

# Safety Verification for Neural Networks Based on Set-boundary Analysis

Zhen Liang<sup>1</sup>, Dejin Ren<sup>2</sup>, Wanwei Liu<sup>1</sup>, Ji Wang<sup>1</sup>,  
Wenjing Yang<sup>1</sup>, and Bai Xue<sup>2</sup> \*

<sup>1</sup> College of Computer Science and Technology, National University of Defense  
Technology, Changsha, China

{liangzhen, wliu, jiwang, wenjing.yang}@nudt.edu.cn

<sup>2</sup> Institute of Software CAS, Beijing, China

{rendj, xuebai}@ios.ac.cn

**Abstract.** Neural networks (NNs) are increasingly applied in safety-critical systems such as autonomous vehicles. However, they are fragile and are often ill-behaved. Consequently, their behaviors should undergo rigorous guarantees before deployment in practice. In this paper we propose a set-boundary reachability method to investigate the safety verification problem of NNs from a topological perspective. Given an NN with an input set and a safe set, the safety verification problem is to determine whether all outputs of the NN resulting from the input set fall within the safe set. In our method, the homeomorphism property of NNs is mainly exploited, which establishes a relationship mapping boundaries to boundaries. The exploitation of this property facilitates reachability computations via extracting subsets of the input set rather than the entire input set, thus controlling the wrapping effect in reachability analysis and facilitating the reduction of computation burdens for safety verification. The homeomorphism property exists in some widely used NNs such as invertible NNs. Notable representations are invertible residual networks (i-ResNets) and Neural ordinary differential equations (Neural ODEs). For these NNs, our set-boundary reachability method only needs to perform reachability analysis on the boundary of the input set. For NNs which do not feature this property with respect to the input set, we explore subsets of the input set for establishing the local homeomorphism property, and then abandon these subsets for reachability computations. Finally, some examples demonstrate the performance of the proposed method.

**Keywords:** Safe verification · Neural networks · Boundary analysis · Homeomorphism.

## 1 Introduction

Machine learning has seen rapid growth due to the high amount of data produced in many industries and the increase in computation power. NNs have emerged

---

\* Corresponding author

as a leading candidate computation model for machine learning, which promote the prosperity of artificial intelligence in various fields, such as computer vision [38,7], natural language processing [49,23] and so on. In recent years, NNs are increasingly applied in safety critical systems. For example, a neural network has been implemented in the ACAS Xu airborne collision avoidance system for unmanned aircraft, which is a highly safety-critical system and currently being developed by the Federal Aviation Administration. Consequently, to gain users' trust and ease their concerns, it is of vital importance to ensure that NNs are able to produce safe outputs and satisfy the essential safety requirements before the deployment.

Safety verification of NNs, which determines whether all outputs of an NN satisfy specified safety requirements via computing output reachable sets, has attracted a huge attention from different communities such as machine learning [29,1], formal methods [19,39,28], and security [41,11]. Because NNs are generally large, nonlinear, and non-convex, exact computation of output reachable sets is challenging. Although there are some methods on exact reachability analysis such as SMT-based [24] and polyhedron-based approaches [43,40], they are usually time-consuming and do not scale well. Moreover, these methods are limited to NNs with ReLU activation functions. Consequently, over-approximate reachability analysis, which mainly involves the computation of super sets of output reachable sets, is often resorted to in practice. The over-approximate analysis is usually more efficient and can be applied to more general NNs beyond ReLU ones. Due to these advantages, an increasing attention has been attracted and thus a large amount of computational techniques have been developed for over-approximate reachability analysis [27].

Overly conservative over-approximations, however, often render many safety properties unverifiable in practice. This conservatism mainly results from the wrapping effect, which is the accumulation of over-approximation errors through layer-by-layer propagation. As the extent of the wrapping effect correlates strongly with the size of the input set [44], techniques that partition the input set and independently compute output reachable sets of the resulting subsets are often adapted to reduce the wrapping effect, especially for large input sets. Such partitioning may, however, produce a large number of subsets, which is generally exponential in the dimensionality. This will induce extensive demand on computation time and memory, often rendering existing reachability analysis techniques not suitable for safety verification of complex NNs in real applications. Therefore, exploring subsets of the input set rather than the entire input set could help reduce computation burdens and thus accelerate computations tremendously.

In this work we investigate the safety verification problem of NNs from the topological perspective and extend the set-boundary reachability method, which is originally proposed for verifying safety properties of systems modeled by ODEs in [45], to safety verification of NNs. In [45], the set-boundary reachability method only performs over-approximate reachability analysis on the initial set's boundary rather than the entire initial set to address safety verification problems. It was built upon the homeomorphism property of ODEs. This nice

property also widely exists in NNs, and typical NNs are invertible NNs such as neural ODEs [5] and invertible residual networks [4]. Consequently, it is straightforward to extend the set-boundary reachability method to safety verification of these NNs, just using the boundary of the input set for reachability analysis which does not involve reachability computations of interior points and thus reducing computation burdens in safety verification. Furthermore, we extend the set-boundary reachability method to general NNs via exploiting the local homeomorphism property with respect to the input set. This exploitation is instrumental for constructing a subset of the input set for reachability computations, which is gained via removing a set of points in the input set such that the NN is a homeomorphism with respect to them. The above methods of extracting subsets for performing reachability computations can also be applied to intermediate layers of NNs rather than just between the input and output layers. Finally, we demonstrate the performance of the proposed method on several examples.

Main contributions of this paper are listed as follows.

- We investigate the safety verification problem of NNs from the topological perspective. More concretely, we exploit the homeomorphism property and aim at extracting a subset of the input set rather than the entire input set for reachability computations. To the best of our knowledge, this is the first work on the use of the homeomorphism property to address safety verification problems of NNs. This might on its own open research directions on digging into topological properties of facilitating reachability computations for NNs.
- The proposed method is able to enhance the capabilities and performances of existing reachability computation methods for safety verification of NNs via reducing computation burdens. Based on the homeomorphism property, the computation burdens of solving the safety verification problem can be reduced for invertible NNs. We further show that the computation burdens can also be reduced for more general NNs via exploiting this property on subsets of the input set.

## 2 Related Work

There has been a dozen of works on safety verification of NNs. The first work on DNN verification was published in [35], which focuses on DNNs with Sigmoid activation functions via a partition-refinement approach. Later, Katz et al. [24] and [10] independently implemented Reluplex and Planet, two SMT solvers to verify DNNs with ReLU activation function on properties expressible with SMT constraints.

Recently, methods based on abstract interpretation attracts more attention, which is to propagate sets in a sound (i.e., over-approximate) way [6] and is more efficient. There are many widely used abstract domains, such as intervals [41], and star-sets [39]. A method based on zonotope abstract domains is proposed in [11], which works for any piece linear activation function with great scalability. Then, it is further improved [36] for obtaining tighter results via imposing

abstract transformation on ReLU, Tanh and Sigmoid. [36] proposed specialized abstract zonotope transformers for handling NNs with ReLU, Sigmoid and Tanh activation functions. [37] proposes an abstract domain that combines floating point polyhedra with intervals to over-approximate output reachable sets. Subsequently, a spurious region guided approach is proposed to infer tighter output reachable sets [48] based on the method in [37]. [9] abstracts an NN by a polynomial, which has the advantage that dependencies can in principle be preserved. This approach can be precise in practice for small input sets. Afterwards, [18] approximates Lipschitz-continuous NNs with Bernstein polynomials. [20] transforms a neural network with Sigmoid activation functions into a hybrid automaton and then uses existing reachability analysis methods for the hybrid automaton to perform reachability computations. [44] proposed a maximum sensitivity based approach for solving safety verification problems for multi-layer perceptrons with monotonic activation functions. In this approach, an exhaustive search of the input set is enabled by discretizing input space to compute output reachable set which consists of a union of reachtubes.

Neural ODEs were first introduced in 2018, which exhibit considerable computational efficiency on time-series modeling tasks [5]. Recent years have witnessed an increase use of them on real-world applications [26,17]. However, the verification techniques for Neural ODEs are rare and still in fancy. The first reachability technique for Neural ODEs appeared in [16], which proposed Stochastic Lagrangian reachability, an abstraction-based technique for constructing an over-approximation of the output reachable set with probabilistic guarantees. Later, this method was improved and implemented in a tool GoTube [15], which is able to perform reachability analysis for long time horizons. Since these methods only provide stochastic bounds on the computed over-approximation and thus cannot provide formal guarantees on the satisfaction of safety properties, [30] presented a deterministic verification framework for a general class of Neural ODEs with multiple continuous- and discrete-time layers.

Based on entire input sets, all the aforementioned works focus on developing computational techniques for reachability analysis and safety verification of appropriate NNs. In contrast, the present work shifts this focus to topological analysis of NNs and guides reachability computations on subsets of the input set rather than the entire input set, reducing computation burdens and thus increasing the power of existing safety verification methods for NNs. Although there are studies on topological properties of NNs [4,8,34], there is no work on the utilization of homeomorphism property to analyze their reachability and safety verification problems, to the best of our knowledge.

### 3 Preliminaries

In this section, we give an introduction on the safety verification problem of interest for NNs and homeomorphisms. Throughout this paper, given a set  $\Delta$ ,  $\Delta^\circ$ ,  $\partial\Delta$  and  $\bar{\Delta}$  respectively denotes its interior, boundary and the closure.

NNs, also known as artificial NNs, are a subset of machine learning and are at the heart of deep learning algorithms. It works by using interconnected nodes or neurons in a layered structure that resembles a human brain, and is generally composed of three layers: an input layer, hidden layers and an output layer. Mathematically, it is a mathematical function  $\mathbf{N}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , where  $n$  and  $m$  respectively denote the dimension of the input and output of the NN.

### 3.1 Problem Statement

Given an input set  $\mathcal{X}_{in}$ , the output reachable set of an NN  $\mathbf{N}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is stated by the following definition.

**Definition 1.** *For a given neural network  $\mathbf{N}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , with an input set  $\mathcal{X}_{in} \subseteq \mathbb{R}^n$ , the output reachable set  $\mathcal{R}(\mathcal{X}_{in})$  is defined as*

$$\mathcal{R}(\mathcal{X}_{in}) = \{\mathbf{y} \in \mathbb{R}^m \mid \mathbf{y} = \mathbf{N}(\mathbf{x}), \mathbf{x} \in \mathcal{X}_{in}\}.$$

The safety verification problem is formulated in Definition 2.

**Definition 2 (Safety Verification Problem).** *Given a neural network  $\mathbf{N}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , an input set  $\mathcal{X}_{in} \subseteq \mathbb{R}^n$  which is compact, and a safe set  $\mathcal{X}_s \subseteq \mathbb{R}^m$  which is simply connected, the safety verification problem is to verify that*

$$\forall \mathbf{x}_0 \in \mathcal{X}_{in}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s.$$

In topology, a simply connected set is a path-connected set where one can continuously shrink any simple closed curve into a point while remaining in it. The requirement that the safe set  $\mathcal{X}_s$  is a simply connected set is not strict, since many widely used sets such as intervals, ellipsoids, convex polyhedra and zonotopes are simply connected.

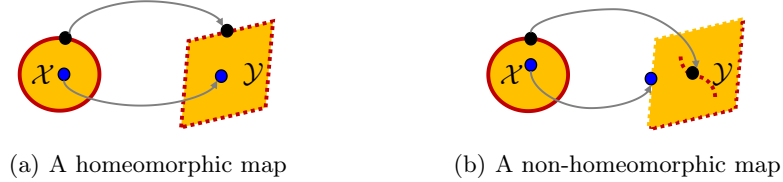
Obviously, the safety property that  $\forall \mathbf{x}_0 \in \mathcal{X}_{in}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s$  holds if and only if  $\mathcal{R}(\mathcal{X}_{in}) \subseteq \mathcal{X}_s$ . However, it is challenging to compute the exact output reachable set  $\mathcal{R}(\mathcal{X}_{in})$  and thus an over-approximation  $\Omega(\mathcal{X}_{in})$ , which is a super set of the set  $\mathcal{R}(\mathcal{X}_{in})$  (i.e.,  $\mathcal{R}(\mathcal{X}_{in}) \subseteq \Omega(\mathcal{X}_{in})$ ), is commonly resorted to in existing literature for formally reasoning about the safety property. If  $\Omega(\mathcal{X}_{in}) \subseteq \mathcal{X}_s$ , the safety property that  $\forall \mathbf{x}_0 \in \mathcal{X}_{in}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s$  holds.

### 3.2 Homeomorphisms

In this subsection, we will recall the definition of a homeomorphism, which is a map between spaces that preserves all topological properties.

**Definition 3.** *A map  $h : \mathcal{X} \rightarrow \mathcal{Y}$  with  $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$  is a homeomorphism with respect to  $\mathcal{X}$  if it is a continuous bijection and its inverse  $h^{-1}(\cdot) : \mathcal{Y} \rightarrow \mathcal{X}$  is also continuous.*

Homeomorphisms are continuous functions that preserve topological properties, which map boundaries to boundaries and interiors to interiors [32].



**Fig. 1.** Homeomorphic and non-homeomorphic maps

**Proposition 1.** *Suppose sets  $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$  are compact. If a map  $h(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$  is a homeomorphism, then  $h$  maps the boundary of the set  $\mathcal{X}$  onto the boundary of the set  $\mathcal{Y}$ , and the interior of the set  $\mathcal{X}$  onto the interior of the set  $\mathcal{Y}$ .*

Based on this property, [45] proposed a set-boundary reachability method for safety verification of ODEs, via only propagating the initial set’s boundary. Later, this method was extended to a class of delay differential equations [47].

## 4 Safety Verification Based on Boundary Analysis

In this section we introduce our set-boundary reachability method for addressing the safety verification problem in the sense of Definition 1. We first consider invertible NNs in Subsection 4.1, and then extend the method to more general NNs in Subsection 4.2.

### 4.1 Safety Verification on Invertible NNs

In this subsection we introduce our set-boundary reachability method for safety verification on invertible NNs, which relies on the homeomorphism property of these NNs.

Invertible NNs, such as i-RevNets [21], RevNets [13], i-ResNets [4] and Neural ODEs [5], are NNs with invertibility by designed architectures, which can reconstruct inputs from their outputs. These NNs are continuous bijective maps. Based on the facts that  $\mathcal{X}_{in}$  is compact, they are homeomorphisms [Corollary 2.4, [22]]<sup>3</sup>. In existing literature, many invertible NNs are constructed by requiring their Jacobian determinants to be non-zero [3]. Consequently, based on the inverse function theorem [25], these NNs are homeomorphisms. In the present work, we also use Jacobian determinants to justify the invertibility of some NNs. It is noteworthy that Jacobian determinants being non-zero is a sufficient but not necessary condition for homeomorphisms and the reason why we resort to this requirement lies in the simple and efficient computations of Jacobian determinants with interval arithmetic. However, this demands the differentiability of

<sup>3</sup> A continuous bijection from a compact space onto a Hausdorff space is a homeomorphism. (Euclidean space and any subset of Euclidean space is Hausdorff.)

---

**Algorithm 1** Safety Verification Framework for Invertible NNs Based on Boundary Analysis

---

**Input:** an invertible NN  $N(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , an input set  $\mathcal{X}_{in}$  and a safe set  $\mathcal{X}_s$ .

**Output:** **Safe** or **Unknown**.

- 1: extract the boundary  $\partial\mathcal{X}_{in}$  of the input set  $\mathcal{X}_{in}$ ;
  - 2: apply existing methods to compute an over-approximation  $\Omega(\partial\mathcal{X}_{in})$ ;
  - 3: **if**  $\Omega(\partial\mathcal{X}_{in}) \subseteq \mathcal{X}_s$  **then**
  - 4:     return **Safe**
  - 5: **else**
  - 6:     return **Unknown**
  - 7: **end if**
- 

NNs. Thus, this technique of computing Jacobian determinants to determining homeomorphisms is not applicable to NNs with ReLU activation functions.

Based on the homeomorphism property of mapping the input set’s boundary onto the output reachable set’s boundary, we propose a set-boundary reachability method for safety verification of invertible NNs, which just performs the over-approximate reachability analysis on the input set’s boundary. Its computation procedure is presented in Algorithm 1.

*Remark 1.* In the second step of Algorithm 1, we may take partition operator on the input set’s boundary to refine the computed over-approximation for addressing the safety verification problem.

**Theorem 1 (Soundness).** *If Algorithm 1 returns **Safe**, the safety property in the sense of Definition 1 holds.*

*Proof.* It is equivalent to show that if  $\mathcal{R}(\partial\mathcal{X}_{in}) \subseteq \mathcal{X}_s$ ,

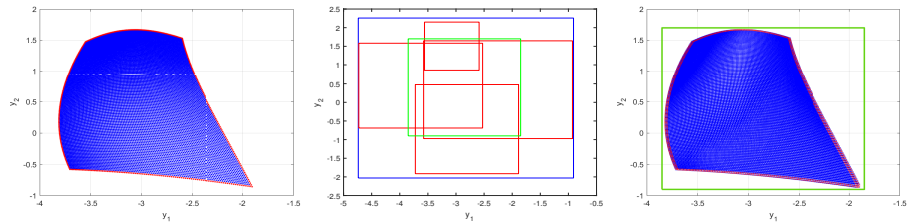
$$\forall \mathbf{x}_0 \in \mathcal{X}_{in}. N(\mathbf{x}_0) \in \mathcal{X}_s.$$

The conclusion holds by Lemma 3 in [45].

In order to enhance the understanding of Algorithm 1 and its benefits, we use a sample example to illustrate it.

*Example 1.* Consider an NN from [44], which has 2 inputs, 2 outputs and 1 hidden layer consisting of 5 neurons. The input set is  $\mathcal{X}_{in} = [0, 1]^2$ . Its boundary is  $\partial\mathcal{X}_{in} = \cup_{i=1}^4 \mathcal{B}_i$ , where  $\mathcal{B}_1 = [0, 0] \times [0, 1]$ ,  $\mathcal{B}_2 = [1, 1] \times [0, 1]$ ,  $\mathcal{B}_3 = [0, 1] \times [0, 0]$  and  $\mathcal{B}_4 = [0, 1] \times [1, 1]$ . The activation functions for the hidden layer and the output layer are **Tanh** and **PureLin** functions, respectively, whose weight matrices and bias vectors can be found in Example 1 in [44]. For this neural network, based on interval arithmetic, we can show that the determinant of the Jacobian matrix  $\frac{\partial \mathbf{y}}{\partial \mathbf{x}_0} = \frac{\partial N(\mathbf{x}_0)}{\partial \mathbf{x}_0}$  is non-zero for any  $\mathbf{x}_0 \in \mathcal{X}_{in}$ . Therefore, this NN is invertible and the map  $N(\cdot) : \mathcal{X}_{in} \rightarrow \mathcal{R}(\mathcal{X}_{in})$  is a homeomorphism with respect to the input set  $\mathcal{X}_{in}$ , leading to  $\mathcal{R}(\partial\mathcal{X}_{in}) = \partial\mathcal{R}(\mathcal{X}_{in})$ . This statement is also verified via the visualized results in Fig. 2(a).

The homeomorphism property facilitates the reduction of the wrapping effect in over-approximate reachability analysis and thus reduces computation burdens in addressing the safety verification problem in the sense of Definition 1. For this example, with the safe set  $\mathcal{X}_s = [-3.85, -1.85] \times [-0.9, 1.7]$ , we first take the input set and its boundary for reachability computations. Based on interval arithmetic, we respectively compute over-approximations  $\Omega(\mathcal{X}_{in})$  and  $\Omega(\partial\mathcal{X}_{in})$ , which are illustrated in Fig. 2(b). Although the approximation  $\Omega(\partial\mathcal{X}_{in})$  is indeed smaller than  $\Omega(\mathcal{X}_{in})$ , it still renders the safety property unverifiable. We next take partition operator for more accurate reachability computations. If the entire input set is used, we can successfully verify the safety property when the entire input set is divided into  $10^4$  small intervals of equal size. In contrast, our set-boundary reachability method just needs 400 equal partitions on the input set's boundary, significantly reducing the computation burdens. The reachability results, i.e., the computation of  $\Omega(\partial\mathcal{X}_{in})$ , are illustrated in Fig. 2(c).



(a)  $\mathcal{R}(\mathcal{X}_{in})$  and  $\mathcal{R}(\partial\mathcal{X}_{in})$  (b)  $\partial\Omega(\mathcal{X}_{in})$ ;  $\Omega(\partial\mathcal{X}_{in})$ ;  $\partial\mathcal{X}_s$  (c)  $\Omega(\mathcal{X}_{in})$ ;  $\Omega(\partial\mathcal{X}_{in})$ ;  $\partial\mathcal{X}_s$   
 estimated via Monte-Carlo method

**Fig. 2.** Illustrations on Example 1

## 4.2 Safety Verification on Non-invertible NNs

When an NN has the homeomorphism property, we can use Algorithm 1 to address the safety verification problem in the sense of Definition 1. However, not all of NNs have such a nice property. In this subsection we extend the set-boundary reachability method to safety verification of non-invertible NNs, via analyzing the homeomorphism property of NNs with respect to subsets of the input set  $\mathcal{X}_{in}$ .

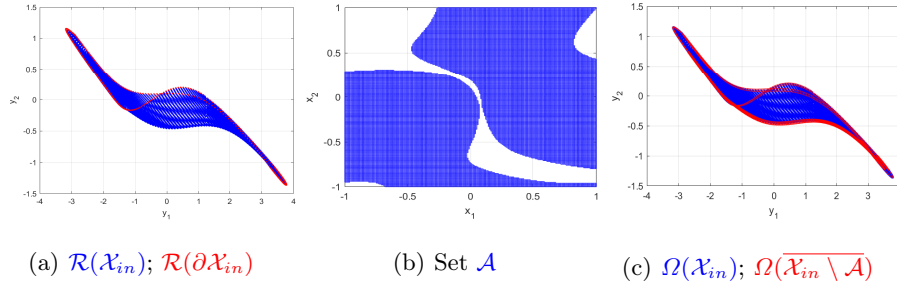
*Example 2.* Consider an NN from [42], which has 2 inputs, 2 outputs and 1 hidden layer consisting of 7 neurons. The input set is  $\mathcal{X}_{in} = [-1, 1]^2$ . The activation functions for the hidden layer and the output layer are `Tanh` and `Purelin` functions, respectively, whose weight matrices and bias vectors can be found in



Example 4.3 in [42]. For this neural network, the boundary of the output reachable set, i.e.,  $\partial\mathcal{R}(\mathcal{X}_{in})$ , is not included in the output reachable set of the input set's boundary  $\mathcal{R}(\partial\mathcal{X}_{in})$ . This statement is visualized in Fig. 3(a).

Example 2 presents us an NN, whose mapping does not admit the homeomorphism property with respect to the input set and the output reachable set. However, the NN may feature the homeomorphism property with respect to a subset of the input set. This is illustrated in Example 3.

*Example 3.* Consider the NN in Example 2 again. We divide the input set  $\mathcal{X}_{in}$  into  $4 \times 10^4$  small intervals of equal size and verify whether the NN is a homeomorphism with respect to each of them based on the use of interval arithmetic to determine the determinant of the corresponding Jacobian matrix  $\frac{\partial \mathbf{y}}{\partial \mathbf{x}_0} = \frac{\partial \mathbf{N}(\mathbf{x}_0)}{\partial \mathbf{x}_0}$ . The blue region in Fig. 3(b) is the set of intervals, which features the NN with the homeomorphism property. The number of these intervals is 31473. For simplicity, we denote these intervals by  $\mathcal{A}$ .



**Fig. 3.** Illustrations on Example 2, 3 and 4

It is interesting to find that the safety verification in the sense of Definition 1 can be addressed by performing reachability analysis on a subset of the input set  $\mathcal{X}_{in}$ . This subset is obtained via removing subsets in the input set  $\mathcal{X}_{in}$ , which features the NN with the homeomorphism property.

**Theorem 2.** *Let  $\mathcal{A} \subseteq \mathcal{X}_{in}$  and  $\mathcal{A} \cap \partial\mathcal{X}_{in} = \emptyset$ , and  $\mathbf{N}(\cdot) : \mathcal{A} \rightarrow \mathcal{R}(\mathcal{A})$  be a homeomorphism with respect to the input set  $\mathcal{A}$ . Then, if the output reachable set of the closure of the set  $\mathcal{X}_{in} \setminus \mathcal{A}$  is a subset of the safe set  $\mathcal{X}_s$ , i.e.,  $\mathcal{R}(\overline{\mathcal{X}_{in} \setminus \mathcal{A}}) \subseteq \mathcal{X}_s$ , the safety property that  $\forall \mathbf{x}_0 \in \mathcal{X}_{in}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s$  holds.*

*Proof.* Obviously, if  $\mathcal{R}(\mathcal{A}) \subseteq \mathcal{X}_s$  and  $\mathcal{R}(\overline{\mathcal{X}_{in} \setminus \mathcal{A}}) \subseteq \mathcal{X}_s$ , the safety property that  $\forall \mathbf{x}_0 \in \mathcal{X}_{in}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s$  holds.

According to Theorem 1, we have that if  $\mathcal{R}(\partial\mathcal{A}) \subseteq \mathcal{X}_{in}$ , the safety property that  $\forall \mathbf{x}_0 \in \mathcal{A}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s$  holds.

According to the condition that  $\mathcal{A} \subseteq \mathcal{X}_{in}$  and  $\mathcal{A} \cap \partial\mathcal{X}_{in} = \emptyset$ , we have that  $\mathcal{A} \subseteq \mathcal{X}_{in}^\circ$  and thus  $\partial\mathcal{A} \subseteq \overline{\mathcal{X}_{in} \setminus \mathcal{A}}$ . Therefore,  $\mathcal{R}(\overline{\mathcal{X}_{in} \setminus \mathcal{A}}) \subseteq \mathcal{X}_s$  implies that  $\forall \mathbf{x}_0 \in \mathcal{X}_{in}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s$ . The proof is completed.

---

**Algorithm 2** Safety Verification Framework for Non-Invertible NNs

---

**Input:** a non-invertible NN  $\mathbf{N}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ , an input set  $\mathcal{X}_{in}$  and a safe set  $\mathcal{X}_s$ .**Output:** **Safe** or **Unknown**.

- 1: determine a subset  $\mathcal{A}$  of the set  $\mathcal{X}_{in}$  such that  $\mathbf{N}(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a homeomorphism with respect to it;
  - 2: apply existing methods to compute an over-approximation  $\Omega(\overline{\mathcal{X}_{in} \setminus \mathcal{A}})$ ;
  - 3: **if**  $\Omega(\overline{\mathcal{X}_{in} \setminus \mathcal{A}}) \subseteq \mathcal{X}_s$  **then**
  - 4:     return **Safe**
  - 5: **else**
  - 6:     return **Unknown**
  - 7: **end if**
- 

Theorem 2 tells us that it is still possible to use a subset of the input set for addressing the problem in Definition 1, even if the given NN is not a homeomorphism with respect to  $\mathcal{X}_{in}$ . This is shown in Example 4.

*Example 4.* Consider the situation in Example 3 again. If the entire input set is used for computations, all of  $4 \times 10^4$  small intervals participate in calculations. However, Theorem 2 tells us that only 9071 intervals (i.e., subset  $\overline{\mathcal{X}_{in} \setminus \mathcal{A}}$ ) are needed, which is much smaller than  $4 \times 10^4$ . The computation results based on interval arithmetic are illustrated in Fig. 3(c). It is noting that 9071 intervals rather than  $8527 (= 4 \times 10^4 - 31473)$  intervals are used since some intervals, which have non-empty intersection with the boundary of the input set  $\mathcal{X}_{in}$  (since Theorem 2 requires  $\mathcal{A} \cap \partial\mathcal{X}_{in} = \emptyset$ ), should participate in calculations.

*Remark 2.* According to Theorem 2, we can also observe that the boundary of the output reachable set  $\mathcal{R}(\mathcal{X}_{in})$  is included in the output reachable set of the input set  $\overline{\mathcal{X}_{in} \setminus \mathcal{A}}$ , i.e.,  $\partial\mathcal{R}(\mathcal{X}_{in}) \subseteq \mathcal{R}(\overline{\mathcal{X}_{in} \setminus \mathcal{A}})$ . This can also be visualized in Fig. 3(c). Consequently, this observation may open new research directions of addressing various problems of NNs [12]. For instance, it may facilitate the generation of adversarial examples, which are inputs causing the NN to falsify the safety property, and the characterization of decision boundaries of NNs, which are a surface that separates data points belonging to different class labels.

Therefore, we arrive at an algorithm for safety verification of non-invertible NNs, which is formulated in Algorithm 2.

**Theorem 3 (Soundness).** *If Algorithm 2 returns **Safe**, the safety property that  $\forall \mathbf{x}_0 \in \mathcal{X}_{in}. \mathbf{N}(\mathbf{x}_0) \in \mathcal{X}_s$  holds.*

*Proof.* This conclusion can be assured by Theorem 2.

*Remark 3.* The set-boundary reachability method can also be applied to intermediate layers in a given NN, rather than just the input and output layers. Suppose that there exists a sub-NN  $\mathbf{N}'(\cdot) : \mathbb{R}^{n'} \rightarrow \mathbb{R}^{n'}$ , which maps the input of the  $l$ -th layer to the output of the  $k$ -th layer, in the given NNs, and its input set is  $\mathcal{X}'_{in}$  which is an over-approximation of the output reachable set of

the  $(l - 1)$ -th layer. If  $\mathbf{N}'(\cdot) : \mathbb{R}^{n'} \rightarrow \mathbb{R}^{n'}$  is a homeomorphism with respect to  $\mathcal{X}'_{in}$ , we can use  $\partial\mathcal{X}'_{in}$  to compute an over-approximation  $\Omega'(\partial\mathcal{X}'_{in})$  of the output reachable set  $\{\mathbf{y} \mid \mathbf{y} = \mathbf{N}'(\mathbf{x}_0), \mathbf{x}_0 \in \partial\mathcal{X}'_{in}\}$ ; otherwise, we can apply Theorem 2 and compute an over-approximation  $\Omega'(\overline{\mathcal{X}'_{in} \setminus \mathcal{A}})$  of the output reachable set  $\{\mathbf{y} \mid \mathbf{y} = \mathbf{N}'(\mathbf{x}_0), \mathbf{x}_0 \in \overline{\mathcal{X}'_{in} \setminus \mathcal{A}}\}$ . In case that the  $k$ -th layer is not the output layer of the NN, we need to construct a simply connected set, like convex polytope, zonotope or interval, to cover  $\Omega'(\partial\mathcal{X}'_{in})$  or  $\Omega'(\overline{\mathcal{X}'_{in} \setminus \mathcal{A}})$  for the subsequent layer-by-layer propagation. This set is an over-approximation of the output reachable set of the  $k$ -th layer, according to Lemma 1 in [46].

*Remark 4.* Any existing over-approximate reachability methods such as interval arithmetic- [41], zonotopes- [36], star sets [39] based methods, which are suitable for given NNs, can be used to compute the involved over-approximations, i.e.,  $\Omega(\partial\mathcal{X}_{in})$  and  $\Omega(\overline{\mathcal{X}_{in} \setminus \mathcal{A}})$ , in Algorithm 1 and 2.

## 5 Experiment

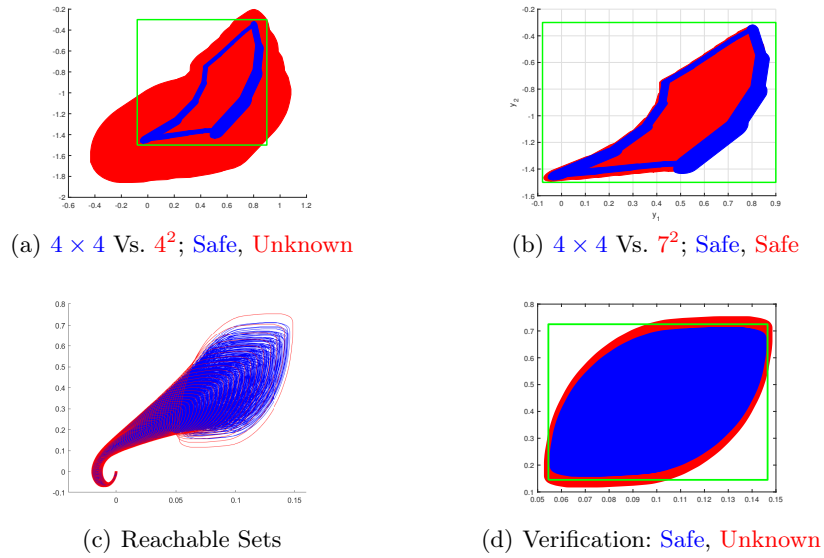
In this section, several examples of NNs are used to demonstrate the performance of the proposed set-boundary reachability method for safety verification. Experiments are conducted on invertible NNs and non-invertible ones respectively. Recall that the proposed set-boundary method is applicable for any reachability analysis algorithm based on set representation, resulting in tighter and verifiable over-approximations when existing approaches fail. Thus, we compare the set-boundary method versus the entire set one on some existing reachability tools in terms of efficiency.

**Experiment Setting.** All the experiments herein are run on MATLAB 2021a with Intel (R) Core (TM) i7-10750H CPU@2.60 GHz and RAM 16 GB. The codes and models are available from <https://github.com/laode2022/BoundaryNN>.

### 5.1 Experiments on Invertible NNs

In this subsection, we carry out some examples involving neural ODEs and invertible feedforward neural networks.

**Neural ODEs.** We experiment on two widely-used neural ODEs in [31], which are respectively a nonlinear 2-dimensional spiral [5] with the input set  $\mathcal{X}_{in} = [1.5, 2.5] \times [-0.5, 0.5]$  and the safe set  $\mathcal{X}_s = [-0.08, 0.9] \times [-1.5, -0.3]$  and a 12-dimensional controlled cartpole [14] with the input set  $\mathcal{X}_{in} = [-0.001, 0.001]^{12}$  and the safe set  $\mathcal{X}_s = [0.0545, 0.1465] \times [0.145, 0.725]$ . For simplicity, we respectively denote them  $N_1$  and  $N_2$ . Here, we take zonotopes as abstract domains and compare the output reachable sets computed by our set-boundary reachability method and the entire input set based method. The over-approximate reachability analysis is performed on the continuous reachability analyzer CORA toolbox [2]. When the time horizon is  $[0, 6]$  and the time step is 0.01, our set-boundary reachability method for  $N_1$  returns ‘Safe’ when the boundary of the input set is partitioned into 16 equal subsets, with the computation time being about 220.83

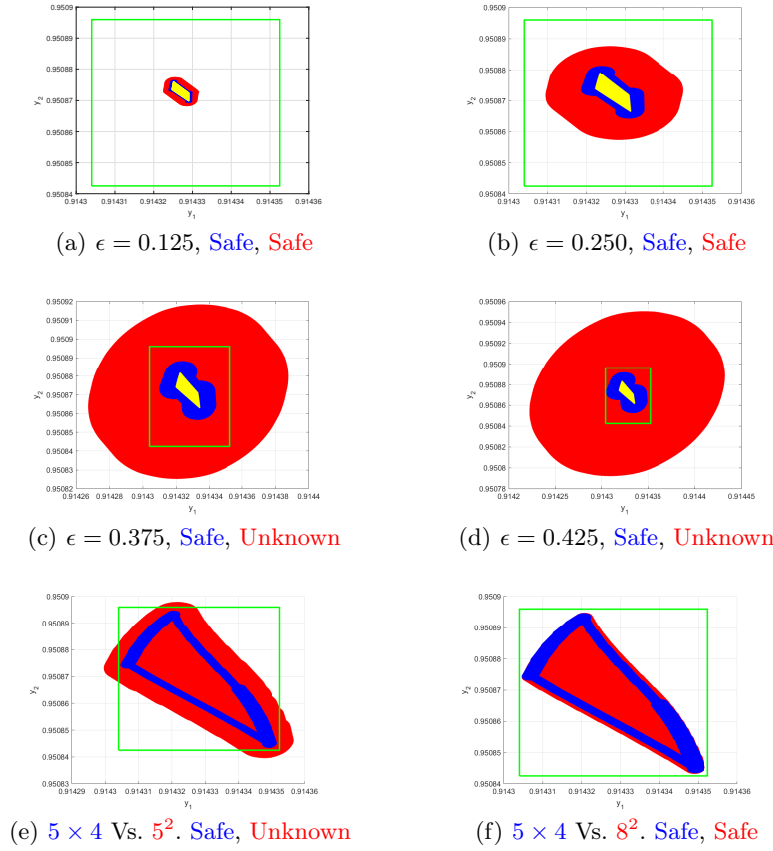


**Fig. 4.** Verification on  $\mathcal{N}_1, \mathcal{N}_2$ .  $\Omega(\partial\mathcal{X}_{in})$ ;  $\Omega(\mathcal{X}_{in})$ ;  $\partial\mathcal{X}_s$

seconds. However, the entire input set based method returns ‘Unknown’ when the input set is partitioned into 16 equal subsets. These color marks for safety verification results also apply to the experiments below. The safety property is verified until the entire input set is partitioned into 49 equal subsets. The corresponding computation time is 671.32 seconds. Consequently, the computation time from our set-boundary reachability method is reduced by 67.1%, compared to the entire input set based method. The computed output reachable sets for  $\mathcal{N}_1$  are displayed in Fig. 4(a) and 4(b). When the time horizon is  $[0.0, 1.1]$  and the time step is 0.01, the computed output reachable sets for  $\mathcal{N}_2$  are displayed in Fig. 4(c). Fig. 4(d) shows the reachable sets at the time instant  $t = 1.1$ .

**Feedforward Neural Networks.** Rather than considering neural ODEs, we instead take more general invertible NNs into account. The invertibility of NNs used here, i.e.,  $\mathcal{N}_3$  and  $\mathcal{N}_4$ , are assured by their Jacobian determinant not being zero. The NN  $\mathcal{N}_3$  is fully connected with Sigmoid activation functions, having an input/output layer with dimension 2 and 10 hidden layers with size 100. The NN  $\mathcal{N}_4$  is similar to  $\mathcal{N}_3$ , except that its input/output dimensions are 3.

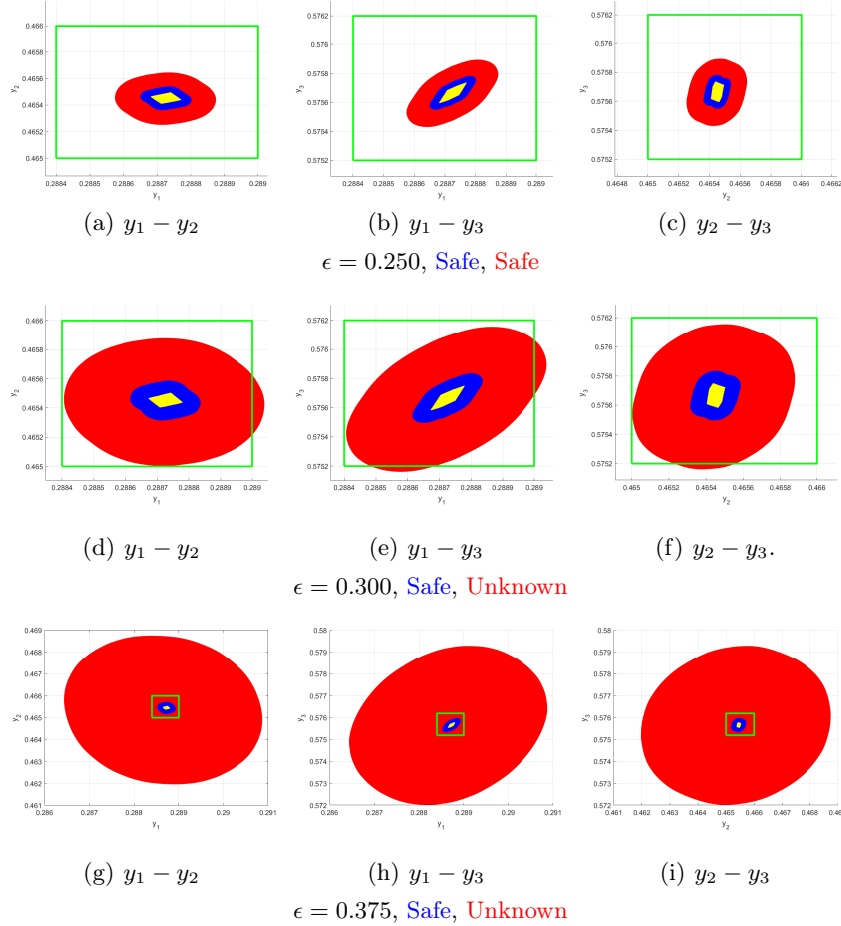
The results of safety verification of  $\mathcal{N}_3$  and  $\mathcal{N}_4$  are demonstrated in Fig. 5 and 6. Their safe sets  $\mathcal{X}_s$  are respectively  $[0.914304, 0.9143525] \times [0.9508425, 0.950896]$  and  $[0.2884, 0.289] \times [0.465, 0.466] \times [0.5752, 0.5762]$ , whose boundaries are shown in green color in Fig. 5 and 6. The input sets in Fig. 5 are  $[-0.125, 0.125]^2$ ,  $[-0.25, 0.25]^2$ ,  $[-0.375, 0.375]^2$ ,  $[-0.425, 0.425]^2$  and  $[-1.0, 1.0]^2$  (Fig. 5(e)-5(f).) respectively and those of Fig. 6 are  $[-0.25, 0.25]^3$ ,  $[-0.30, 0.30]^3$ ,  $[-0.375, 0.375]^3$ . The over-approximate reachability analysis is implemented using DeepZ [36],



**Fig. 5.** Safety verification on  $N_3$ .  $\Omega(\partial\mathcal{X}_{in})$ ;  $\Omega(\mathcal{X}_{in})$ ;  $\partial\mathcal{X}_s$

which is a tool for safety verification of large feed-forward, convolutional, and residual networks via propagating zonotopes through networks.

The output reachable sets from our set-boundary reachability method and the entire set based method are displayed in blue and red in Fig. 5 and 6, respectively. Further, we also show the exact output reachable sets estimated via the Monte-Carlo simulation method in Fig. 5 and 6, which correspond to the yellow regions. The visualized results show that the set-boundary reachability method can generate tighter output reachable sets than the entire set based method. As a result, our set-boundary reachability method can verify the safety properties successfully for all cases. In contrast, the entire set based method fails for large input sets, as shown in Fig. 5(c), 5(d), 6(d)-6(f) and 6(g)-6(i), since the computed output reachable sets are not included in safe sets. Furthermore, when the safety property cannot be verified with the input set  $[-1.0, 1.0]^2$ , we impose uniform partition operator on both the entire input set and its boundary



**Fig. 6.** Safety verification on  $N_4$ .  $\Omega(\partial\mathcal{X}_{in})$ ;  $\Omega(\mathcal{X}_{in})$ ;  $\partial\mathcal{X}_s$

for verifying the safety property. When the boundary is divided into 20 equal subsets, the safety verification can be verified using our set-boundary reachability method (Fig. 5(e)) with the computation time of 0.0624 seconds. However, when the entire input set is used, it should be partitioned into 64 equal subsets (Fig. 5(f)) and the computation time for verification is 0.7405 seconds. Consequently, the computation time from our set-boundary reachability method is reduced by 91.6%, as opposed to the entire input set based method.

## 5.2 Experiments on Non-invertible NNs

When homeomorphisms cannot be assured with respect to given input regions, our method is also able to facilitate the extraction of subsets from the input

region for safety verification, as done in Algorithm 2. In this subsection, we experiment on a non-invertible NN  $\mathcal{N}_5$ , which shares the same structure with  $\mathcal{N}_3$ . The input set  $\mathcal{X}_{in}$  and safe set  $\mathcal{X}_s$  are  $[-0.5, 0.5]^2$  and  $[0.06546, 0.06555] \times [0.07828, 0.07832]$ , respectively. Then, based on the tool DeepZ, we follow the computational procedure in Algorithm 2 for verifying the safety property. The computed output reachable sets and the verification result are shown in Fig. 7. The subset  $\mathcal{A} = [-0.45, 0.3]^2$  rendering the NN homeomorphic is illustrated in Fig. 7(a), which is the orange region, and the subset  $\overline{\mathcal{X}_{in} \setminus \mathcal{A}}$  is the blue region in Fig. 7(a). It can be seen that the subset  $\mathcal{A}$  extracted by set-boundary analysis for safety verification covers only 56.25% of the initial input set. The output reachable set computed from the entire input set is also displayed in Fig.7(b), which correspond to the red region. Moreover, the boundary of the safe region and the output reachable set estimated via the Monte-Carlo simulation method are shown in green and yellow in Fig.7(b), respectively. It can be observed that our set-boundary reachability method facilitates the generation of a tighter output reachable set, which is included in the safe set  $\mathcal{X}_s$ . Thus, the safety property is ensured by our set-boundary reachability method. However, the entire set based method fails. Moreover, the computation time of safe verification on  $\mathcal{N}_5$  based on our set-boundary reachability method is 0.0459 seconds, while the verification time from the entire set based method takes 0.0522 seconds with 4 equal subsets.

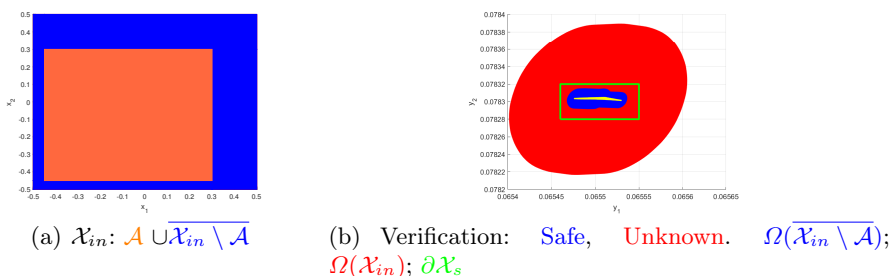


Fig. 7. Safety verification on  $\mathcal{N}_5$

## 6 Conclusion

In this paper we proposed a set-boundary reachability method to verify safety property of NNs. Different from existing works on developing computational techniques for output reachable sets estimation of NNs, the set-boundary reachability method analyzed the reachability from the topology point of view. Based on homeomorphism property, this analysis took a careful inspection on what happens at boundaries of input sets, and uncovered that the homeomorphism property facilitates the reduction of computational burdens on safety verifica-

tion of NNs. Several examples demonstrated the performance of the proposed method.

There are a lot of works remaining to be done in order to render the proposed approach more practical. For instance, in this paper a homeomorphism is determined via the use of interval arithmetic to calculate the determinant of the Jacobian matrix. Such an interval estimation is coarse, which affects the determination of a homeomorphism and thus the extraction of the small subset for reachability computations. In the future we will develop more efficient and accurate methods for calculating Jacobian matrices. Besides, the homeomorphism property may be strict. Different from homeomorphisms, open maps, mapping open sets to open sets [33], can also ensure that the output reachable set's boundary corresponds to the input's boundary. Moreover, the open mapping condition is weaker than the one for a homeomorphism. Consequently, in future work we would exploit the open mapping property to facilitate reachability computations for safety verification.

## References

1. Akintunde, M.E., Kevorchian, A., Lomuscio, A., Pirovano, E.: Verification of rnn-based neural agent-environment systems. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 6006–6013 (2019)
2. Althoff, M.: An introduction to cora 2015. In: Proc. of the workshop on applied verification for continuous and hybrid systems. pp. 120–151 (2015)
3. Ardizzone, L., Kruse, J., Rother, C., Köthe, U.: Analyzing inverse problems with invertible neural networks. In: International Conference on Learning Representations (2018)
4. Behrmann, J., Grathwohl, W., Chen, R.T., Duvenaud, D., Jacobsen, J.H.: Invertible residual networks. In: International Conference on Machine Learning. pp. 573–582. PMLR (2019)
5. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. *Advances in neural information processing systems* **31** (2018)
6. Cousot, P., Cousot, R.: Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages. pp. 238–252 (1977)
7. Dahnert, M., Hou, J., Nießner, M., Dai, A.: Panoptic 3d scene reconstruction from a single rgb image. *Advances in Neural Information Processing Systems* **34** (2021)
8. Dupont, E., Doucet, A., Teh, Y.W.: Augmented neural odes. *Advances in Neural Information Processing Systems* **32** (2019)
9. Dutta, S., Jha, S., Sanakaranarayanan, S., Tiwari, A.: Output range analysis for deep neural networks. *arXiv preprint arXiv:1709.09130* (2017)
10. Ehlers, R.: Formal verification of piece-wise linear feed-forward neural networks. In: International Symposium on Automated Technology for Verification and Analysis. pp. 269–286. Springer (2017)
11. Gehr, T., Mirman, M., Drachler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.: Ai2: Safety and robustness certification of neural networks with abstract interpretation. In: 2018 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2018)



12. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 3681–3688 (2019)
13. Gomez, A.N., Ren, M., Urtasun, R., Grosse, R.B.: The reversible residual network: Backpropagation without storing activations. *Advances in neural information processing systems* **30** (2017)
14. Gruenbacher, S., Cyranka, J., Lechner, M., Islam, M.A., Smolka, S.A., Grosu, R.: Lagrangian reachtubes: The next generation. In: 2020 59th IEEE Conference on Decision and Control (CDC). pp. 1556–1563. IEEE (2020)
15. Gruenbacher, S., Lechner, M., Hasani, R., Rus, D., Henzinger, T.A., Smolka, S., Grosu, R.: Gotube: Scalable stochastic verification of continuous-depth models. arXiv preprint arXiv:2107.08467 (2021)
16. Grunbacher, S., Hasani, R., Lechner, M., Cyranka, J., Smolka, S.A., Grosu, R.: On the verification of neural odes with stochastic guarantees. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11525–11535 (2021)
17. Hasani, R., Lechner, M., Amini, A., Rus, D., Grosu, R.: A natural lottery ticket winner: Reinforcement learning with ordinary neural circuits. In: International Conference on Machine Learning. pp. 4082–4093. PMLR (2020)
18. Huang, C., Fan, J., Li, W., Chen, X., Zhu, Q.: Reachnn: Reachability analysis of neural-network controlled systems. *ACM Transactions on Embedded Computing Systems (TECS)* **18**(5s), 1–22 (2019)
19. Huang, X., Kwiatkowska, M., Wang, S., Wu, M.: Safety verification of deep neural networks. In: International conference on computer aided verification. pp. 3–29. Springer (2017)
20. Ivanov, R., Carpenter, T.J., Weimer, J., Alur, R., Pappas, G.J., Lee, I.: Verifying the safety of autonomous systems with neural network controllers. *ACM Transactions on Embedded Computing Systems (TECS)* **20**(1), 1–26 (2020)
21. Jacobsen, J.H., Smeulders, A., Oyallon, E.: i-revnet: Deep invertible networks. arXiv preprint arXiv:1802.07088 (2018)
22. Joshi, K.D.: Introduction to general topology. New Age International (1983)
23. Karch, T., Teodorescu, L., Hofmann, K., Moulin-Frier, C., Oudeyer, P.Y.: Grounding spatio-temporal language with transformers. arXiv preprint arXiv:2106.08858 (2021)
24. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: International conference on computer aided verification. pp. 97–117. Springer (2017)
25. Krantz, S.G., Parks, H.R.: The implicit function theorem: history, theory, and applications. Springer Science & Business Media (2002)
26. Lechner, M., Hasani, R., Amini, A., Henzinger, T.A., Rus, D., Grosu, R.: Neural circuit policies enabling auditable autonomy. *Nature Machine Intelligence* **2**(10), 642–652 (2020)
27. Liu, C., Arnon, T., Lazarus, C., Strong, C., Barrett, C., Kochenderfer, M.J., et al.: Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization* **4**(3-4), 244–404 (2021)
28. Liu, W., Song, F., Zhang, T., Wang, J.: Verifying relu neural networks from a model checking perspective. *Journal of Computer Science and Technology* **35**, 1365–1381 (11 2020). <https://doi.org/10.1007/s11390-020-0546-7>
29. Lomuscio, A., Maganti, L.: An approach to reachability analysis for feed-forward relu neural networks. arXiv preprint arXiv:1706.07351 (2017)

30. Lopez, D.M., Musau, P., Hamilton, N., Johnson, T.T.: Reachability analysis of a general class of neural ordinary differential equations. arXiv preprint arXiv:2207.06531 (2022)
31. Manzananas Lopez, D., Musau, P., Hamilton, N., Johnson, T.T.: Reachability analysis of a general class of neural ordinary differential equations. arXiv e-prints pp. arXiv-2207 (2022)
32. Massey, W.S.: A basic course in algebraic topology, vol. 127. Springer (2019)
33. Mendelson, B.: Introduction to topology. Courier Corporation (1990)
34. Naitzat, G., Zhitnikov, A., Lim, L.H.: Topology of deep neural networks. *J. Mach. Learn. Res.* **21**(184), 1–40 (2020)
35. Pulina, L., Tacchella, A.: An abstraction-refinement approach to verification of artificial neural networks. In: International Conference on Computer Aided Verification. pp. 243–257. Springer (2010)
36. Singh, G., Gehr, T., Mirman, M., Püschel, M., Vechev, M.: Fast and effective robustness certification. *Advances in neural information processing systems* **31** (2018)
37. Singh, G., Gehr, T., Püschel, M., Vechev, M.: An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages* **3**(POPL), 1–30 (2019)
38. Tian, Y., Yang, W., Wang, J.: Image fusion using a multi-level image decomposition and fusion method. *Applied Optics* **60**(24), 7466–7479 (2021)
39. Tran, H.D., Manzananas Lopez, D., Musau, P., Yang, X., Nguyen, L.V., Xiang, W., Johnson, T.T.: Star-based reachability analysis of deep neural networks. In: International symposium on formal methods. pp. 670–686. Springer (2019)
40. Tran, H.D., Musau, P., Lopez, D.M., Yang, X., Nguyen, L.V., Xiang, W., Johnson, T.T.: Parallelizable reachability analysis algorithms for feed-forward neural networks. In: 2019 IEEE/ACM 7th International Conference on Formal Methods in Software Engineering (FormaliSE). pp. 51–60. IEEE (2019)
41. Wang, S., Pei, K., Whitehouse, J., Yang, J., Jana, S.: Efficient formal safety analysis of neural networks. *Advances in Neural Information Processing Systems* **31** (2018)
42. Xiang, W., Johnson, T.T.: Reachability analysis and safety verification for neural network control systems. arXiv preprint arXiv:1805.09944 (2018)
43. Xiang, W., Tran, H.D., Johnson, T.T.: Reachable set computation and safety verification for neural networks with relu activations. arXiv preprint arXiv:1712.08163 (2017)
44. Xiang, W., Tran, H.D., Johnson, T.T.: Output reachable set estimation and verification for multilayer neural networks. *IEEE transactions on neural networks and learning systems* **29**(11), 5777–5783 (2018)
45. Xue, B., Easwaran, A., Cho, N.J., Fränzle, M.: Reach-avoid verification for nonlinear systems based on boundary analysis. *IEEE Transactions on Automatic Control* **62**(7), 3518–3523 (2016)
46. Xue, B., She, Z., Easwaran, A.: Under-approximating backward reachable sets by polytopes. In: International Conference on Computer Aided Verification. pp. 457–476. Springer (2016)
47. Xue, B., Wang, Q., Feng, S., Zhan, N.: Over-and underapproximating reach sets for perturbed delay differential equations. *IEEE Transactions on Automatic Control* **66**(1), 283–290 (2020)
48. Yang, P., Li, R., Li, J., Huang, C.C., Wang, J., Sun, J., Xue, B., Zhang, L.: Improving neural network verification through spurious region guided refinement. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems. pp. 389–408. Springer (2021)

49. Yuan, W., Neubig, G., Liu, P.: Bartscore: Evaluating generated text as text generation. arXiv preprint arXiv:2106.11520 (2021)