate adequate datasets for training the candidate network. Datasets are generated while aiming at making the network fulfill the barrier certificate conditions given by Theorem 3. We sample points from the domain Θ, initial secret region $\mathbb{X}_0 \cap \mathbb{X}_s$, and unsafe region Θ_u of the considered system $\Pi \times \Pi$ for obtaining datasets D_1, D_2, and D_3, respectively.

The method employed for sampling is mesh generation. At first, we mesh the super rectangles in the sampling regions bounded with a fixed mesh size. Then, we filter out the

points that do not satisfy the constraints of Equations (6), (10) and (12). As for the dataset D_4 of neural networks q, it is essentially a vector splicing of D_3 with the control input $\boldsymbol{u}$. Finally, we obtain four finite data sets, D_1, D_2, D_3, and D_4.

3.1.3. Loss Function Encoding

Given datasets D_1, D_2, D_3, and D_4, according to Equations (6), (10) and (12) given by Theorem 3, the neural network B is trained to satisfy the following conditions:

$$B(\boldsymbol{x}, p(\boldsymbol{x})) \leq 0 \qquad \forall \boldsymbol{x} \in D_1 \tag{14}$$

$$B(\boldsymbol{x}, \hat{\boldsymbol{x}}) > 0 \qquad \forall (\boldsymbol{x}, \hat{\boldsymbol{x}}) \in D_2 \tag{15}$$

$$B(f(\boldsymbol{x}, \boldsymbol{u}), f(\hat{\boldsymbol{x}}, q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u}))) - B(\boldsymbol{x}, \hat{\boldsymbol{x}}) \leq 0 \qquad \forall (\boldsymbol{x}, \hat{\boldsymbol{x}}) \in D_3, \forall \boldsymbol{u} \in \mathbb{U} \tag{16}$$

According to Equations (14)–(16) and the output bound of the control policy given by the control input $\boldsymbol{u}$ and p, we construct sub-loss functions that lead the optimizers to build the network by minimizing the loss.

For satisfying Equation (14), with the input from dataset D_1, the output of B should satisfy $B(\boldsymbol{x}, p(\boldsymbol{x})) \leq 0$. Otherwise, there would be a non-zero loss. Let c_1 be a small non-negative tolerance. The first sub-loss function is defined as follows:

$$L_1 = \sum_{\boldsymbol{x} \in D_1} ReLU(B(\boldsymbol{x}, p(\boldsymbol{x})) + c_1) \tag{17}$$

Similarly, for Equation (15), the output of the neural network is supposed to conform with $B(\boldsymbol{x}, \hat{\boldsymbol{x}}) > 0$. A non-zero loss is also incurred if the condition is violated. Given a small positive constant c_2, the second sub-loss function is defined as follows:

$$L_2 = \sum_{(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in D_2} ReLU(-B(\boldsymbol{x}, \hat{\boldsymbol{x}}) + c_2) \tag{18}$$

To meet Equation (16), with a small positive number c_3, the third sub-loss function is defined as follows:

$$L_3 = \sum_{(\boldsymbol{x}, \hat{\boldsymbol{x}}) \in D_3, \boldsymbol{u} \in \mathbb{U}} ReLU(B(f(\boldsymbol{x}, \boldsymbol{u}), f(\hat{\boldsymbol{x}}, q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u}))) - B(\boldsymbol{x}, \hat{\boldsymbol{x}}) + c_3) \tag{19}$$

Aside from this, the output of the control policy q has upper and lower bounds, since the value of the control input is within the given closed interval. Let l_q, u_q be the upper and lower bounds of the control input, respectively. The output of q is between l_q and u_q (i.e., $q(\boldsymbol{x}) \in [l_q, u_q]$). After that, the fourth sub-loss function which leads the training of q is defined as follows:

$$\begin{aligned} L_4 = \sum_{(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u}) \in D_4} \Big(& ReLU(q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u}) - u_q) \\ & + ReLU(-q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u}) + l_q) \Big) \end{aligned} \tag{20}$$

According to Equation (11), we know that $p(\boldsymbol{x}) \in \mathbb{X}_0 \backslash \mathbb{X}_s$ when input $\boldsymbol{x} \in \mathbb{X}_0 \cap \mathbb{X}_s$. Furthermore, each unit of the output of p will be in $[l_p, u_p]$. Then we obtain the sub-loss function L_5 as:

$$\begin{aligned} L_5 = \sum_{\boldsymbol{x} \in D_1} \Big(& ReLU(\|h(p(\boldsymbol{x})) - h(\boldsymbol{x})\| - \delta + c_4) \\ & + ReLU(p(\boldsymbol{x}) - u_p) + ReLU(-p(\boldsymbol{x}) + l_p) \Big) \end{aligned} \tag{21}$$

where c_4 is a small positive constant. The role of the four tolerances c_1, c_2, c_3, and c_4 is to make the non-sampled points in the neighborhood of the sampled data obtain a loss of zero. Finally, let $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5 > 0$ denote the weights of the sub-losses. The total loss function for training B, p, q can be expressed as

$$L = \alpha_1 L_1 + \alpha_2 L_2 + \alpha_3 L_3 + \alpha_4 L_4 + \alpha_5 L_5 \tag{22}$$

Significantly, the sub-loss L_3 is related to the training of networks B and q. Hence, we can make α_3 slightly larger to promote effective training. We minimize the total loss L for training three neural networks and adopt an Adam optimizer, which achieves quick training. When L decreases to zero, we obtain candidate networks B, p, and q at the same time.

3.2. Certificate Function Verification

Since the candidate neural certificate functions B, p, and q are trained over the sampled datasets, the empirical loss L converges to zero, which only guarantees that the candidate certificate functions satisfy the barrier certificate conditions in Theorem 3 over such sampled datasets. Thus, the candidate neural certificate functions require further verification. In this section, we present an approach to formally check whether the candidate neural certificate functions B, p, and q precisely satisfy the following barrier certificate conditions in Theorem 3:

$$\forall (x, \hat{x}) \in \widetilde{\Theta}_0 \qquad B(x, \hat{x}) \leq 0 \tag{23}$$

$$\forall (x, \hat{x}) \in \Theta_u \qquad B(x, \hat{x}) > 0, \tag{24}$$

$$\begin{aligned} &\forall (x, \hat{x}) \in \Theta, \forall (u, \hat{u}) \in \mathbb{U} \times \mathbb{U}, \\ &\quad \hat{u} = q(x, \hat{x}, u) \in \mathbb{U} \wedge B(f(x, u), f(\hat{x}, \hat{u})) - B(x, \hat{x}) \leq 0. \end{aligned} \tag{25}$$

By utilizing the special structure of ReLU networks, the problem of verifying the candidate neural certificate functions B, p, and q can be encoded as several optimization problems, whose globally optimal objective values can be computed efficiently via the optimizer Gurobi. In the following, we will verify the neural certificate functions via MILP-based encoding.

Verifying Condition (Equation (23)). According to Equation (23), the neural augmented barrier certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$ should be nonpositive over the initial set $\widetilde{\Theta}_0$. To ease the presentation, we use $\boldsymbol{y}_0$ and y_η to denote the input and output of the neural certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$, respectively, where $\boldsymbol{y}_0 = (\boldsymbol{x}_0, \hat{\boldsymbol{x}}_0)$ and $y_\eta = B(\boldsymbol{y}_0)$. Based on the forward propagation of the neural networks in Equation (13), the problem of verifying the condition in Equation (23) can be transformed into the following optimization problem:

$$\begin{aligned} &\rho^*_{Init} = \max y_\eta \\ &s.t. \begin{cases} \boldsymbol{x}_0 \in \mathbb{X}_0 \cap \mathbb{X}_s, \\ p(\boldsymbol{x}_0) \in \mathbb{X}_0 \backslash \mathbb{X}_s, \\ \boldsymbol{z}_k = W_k \boldsymbol{y}_{k-1} + \boldsymbol{b}_k, & k = 1, \ldots, \eta - 1, \\ \boldsymbol{y}_k = ReLU(\boldsymbol{z}_k), & k = 1, \ldots, \eta - 1, \\ y_\eta = W_\eta \boldsymbol{y}_{\eta-1} + \boldsymbol{b}_\eta, \end{cases} \end{aligned} \tag{26}$$

where η is the number of hidden layers of the neural certificate B and ρ^*_{Init} is the globally optimal solution. Clearly, if the globally optimal solution $\rho^*_{Init} > 0$, then the condition in Equation (23) is verified to be satisfied.

The application of ReLU activation functions improves the approximation capability of the neural networks, but additionally, the nonlinearity of the ReLU function makes the problem in Equation (26) difficult to solve directly. Fortunately, an exact MILP-based encoding of ReLU functions is introduced to treat ReLU [33]. More concretely, the ReLU

function, which is defined over $\boldsymbol{z} \in [l_z, u_z]$, can be encoded by the following linear and binary constraints:

$$\begin{cases} -\boldsymbol{z} + \boldsymbol{y} + l - l \cdot t \leq 0, \\ \boldsymbol{z} - \boldsymbol{y} \leq 0, \\ \boldsymbol{y} - u \cdot t \leq 0, \\ 0 \leq \boldsymbol{y}, \\ t \in \{0, 1\}, \end{cases} \tag{27}$$

where t is a binary variable and l and u are constants satisfying $l \leq l_z$ and $u \geq u_z$, respectively. Now, by utilizing the above MILP-based encoding, the optimization problem in Equation (26) can be rewritten as the following MILP problem:

$$\rho^*_{Init} = \max y_\eta$$
$$s.t. \begin{cases} \boldsymbol{x}_0 \in \mathbb{X}_0 \cap \mathbb{X}_s, \\ p(\boldsymbol{x}_0) \in \mathbb{X}_0 \backslash \mathbb{X}_s, \\ \boldsymbol{y}_0 = (\boldsymbol{x}_0, p(\boldsymbol{x}_0)), \\ \boldsymbol{z}_k = W_k \boldsymbol{y}_{k-1} + \boldsymbol{b}_k, \\ -\boldsymbol{z}_k + \boldsymbol{y}_k \leq l_k(t_k - 1), \\ \boldsymbol{z}_k - \boldsymbol{y}_k \leq 0, \\ \boldsymbol{y}_k - u_k \cdot t_k \leq 0, \\ t_k \in \{0, 1\}, \\ \boldsymbol{y}_k \geq 0, \\ k = 1, \ldots, \eta - 1, \\ y_\eta = W_\eta \boldsymbol{y}_{\eta-1} + \boldsymbol{b}_\eta. \end{cases} \tag{28}$$

For simplicity, we set $l_k = -10^9$ and $u_k = 10^9$. This MILP problem (Equation (28)) can be solved via the optimizer Gurobi. Indeed, Gurobi returns the optimum $\tilde{\rho}_{Init}$ and the corresponding optimal solution $\tilde{x}_{Init}$, together with the maximum relative error ξ_{Init} between $\tilde{\rho}_{Init}$ and the globally optimum ρ^*_{Init}, which is formally expressed as

$$\xi_{Init} \geq \frac{|\tilde{\rho}_{Init} - \rho^*_{Init}|}{|\tilde{\rho}_{Init}|}.$$

As stated in [17], if $\xi_{Init} < 1$, then $\tilde{\rho}_{Init}$ and ρ^*_{Init} have the same sign. Therefore, if $\tilde{\rho}_{Init} \leq 0$ while the maximum relative error $\xi_{Init} < 1$, then $\rho^*_{Init} \leq 0$, which means that the condition in Equation (23) is verified to be satisfied. Otherwise, we randomly sample a set of points in a neighborhood of the optimal solution $\tilde{\boldsymbol{x}}_{Init}$ and add those data points that violate the condition in Equation (23) to the training dataset D_1 in the next iteration to refine the networks.

Verifying Condition (Equation (24)). According to Equation (24), the neural augmented barrier certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$ should be positive over the unsafe set Θ_u. The verification of the condition in Equation (24) takes the same form. Based on the forward propagation of neural networks in Equation (13) and the MILP-based encoding of ReLU functions, the problem of this verifying condition (Equation (24)) can be transformed into the following MILP problem:

$$\rho^*_{Unsafe} = \min x_\eta$$
$$s.t. \begin{cases} \boldsymbol{x}_0 \in \Theta_u, \\ z_k = W_k \boldsymbol{x}_{k-1} + \boldsymbol{b}_k, \\ -z_k + \boldsymbol{x}_k \leq l_k(t_k - 1), \\ z_k - \boldsymbol{x}_k \leq 0, \\ \boldsymbol{x}_k - u_k \cdot t_k \leq 0, \\ t_k \in \{0, 1\}, \\ \boldsymbol{x}_k \geq 0, \\ k = 1, \ldots, \eta - 1, \\ x_\eta = W_\eta \boldsymbol{x}_{\eta-1} + \boldsymbol{b}_\eta. \end{cases} \tag{29}$$

By solving the above MILP problem via the optimizer Gurobi, we can obtain the optimum $\tilde{\rho}_{Unsafe}$ and the corresponding optimal solution $\tilde{\boldsymbol{x}}_{Unsafe}$ together with the maximum relative error ξ_{Unsafe}. If $\tilde{\rho}_{Unsafe} \geq 0$ while the maximum relative error $\xi_{Unsafe} < 1$, then the globally optimum $\rho^*_{Unsafe} \geq 0$, and the condition in Equation (24) is verified to be satisfied. Otherwise, we randomly sample a set of points in a neighborhood of the optimal solution $\tilde{x}_{Unsafe}$ and add those data points that violate the condition in Equation (24) to the training dataset D_2 in the next iteration to refine the networks.

Verifying condition (Equation (25)). For the condition in Equation (25), the difference between $B(f(\boldsymbol{x},\boldsymbol{u}), f(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}))$, and $B(\boldsymbol{x},\hat{\boldsymbol{x}})$ should be non-positive over the state set Θ and input set $\mathbb{U}\times\mathbb{U}$. Here, $\hat{\boldsymbol{u}} = q(\boldsymbol{x},\hat{\boldsymbol{x}},\boldsymbol{u})$. Note that the condition in Equation (25) involves the valve of $\boldsymbol{y}_0$ after the transition function; in other words, $\boldsymbol{y}'_0 = (\boldsymbol{x}'_0, \hat{\boldsymbol{x}}'_0) = (f(\boldsymbol{x}_0,\boldsymbol{u}), f(\hat{\boldsymbol{x}}_0,\hat{\boldsymbol{u}}))$. Therefore, the problem of $B(f(x)) - B(x) < 0$ can be transformed into finding whether the maximum value of $B(\boldsymbol{y}'_0) - B(y_0)$ is negative.

Based on the forward propagation of neural networks in Equation (14) and the MILP-based encoding of ReLU functions, the problem of this verifying condition (Equation (25)) can be transformed into the following optimization problem:

$$\rho^*_{Evol} = \max y'_\eta - y_\eta$$
$$s.t. \begin{cases} \boldsymbol{y}_0 \in \Theta, \\ \boldsymbol{u}, \hat{\boldsymbol{u}} \in \mathbb{U}, \\ \boldsymbol{y}'_0 = (\boldsymbol{x}'_0, \hat{\boldsymbol{x}}'_0), \\ z_k = W_k \boldsymbol{y}_{k-1} + b_k, & z'_k = W_k \boldsymbol{y}'_{k-1} + b_k, \\ -z_k + \boldsymbol{y}_k \leq l_k(t_k - 1), & -z'_k + \boldsymbol{y}'_k \leq l_k(t'_k - 1), \\ z_k - \boldsymbol{y}_k \leq 0, & z'_k - \boldsymbol{y}'_k \leq 0, \\ \boldsymbol{y}_k - u_k \cdot t_k \leq 0, & \boldsymbol{y}'_k - u_k \cdot t'_k \leq 0, \\ t_k \in \{0,1\}, & t'_k \in \{0,1\}, \\ y_\eta = W_\eta \boldsymbol{y}_{\eta-1} + b_\eta, & y'_\eta = W_\eta \boldsymbol{y}'_{\eta-1} + b_\eta, \\ \boldsymbol{y}_k \geq 0, \quad \boldsymbol{y}'_k \geq 0, & k = 1, \ldots, \eta - 1. \end{cases} \tag{30}$$

However, there may exist nonlinear terms in transition function f involved in the condition in Equation (25). To handle the nonlinearity, a piecewise linear approximation method is utilized. For a non-linear function $g(x)$ with $x \in [l_x, u_x]$, it can be approximated by two piecewise linear functions $g_l(x)$, $g_u(x)$ with the controlled approximation error ζ, defined as

$$g_l(x) \leq g(x) \leq g_u(x), |g_u(x) - g_l(x)| \leq \zeta, \forall x \in [l_x, u_x].$$

Consider the condition in Equation (25) again. By utilizing the piecewise linear approximation method, the optimization problem in Equation (30) can be transformed into an MILP problem and be solved via the optimizer Gurobi. Thus, we can obtain the optimum $\tilde{\rho}_{Evol}$ and the corresponding optimal solution $\tilde{\boldsymbol{x}}_{Evol}$ together with the maximum relative error ξ_{Evol}. If $\tilde{\rho}_{Evol} \geq 0$ while the maximum relative error $\xi_{Evol} < 1$, then the globally optimum $\rho^*_{Evol} \geq 0$, and the condition in Equation (25) is verified to be satisfied. Otherwise, we randomly sample a set of points in a neighborhood of the optimal solution $\tilde{x}_{Evol}$ and add those data points that violate the condition in Equation (25) to the training dataset D_3 in the next iteration to refine the networks.

3.3. Algorithm

In the previous section, we introduced the complete synthesis process of candidates B, p, and q and the MILP-based verification method. Next, the synthesis process of the barrier certificate and relation functions is given in Algorithm 1.

Algorithm 1 Barrier certificate and relation function synthesis.

Require:
Dt-cs Π, control input $\boldsymbol{u}$, maximum epoch number n_{epoch}, data batches n_{batch}, loss function parameters α_{1-5}, c_{1-4}, learning rate lr.

Ensure:
B, p, q

1: Initialize parameters of neural network B, p and q
2: Generate training datasets D_1, D_2, D_3, D_4 and generate n_{batch} batches
3: for $i = 1$ to n_{epoch} do:
4: $L_{epoch} = 0$;
5: for $j = 1$ to n_{batch} do:
6: calculate L_{epoch} according to Equation (22);
7: update(B, p, q);
8: end for
9: If $L_{epoch} = 0$:
10: return candidate B, p, q;
11: Validation of candidate B, p, q by solving MILP problems (28)–(30);
12: The optimizer returns the optimal solution; $\tilde{\rho}_{Init}$, $\tilde{\rho}_{Unsafe}$, $\tilde{\rho}_{Evol}$ and the corresponding input $\tilde{\boldsymbol{x}}_{Init}$, $\tilde{\boldsymbol{x}}_{Unsafe}$, $\tilde{\boldsymbol{x}}_{Evol}$;
13: If $\tilde{\rho}_{Init}$, $\tilde{\rho}_{Unsafe}$, $\tilde{\rho}_{Evol}$ have the expected signs and $\tilde{\xi}_{Init}$, $\tilde{\xi}_{Unsafe}$, $\tilde{\xi}_{Evol} < 1$:
14: Return *success* and B, p, q;
15: Else:
16: For the violation $\tilde{\boldsymbol{x}}_{Init}$, $\tilde{\boldsymbol{x}}_{Unsafe}$, $\tilde{\boldsymbol{x}}_{Evol}$, sampling around them and add them to the training set;
17: Repeat 3–10;
18: **return** *failure*;

First, the respective weights among the layers and biases of neural networks B, p, and q are stochastically initialized with a standard Gaussian distribution. Then, the training datasets are generated (lines 1–2). In each epoch, with the loss function according to Equation (22) and the Adam [41] optimizer, the weights and biases of the networks are updated in a backpropagation. By minimizing L to zero, we find the candidates B, p, and q (lines 3–10). With the help of the optimization tool Gurobi, the verification of candidates B, p, and q can be transformed into the optimization problems in Equations (28)–(30). We find the current optimal solutions $\tilde{\rho}_{Init}$, $\tilde{\rho}_{Unsafe}$, $\tilde{\rho}_{Evol}$ and the corresponding inputs $\tilde{\boldsymbol{x}}_{Init}$, $\tilde{\boldsymbol{x}}_{Unsafe}$, $\tilde{\boldsymbol{x}}_{Evol}$ respectively. If the current optimal solutions $\tilde{\rho}_{Init}$, $\tilde{\rho}_{Unsafe}$, $\tilde{\rho}_{Evol}$ have the expected signs, and $\tilde{\xi}_{Init}$, $\tilde{\xi}_{Unsafe}$, $\tilde{\xi}_{Evol} < 1$, then the candidates B, p, and q are globally consistent with the constraints in the conditions in Equations (23)–(25). Then, we return *success* and B, p, and q (Line 14). Otherwise, the sampling data points around the counterexamples are returned and added to the corresponding training datasets to refine the networks (lines 16–17). The training method based on Counterexample-Guided Inductive Synthesis (CEGIS) can train the neural network quickly and accurately. The candidates are trained and refined continuously until finding the real B, p, and q or reaching the maximum epoch number. Figure 2 is the flowchart of Algorithm 1.

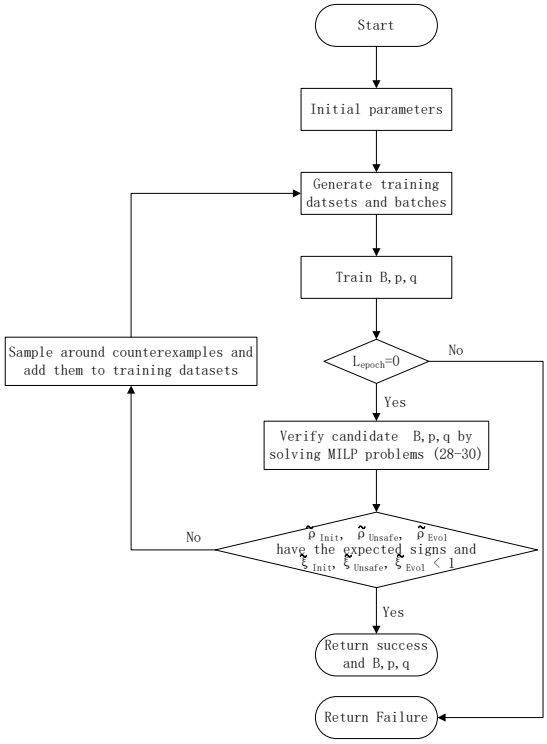

Figure 2. The flowchart of Algorithm 1.

4. Experimental Results and Analysis

We have implemented an augmented barrier certificate synthesis tool based on Algorithm 1 using the Pytorch 1.7 platform and Gurobi 9.5. We apply our tool to verify a number of benchmark examples from the literature comprising both polynomial and non-polynomial dynamics and compare it with the SOS programming method proposed in [11].

4.1. Examples

Vehicle Model. Consider the following vehicle model [11]:

$$\Pi : \begin{cases} x_1(t+1) = x_1(t) + \Delta\tau x_2(t) + 0.5\Delta\tau^2 u(t) \\ x_2(t+1) = x_2(t) + \Delta\tau u(t) \\ y(t) = h(x(t)) \end{cases}$$

where x_1 and x_2 denote the absolute position and velocity of the vehicle, respectively, u is the control input (acceleration), and $\Delta\tau$ is the sampling time. The output is assumed to be the position of the vehicle on the road (i.e., $h(x(t)) = x_1(t)$). Suppose that the initial positions of the vehicle contain critical information which needs to be kept secret and there is a malicious intruder who can observe the behavior of the vehicle and intends to carry out an attack. Thus, we aim to verify the δ-approximate initial-state opacity property for this system, where δ captures the security guarantee level in terms of the measurement precision of the intruder.

Suppose that state space $\mathbb{X} = [0, 10] \times [0, 0.1]$, the initial set $\mathbb{X}_0 = [0, 10] \times \{0\}$, the secret set $\mathbb{X}_s = [0, 1] \times [0, 0.1]$, the non-secret set $\mathbb{X}_{ns} = [1, 10] \times [0, 0.1]$, and the control input set $\mathbb{U} = [-0.05, 0.05]$. For the augmented system $\Pi \times \Pi$, the initial set $\widetilde{\Theta}_0 = \{\boldsymbol{x} \in [0, 1] \times \{0\}$ and $\hat{\boldsymbol{x}} \in [1, 10] \times \{0\} | \hat{\boldsymbol{x}} = p(\boldsymbol{x}), (x_1 - \hat{x}_1)^2 \leq \delta^2\}$, where $p(\boldsymbol{x})$ is a relation function defined in Equation (10) and the unsafe set $\Theta_u = \{(\boldsymbol{x}; \hat{\boldsymbol{x}}) \in \mathbb{X} \times \mathbb{X} | (x_1 - \hat{x}_1)^2 > \delta^2\}$.

Now, we set the measurement accuracy to be $\delta = 1$ and search for the augmented barrier certificate B satisfying the conditions in Theorem 2. By applying our approach, we successfully obtain the neural augmented barrier certificate B and the neural relation functions p and q, which are represented as once-hidden layer ReLU-activated neural

networks with 16 neurons per layer. Therefore, the δ-approximate initial-state opacity of the system Π is guaranteed.

Figure 3 (left) depicts a phase portrait of the augmented system $\Pi \times \Pi$ with $x_2 = \hat{x}_2 = 0$. It is seen that in the projection space on the position plane, the zero sublevel set of the barrier certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$ (the yellow surface) separates Θ_u (the red region) from all trajectories starting from $\widetilde{\Theta}_0$ (the pink line). Note that the initial set $\widetilde{\Theta}_0$ (the pink line), described by the neural relation function p, is a subset of the initial set Θ_0 (the green triangle), which is defined in [11]. This means that our relaxed augmented barrier certificate yields a weaker sufficient condition for the δ-approximate initial-state opacity.

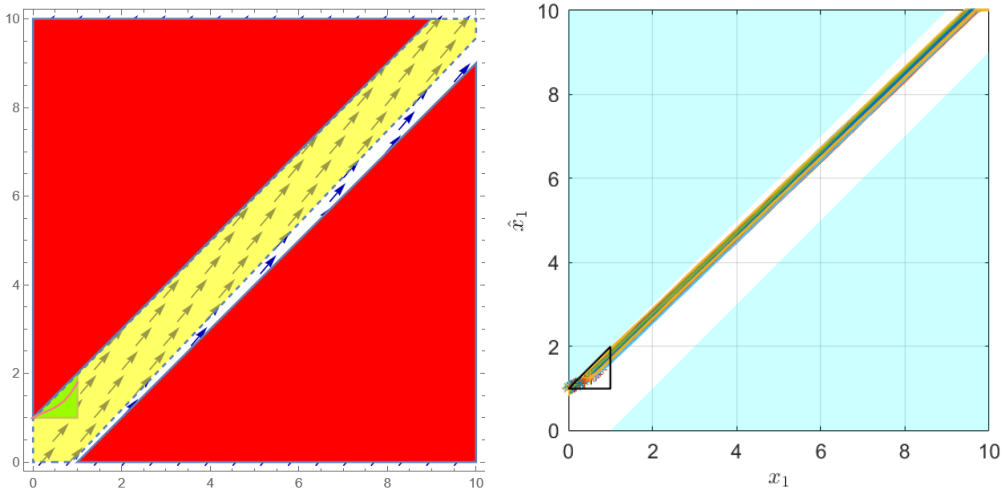

Figure 3. (**Left**) Phase portrait of Example 1 with $x_2 = \hat{x}_2 = 0$. The pink line and the green, red, and yellow regions are the initial sets $\widetilde{\Theta}_0$ and Θ_0, the unsafe set Θ_u and the sub-level set of $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$, respectively. (**Right**) State runs of $\Pi \times \Pi$ on the projection of the position plane starting from $\widetilde{\Theta}_0$. The blue region represents the unsafe set Θ_u.

Figure 3 (right) shows the projection of a number of state trajectories on the position plane of the augmented system $\Pi \times \Pi$ starting from the random initial states in $\widetilde{\Theta}_0$ under control input $\hat{\boldsymbol{u}} = q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u})$, with $\boldsymbol{u}$ taking values in $\mathbb{U}$. It can be seen that any trajectory starting from $\widetilde{\Theta}_0$ does not reach the unsafe set Θ_u (the blue region) as the time increases.

Three-dimensional model. In this example, we consider the following three-dimensional system:

$$\Pi : \begin{cases} x_1(t+1) = x_1(t) + 0.5x_2(t) + x_3(t) + 0.5u(t) \\ x_2(t+1) = x_2(t) + 0.5u(t) \\ x_3(t+1) = 0.2(x_3(t))^3 + 0.05u(t) \\ y(t) = x_1(t). \end{cases}$$

Suppose that state space $\mathbb{X} = [0,4] \times [0,0.1] \times [0,0.5]$, the initial set $\mathbb{X}_0 = \mathbb{X}$, the secret set $\mathbb{X}_s = [0,0.8] \times [0,0.1] \times [0,0.5]$, the non-secret set $\mathbb{X}_{ns} = [0.8,4] \times [0,0.1] \times [0,0.5]$, and the control input set $\mathbb{U} = [0.03, 0.05]$. For the augmented system $\Pi \times \Pi$, the initial set $\widetilde{\Theta}_0 = \{\boldsymbol{x} \in [0,0.8] \times [0,0.1] \times [0,0.5], \hat{\boldsymbol{x}} \in [0.8,4] \times [0,0.1] \times [0,0.5] | \hat{\boldsymbol{x}} = p(\boldsymbol{x}), (x_1 - \hat{x}_1)^2 \leq \delta^2\}$, where $p(\boldsymbol{x})$ is a relation function defined in Equation (10) and the unsafe set $\Theta_u = \{(\boldsymbol{x}; \hat{\boldsymbol{x}}) \in \mathbb{X} \times \mathbb{X} | (x_1 - \hat{x}_1)^2 > \delta^2\}$. We aim to verify the δ-approximate initial-state opacity property for the system, where $\delta = 2.2$ captures the security guarantee level in terms of the measurement precision of the intruder.

It suffices to synthesize certificate functions B, p, and q, satisfying the conditions in Theorem 3. By applying our approach, we successfully obtain the neural augmented barrier certificate B, represented as two hidden layer ReLU activated neural networks with 16 neurons per layer, and the neural relation functions p and q, represented as one

hidden layer ReLU activated neural networks with 10 neurons per layer. Therefore, the δ-approximate initial-state opacity of the given system Π is guaranteed.

Figure 4 (left) depicts phase portrait of the augmented system $\Pi \times \Pi$ with $x_2 = \hat{x}_2 = 0.05$ and $x_3 = \hat{x}_3 = 0.1$. It is seen that in the projection space on the position plane, the zero sub-level set of the barrier certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$ (the yellow surface) separates Θ_u (the red region) from all trajectories starting from $\widetilde{\Theta}_0$ (the pink line). Compared with the initial set Θ_0 (the green triangle), which is defined in [11], the initial set $\widetilde{\Theta}_0$ (the pink line) is much smaller, which means that our relaxed augmented barrier certificate yields a weaker sufficient condition for the δ-approximate initial-state opacity.

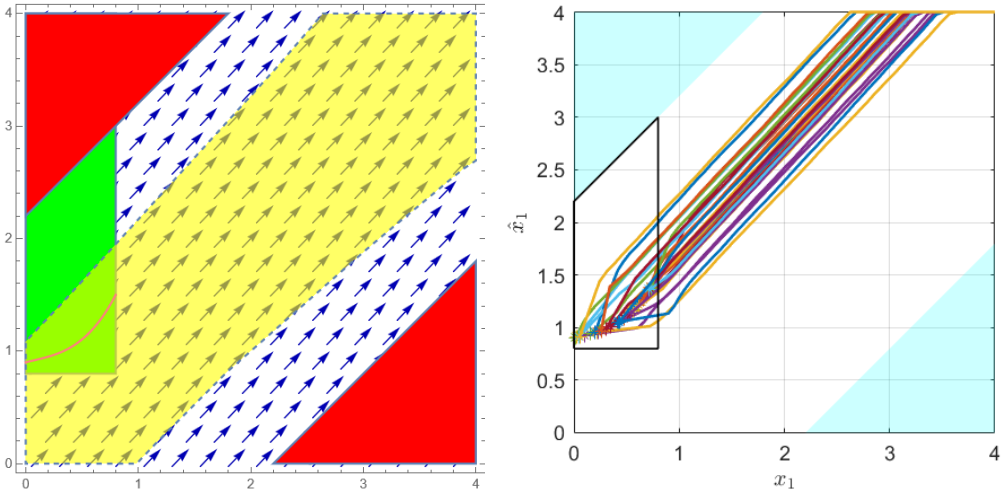

Figure 4. (**Left**) Phase portrait of Example 2 with $x_2 = \hat{x}_2 = 0.05$ and $x_3 = \hat{x}_3 = 0.05$. The pink line and the green, red, and yellow regions are the initial sets $\widetilde{\Theta}_0$ and Θ_0, the unsafe set Θ_u, and the sub-level set of $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$, respectively. (**Right**) State runs of $\Pi \times \Pi$ on the projection of the x_1 plane starting from $\widetilde{\Theta}_0$. The blue region represents the unsafe set Θ_u.

Figure 4 (right) shows the projection of a number of state trajectories on the position plane of the augmented system $\Pi \times \Pi$ starting from the random initial states in $\widetilde{\Theta}_0$ under control input $\hat{\boldsymbol{u}} = q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u})$, with $\boldsymbol{u}$ taking values in $\mathbb{U}$. It can be seen that all such trajectories do not reach the unsafe set Θ_u (the blue region).

Non-polynomial model. Consider a non-polynomial system Π:

$$\Pi = \begin{cases} x_1(t+1) = x_1(t) + x_2(t) + u(t) \\ x_2(t+1) = e^{x_2(t)} + u(t) - 1 \\ y(k) = x_1(k). \end{cases} \tag{31}$$

Suppose that the state set $\mathbb{X} = [1, 5] \times [0, 0.1]$, the initial set $\mathbb{X}_0 = [1, 5] \times [0, 0.1]$, the secret set $\mathbb{X}_s = [1, 1.1] \times [0, 0.1]$, the non-secret set $\mathbb{X}_{ns} = \mathbb{X} \backslash \mathbb{X}_s$, and the control input set $\mathbb{U} = [0.03, 0.05]$. We aim to verify the δ-approximate initial-state opacity property for the system, where $\delta = 0.95$ captures the security guarantee level in terms of the measurement precision of the intruder.

For the augmented system $\Pi \times \Pi$, the initial set $\widetilde{\Theta}_0 = \{\boldsymbol{x} \in [1, 1.1] \times [0, 0.1], \hat{\boldsymbol{x}} \in [1.1, 5] \times [0, 0.1] | \hat{\boldsymbol{x}} = p(\boldsymbol{x}), (x_1 - \hat{x}_1)^2 \leq \delta^2\}$, where $p(\boldsymbol{x})$ is a relation function defined in Equation (10) and the unsafe set $\Theta_u = \{[\boldsymbol{x}; \hat{\boldsymbol{x}}] \in \mathbb{X} \times \mathbb{X} | (x_1 - \hat{x}_1)^2 > \delta^2\}$. It suffices to synthesize certificate functions B, p, and q, satisfying the conditions in Theorem 3. By applying our approach, we successfully obtain the neural augmented barrier certificate B, represented as two hidden layer ReLU activated neural networks with 16 neurons per layer, and the neural relation functions p and q, represented as one hidden layer ReLU activated neural networks with 16 neurons per layer. Therefore, the δ-approximate initial-state opacity of the given system Π is guaranteed.

Figure 5 (left) depicts a phase portrait of the augmented system $\Pi \times \Pi$ with $x_2 = \hat{x}_2 = 0$. It is seen that in the projection space on the position plane, the zero sub-level set of the barrier certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$ (the yellow surface) separates Θ_u (the red region) from all trajectories starting from $\widetilde{\Theta}_0$ (the pink line). Clearly, the initial set $\widetilde{\Theta}_0$ (the pink line) is much smaller than the initial set Θ_0 (the green triangle), which is defined in [11]. Figure 5 (right) shows the projection of a number of state trajectories on the position plane of the augmented system $\Pi \times \Pi$, starting from the random initial states in $\widetilde{\Theta}_0$ under control input $\hat{\boldsymbol{u}} = q(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u})$, with $\boldsymbol{u}$ taking values in $\mathbb{U}$. It is obvious that such trajectories does not reach the unsafe set Θ_u (the blue region) as the time increases.

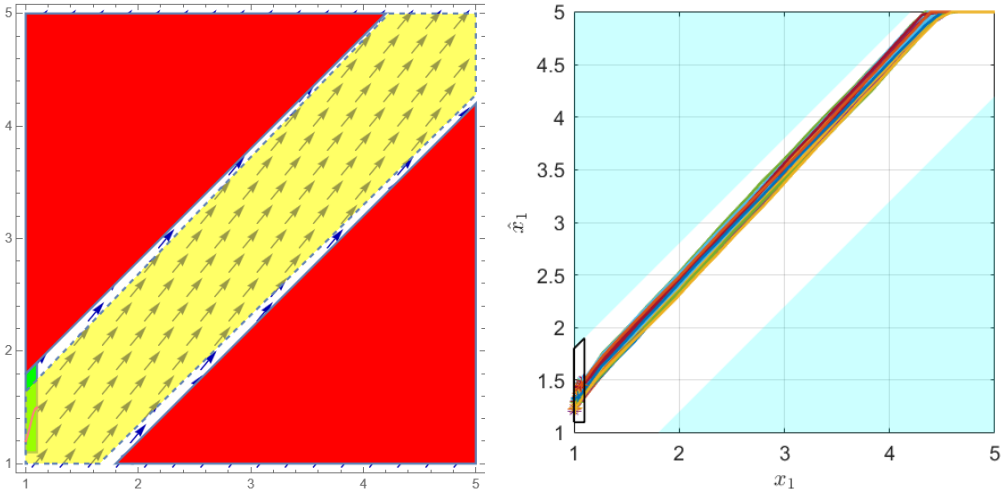

Figure 5. (**Left**) Phase portrait of Example 2 with $x_2 = \hat{x}_2 = 0.05$. The pink line and the green, red, and yellow regions are the initial sets $\widetilde{\Theta}_0$ and Θ_0, the unsafe set Θ_u, and the sub-level set of $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$, respectively. (**Right**) State runs of $\Pi \times \Pi$ on the projection of the x_1 plane starting from $\widetilde{\Theta}_0$. The blue region represents the unsafe set Θ_u.

4.2. Performance Evaluation

The statistics of our case studies are summarized in Table 1. $|x|$ denotes the number of augmented system variables, B, p, and q denote the structure of the neural network barrier certificates and relation functions, and lr denotes the learning rate. $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, and α_5 denote the weights of the sub-losses. The tolerances c_1, c_2, c_3, and c_4 are used for training neural barrier certificates and relation functions.

Table 1. Performance evaluation.

Ex	$\|x\|$	B	p	q	lr	α_3	$\alpha_{1,2,4,5}$	c_1	c_2	c_3	c_4
Ex 1	4	4-16-1	2-16-2	5-16-1	0.01	1.2	1	0	0.01	0.01	0.001
Ex 2	6	6-16-16-1	3-10-3	7-10-1	0.1	1.5	1	0.01	0.05	0.01	0.001
Ex 3	4	4-16-1	2-16-2	5-16-1	0.01	1.2	1	0.01	0.05	0.01	0.001

For the vehicle model, note that the SOS programming method is presented in [11] for computing the augmented barrier certificates, but it lacks soundness due to numerical errors. More concretely, for the augmented barrier certificate $B(\boldsymbol{x}, \hat{\boldsymbol{x}})$ and corresponding control policy $\hat{\boldsymbol{u}}(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{u})$ given in [11], there exist some counterexamples, such as (x_1, $\hat{x}_1$, x_2, $\hat{x}_2$) = (0.5, 1.25, 0, 0), (0, 1.0312, 0.0625, 0.0625), (0.25, 0.0625, 0.0625, 0.0977, −0.0488), which violate the three conditions of the augmented barrier certificates defined in Theorem 1.

For the three-dimensional model, we also apply the SOS programming method presented in [11] to search for the augmented barrier certificate and the relation functions. However, the SOS programming method cannot yield such certificate functions with a degree less than six within one hour. The reason for this is that the size of the optimization

problem derived from opacity verification grows exponentially with the number of state variables and the degree of the polynomial involved in the system, which imposes a serious size limitation on the system to be verified. For the augmented system $\Pi \times \Pi$, the number of state variables is six, and the degree of the involved polynomials is three. Therefore, due to the large size, the bilinear SOS program associated to opacity verification cannot be solved efficiently.

For the non-polynomial model, note that SOS programming method in [11] cannot deal with this non-polynomial model directly, and one should combine a recasting procedure to transform it into an equivalent polynomial model with a much higher dimension.

It is interesting to investigate what impact a less conservative sufficient condition on the synthesis of neural networks as barrier certificates and relation functions to verify the approximate initial-state opacity has. For the three examples mentioned above, there still exist counterexamples after multiple iterations of synthetic neural barrier certificates and relation functions based on the conditions in [11], and there are also counterexamples based on the SOS programming. This is because the conditions in [11] are too strict on the initial region. Based on Theorem 3, we can synthesize the valid neural barrier certificates and relation functions successfully with no counterexamples. As shown in the left part of Figures 3–5, the initial region becomes much smaller under the premise of satisfying the approximate initial-state opacity.

5. Conclusions

In this paper, a method for synthesizing neural augmented barrier certificates to verify the approximate initial-state opacity of discrete-time control systems is presented. The existence of augmented barrier certificates can serve as a guarantee for approximate initial-state opacity. We propose a novel form of augmented barrier certificates that generates a weaker sufficient condition for the approximate initial-state opacity. We then employ an algorithmic framework involving a *learner* for training neural certificates via the deep learning method and alternate a *verifier* for verifying the candidate certificates by solving several mixed-integer linear programs. The *verifier* either ensures the validity of the candidate certificates or yields counterexamples for further training. The effectiveness of the proposed approach is illustrated via several numerical examples.

Author Contributions: Conceptualization, W.L. and Y.J.; methodology, W.L.; software, S.W.; data curation, S.W.; validation, W.L. and Y.J.; writing—original draft preparation, S.W. and M.D.; writing—review and editing, S.W., M.D. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Zhejiang Provincial Natural Science Foundation of China under Grant LY20F020020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Sun, X.; Khedr, H.; Shoukry, Y. Formal verification of neural network controlled autonomous systems. In Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, Montreal, QC, Canada, 16–18 April 2019; pp. 147–156.
2. Corsi, D.; Marchesini, E.; Farinelli, A. Formal verification of neural networks for safety-critical tasks in deep reinforcement learning. In Proceedings of the Uncertainty in Artificial Intelligence, PMLR, Virtual Event, 27–30 July 2021; pp. 333–343.
3. Focardi, R.; Gorrieri, R. A taxonomy of trace-based security properties for CCS. In Proceedings of the Computer Security Foundations Workshop VII, Franconia, NH, USA, 14–16 June 1994; pp. 126–127.
4. Sandberg, H.; Amin, S.; Johansson, K. Cyberphysical Security in Networked Control Systems: An Introduction to the Issue. *Control Syst. IEEE* **2015**, *35*, 20–23.

5. Mazaré, L. Using unification for opacity properties. In Proceedings of the 4th IFIP WG1, Barcelona, Spain, 22–27 August 2004; Volume 7, pp. 165–176.
6. Lafortune, S.; Lin, F.; Hadjicostis, C.N. On the history of diagnosability and opacity in discrete event systems. *Annu. Rev. Control* **2018**, *45*, 257–266. [CrossRef]
7. Wu, Y.C.; Lafortune, S. Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discret. Event Dyn. Syst.* **2013**, *23*, 307–339. [CrossRef]
8. Yin, X.; Zamani, M.; Liu, S. On approximate opacity of cyber-physical systems. *IEEE Trans. Autom. Control* **2020**, *66*, 1630–1645. [CrossRef]
9. Yang, J.; Deng, W.; Qiu, D.; Jiang, C. Opacity of networked discrete event systems. *Inf. Sci.* **2021**, *543*, 328–344. [CrossRef]
10. Balun, J.; Masopust, T. Comparing the notions of opacity for discrete-event systems. *Discret. Event Dyn. Syst.* **2021**, *31*, 553–582. [CrossRef]
11. Liu, S.; Zamani, M. Verification of Approximate Opacity via Barrier Certificates. *IEEE Control Syst. Lett.* **2021**, *5*, 1369–1374. [CrossRef]
12. Anand, M.; Murali, V.; Trivedi, A.; Zamani, M. Formal Verification of Control Systems against Hyperproperties via Barrier Certificates. *arXiv* **2021**, arXiv:2105.05493.
13. Prajna, S.; Jadbabaie, A. Safety verification of hybrid systems using barrier certificates. In Proceedings of the International Workshop on Hybrid Systems: Computation and Control, Philadelphia, PA, USA, 25–27 March 2004; pp. 477–492.
14. Kong, H.; He, F.; Song, X.; Hung, W.N.; Gu, M. Exponential-condition-based barrier certificate generation for safety verification of hybrid systems. In Proceedings of the International Conference on Computer Aided Verification, Saint Petersburg, Russia, 13–19 July 2013; pp. 242–257.
15. Ames, A.D.; Coogan, S.; Egerstedt, M.; Notomista, G.; Sreenath, K.; Tabuada, P. Control barrier functions: Theory and applications. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 3420–3431.
16. Zhang, Y.; Yang, Z.; Lin, W.; Zhu, H.; Chen, X.; Li, X. Safety verification of nonlinear hybrid systems based on bilinear programming. *IEEE Trans. Comput. -Aided Des. Integr. Circuits Syst.* **2018**, *37*, 2768–2778. [CrossRef]
17. Zhao, Q.; Chen, X.; Zhang, Y.; Sha, M.; Yang, Z.; Lin, W.; Tang, E.; Chen, Q.; Li, X. Synthesizing ReLU neural networks with two hidden layers as barrier certificates for hybrid systems. In Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control, Nashville, TN, USA, 19–21 May 2021; pp. 1–11.
18. Peruffo, A.; Ahmed, D.; Abate, A. Automated and formal synthesis of neural barrier certificates for dynamical models. In Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Luxembourg, 27 March–1 April 2021; pp. 370–388.
19. Feng, L. Opacity of discrete event systems and its applications. *Automatica* **2011**, *47*, 496–503.
20. Saboori, A.; Hadjicostis, C.N. Verification of initial-state opacity in security applications of discrete event systems. *Inf. Sci.* **2013**, *246*, 115–132. [CrossRef]
21. Saboori, A.; Hadjicostis, C.N. Verification of *K*-step opacity and analysis of its complexity. *IEEE Trans. Autom. Sci. Eng.* **2011**, *8*, 549–559. [CrossRef]
22. Saboori, A.; Hadjicostis, C.N. Notions of security and opacity in discrete event systems. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 5056–5061.
23. Saboori, A.; Hadjicostis, C.N. Verification of infinite-step opacity and complexity considerations. *IEEE Trans. Autom. Control* **2011**, *57*, 1265–1269. [CrossRef]
24. Xie, Y.; Yin, X.; Li, S. Opacity enforcing supervisory control using non-deterministic supervisors. *IEEE Trans. Autom. Control.* **2021**. [CrossRef]
25. Xie, Y.; Yin, X.; Li, S. Optimal Synthesis of Opacity-Enforcing Supervisors for Qualitative and Quantitative Specifications. *arXiv* **2021**, arXiv:2102.01402.
26. Liu, S.; Swikir, A.; Zamani, M. Compositional verification of initial-state opacity for switched systems. In Proceedings of the 2020 59th IEEE Conference on Decision and Control (CDC), Jeju Island, Korea, 14–18 December 2020; pp. 2146–2151.
27. Noori-Hosseini, M.; Lennartson, B.; Hadjicostis, C. Compositional Visible Bisimulation Abstraction Applied to Opacity Verification. *IFAC-PapersOnLine* **2018**, *51*, 434–441. [CrossRef]
28. Yin, X.; Li, Z.; Wang, W.; Li, S. Infinite-step opacity and K-step opacity of stochastic discrete-event systems. *Automatica* **2019**, *99*, 266–274. [CrossRef]
29. Tong, Y.; Li, Z.; Seatzu, C.; Giua, A. Verification of state-based opacity using Petri nets. *IEEE Trans. Autom. Control* **2016**, *62*, 2823–2837. [CrossRef]
30. Tasdighi Kalat, S.; Liu, S.; Zamani, M. Modular Verification of Opacity for Interconnected Control Systems via Barrier Certificates. *IEEE Control Syst. Lett.* **2022**, *6*, 890–895. doi: 10.1109/LCSYS.2021.3087103. [CrossRef]
31. Chédor, S.; Morvan, C.; Pinchinat, S.; Marchand, H. Diagnosis and opacity problems for infinite state systems modeled by recursive tile systems. *Discret. Event Dyn. Syst.* **2015**, *25*, 271–294. [CrossRef]
32. Liu, S.; Yin, X.; Zamani, M. On a notion of approximate opacity for discrete-time stochastic control systems. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 5413–5418.
33. Tjeng, V.; Xiao, K.; Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. *arXiv* **2017**, arXiv:1711.07356.

34. Dutta, S.; Jha, S.; Sankaranarayanan, S.; Tiwari, A. Output range analysis for deep feedforward neural networks. In Proceedings of the NASA Formal Methods Symposium, Newport News, VA, USA, 17–19 April 2018; pp. 121–138.
35. Zhao, H.; Zeng, X.; Chen, T.; Liu, Z.; Woodcock, J. Learning safe neural network controllers with barrier certificates. *Form. Asp. Comput.* **2021**, *33*, 437–455. [CrossRef]
36. Katz, G.; Barrett, C.; Dill, D.L.; Julian, K.; Kochenderfer, M.J. Reluplex: An efficient SMT solver for verifying deep neural networks. In Proceedings of the International Conference on Computer Aided Verification, Heidelberg, Germany, 24–28 July 2017; pp. 97–117.
37. Huang, X.; Kwiatkowska, M.; Wang, S.; Wu, M. Safety verification of deep neural networks. In Proceedings of the International Conference on Computer Aided Verification, Heidelberg, Germany, 24–28 July 2017; pp. 3–29.
38. Rössig, A.; Petkovic, M. Advances in verification of ReLU neural networks. *J. Glob. Optim.* **2021**, *81*, 109–152. [CrossRef]
39. Botoeva, E.; Kouvaros, P.; Kronqvist, J.; Lomuscio, A.; Misener, R. Efficient verification of relu-based neural networks via dependency analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 3291–3299.
40. Zhao, Q.; Chen, X.; Zhao, Z.; Zhang, Y.; Tang, E.; Li, X. Verifying Neural Network Controlled Systems Using Neural Networks. In Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control, Milan, Italy, 4–6 May 2022; pp. 1–11.
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.