Artificial Intelligence Review (2024) 57:120
https://doi.org/10.1007/s10462-024-10749-8

A stable mapping of nmODE

Haiying Luo[1] · Tao He[1] · Zhang Yi[1]

Accepted: 14 March 2024

Abstract
Adversarial attacks pose significant challenges to the reliability and performance of neural networks. Despite the development of several defense mechanisms targeting various types of adversarial perturbations, only a few manage to strike a balance between theoretical soundness and practical efficacy. *nmODE* (neural memory ordinary differential equation) is a recently proposed model with several intriguing properties. By delving into the rare attribute of global attractors inherent in *nmODE*, this paper unveils its stable mapping, thereby conferring certified defense capabilities upon it. Moreover, a novel quantitative approach is proposed, establishing a mathematical link between perturbations and *nmODE*'s defense proficiency. Additionally, a training technique termed as *nmODE*$^+$ is put forward, enhancing the defense capability of *nmODE* without imposing additional training burdens. Extensive experiments demonstrate *nmODE*'s resilience to various perturbations, showcasing its seamless integration with neural networks and existing defense mechanisms. These findings offer valuable insights into leveraging differential equations for robust neural network security.

Keywords Adversarial defense · Ordinary differential equations · Neural ordinary differential equations · Neural networks

1 Introduction

With the development of deep learning, neural networks have demonstrated exceptional performance in computer vision and natural language processing tasks. However, studies show that neural networks are vulnerable to kinds of attacks. Szegedy et al. (2013) introduced the concept of adversarial samples, denoting modified data that can induce neural network models to produce incorrect predictions, with these modifications being

✉ Zhang Yi
zhangyi@scu.edu.cn

Haiying Luo
luohaiying_cs@stu.scu.edu.cn

Tao He
tao_he@scu.edu.cn

[1] Intelligent Interdisciplinary Research Center and College of Computer Science, Sichuan University, Chengdu 610065, China

Published online: 15 April 2024

almost imperceptible to humans. By artificially crafting adversarial samples, even the state-of-the-art classifiers can give wrong results with high confidence (Goodfellow et al. 2014). This problem has threatened the security of neural networks, attracting the attention of numerous researchers.

To solve this problem, kinds of defense methods have been explored, which can be mainly categorized into heuristic defenses and certified defenses. Heuristic defenses are effective in practice, with numerous studies focusing on it, represented by adversarial training (Goodfellow et al. 2014; Madry et al. 2017), defensive distillation (Papernot et al. 2016), and gradient masking (Gu and Rigazio 2014). However, heuristic defenses lack theoretical guarantees, raising concerns about the ability to resist future novel attacks. In contrast, certified defenses (Wong and Kolter 2018; Raghunathan et al. 2018; Weng et al. 2018; Lecuyer et al. 2019) have the theoretical guarantee for defense ability, offering both theoretical and practical effectiveness, which are even more worth exploring.

The field of neural networks defined by differential equations has been a highly active area of research in recent years. Utilizing differential equations, scholars have explored the behavior of neural networks as dynamic systems. For example, multilayer neural networks can be considered as the discretization of continuous dynamic systems (Weinan 2017), convolutional neural networks can be interpreted as a discrete form of nonlinear partial differential equations (Haber et al. 2018), and recurrent neural networks can be viewed as ODEs (ordinary differential equations) (Chang et al. 2019). Differential equations have provided us with deeper insights into neural network properties and potential applications.

In the theory of differential equations, attractors are used to describe the behavior of a system as it converges to a certain state during its evolution. Particularly, a global attractor can attract all initial conditions in the system, guiding to a stable state. Stability measures whether a system can return to its original equilibrium state after small disturbances, and a global attractor ensures such return. Given the stabilizing effect of global attractors, why not leverage global attractors for defense against adversarial attacks?

This is because the global attractor property is very rare, which normal dynamic systems do not possess. Fortunately, the recently proposed *nmODE* (neuron memory ordinary differential equation) (Yi 2023), a variation of neural ODE, possesses the property of global attractors. The global attractor represents the long-term behavior of the *nmODE*, and its stability is described and understood through stable mappings. Inspired by this rare property of *nmODE*, we find that *nmODE* possesses intrinsic stable mapping to defense against perturbations. In this paper, we further explore the stability of *nmODE*. The main contributions are summarized as follows:

- We propose a certified defense method leveraging stable mapping in *nmODE*, featuring inherent defense capability and mathematical provability, which enhances the security and reliability of machine learning models.
- A quantitative approach to assess *nmODE* defense ability has been proposed, establishing mathematical relationships between perturbations and defense capability. This offers valuable insights for future quantitative analysis of network defense ability.
- We propose a training method termed as $nmODE^{+}$ to enhance defense capabilities, building upon the theoretical underpinnings of stable mapping from *nmODE*, while incurring no additional training costs. This holds value for the training methods development aimed at defense ability enhancement.
- Extensive experiments demonstrate that *nmODE* can resist types of adversarial perturbations, and can be seamlessly integrated with neural networks and defense methods.

2 Related work

2.1 Certified defense

Wong and Kolter (2018) propose a method to train provably robust neural networks by optimizing convex outer bounds on the adversarial polytope, and the approach is guaranteed to detect all adversarial examples. Raghunathan et al. (2018) develop a new differentiable upper bound on the performance of two-layer networks when the adversarial input in l_∞ is assumed to be applied. Weng et al. (2018) develop two fast algorithms that can certify non-trivial lower bounds of minimum adversarial distortions for obtaining a tight and certified lower bound β_L on ReLU networks. Lecuyer et al. (2019) propose a novel and orthogonal approach for certified robustness against adversarial examples that is broadly applicable and scalable, and they also develop PixelDP, the first certified defense that scales effectively to large networks and datasets. Zhai et al. (2020) propose the MACER algorithm, which learns robust models without using adversarial training but performs better than all existing provable l2-defenses. Levine and Feizi (2020) introduce a certifiable defense against patch attacks that guarantees for a given image and patch attack size, no patch adversarial examples exist. Chiang et al. (2020) propose the first certified defense against patch attacks, and propose faster methods for its training. Zizzo et al. (2021) model an attacker who poisons the model to insert a weakness into the adversarial training such that the model displays apparent adversarial robustness, while the attacker can exploit the inserted weakness to bypass the adversarial training and force the model to misclassify adversarial examples. Cullen et al. (2022) demonstrate how these best-possible certificates can be improved upon by exploiting both the transitivity of certifications, and the geometry of the input space, giving rise to what has been called Geometrically Informed Certified Robustness.

Overall, although there have been numerous certified defense mechanisms, there are still some shortcomings that require further improvement. These include issues such as scalability when dealing with large-scale networks and datasets, as well as the adaptability to various deep neural network architectures.

2.2 ODE-based defense

Neural ODE has been proposed as a continuous approximation to the ResNet architecture. Recent studies have demonstrated that neural ODEs are intrinsically more robust against adversarial attacks compared to vanilla DNNs.

Yan et al. (2019) present an empirical study on the robustness of ODE-based networks, finding that they are more robust against both random Gaussian perturbations and L_∞ adversarial perturbations crafted by FGSM and PGD compared to conventional CNNs. Liu et al. (2020) introduce a provably stable architecture for neural ODEs that achieves non-trivial adversarial robustness under white-box adversarial attacks even when the network is trained naturally. Kang et al. (2021) propose a neural ODE with Lyapunov-stable equilibrium points for defending against adversarial attacks (SODEF). Inspired by the asymptotic stability of the general nonautonomous dynamical system, Li et al. (2022) propose to make each clean instance be the asymptotically stable equilibrium point of a slowly time-varying system to defend against adversarial attacks. Huang et al. (2022) present a framework called FI-ODE, using Lyapunov functions, barrier functions, and control policies for

certifiably robust forward invariance in neural ODEs. Arvinte et al. (2023) investigate the robustness of density estimation using the probability flow neural ODE model against gradient-based likelihood maximization attacks and the relation to sample complexity, where the compressed size of a sample is used as a measure of its complexity. Yang et al. (2023) present the B-NODE, incorporating barrier functions into the training process, which ensures that the system remains stable and does not deviate too far from the original trajectory and improves the robustness of neural ODEs against adversarial attacks.

Although some works propose theoretically grounded methods for enhancing the robustness of neural ODEs, such as stability analysis and Lyapunov functions, there is a need for further theoretical exploration. The theoretical foundation of these approaches could be strengthened to provide deeper insights into the mechanisms underlying the improved robustness and to ensure the reliability of the proposed methods across different scenarios.

3 Preliminary

3.1 Neural ODE mapping

The neural ODE mapping involves training a neural network to represent a continuous transformation of data, where the parameters of the network are learned such that they define the behavior of the ODE. This allows the model to capture complex temporal and spatial dependencies within data. Neural ODE mappings combine neural networks with the principles of differential equations, which are particularly useful for modeling dynamic and continuous processes (Kidger 2022). A general neural ODE is defined as

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

where W denotes learning parameters, x denotes external input, and y denotes ODE state.

For neural ODEs, a mapping is considered stable if, for any small change δ in the input, the corresponding change ϵ in the output remains bounded. A stable mapping refers to a mathematical function or transformation that exhibits certain desirable properties related to the behavior of nearby points when the input or domain is perturbed. We provide the definition of stable mapping as follows, where x and $y(t)$ represent the input and output, and $\bar{x}$ and $\bar{y}(t)$ represent the perturbed input and its corresponding perturbed output.

Definition 1 The neural ODE mapping $F : x \to y(t)$ defined by (1) is called stable, if given any $\epsilon > 0$, there exists a $\delta > 0$ such that $\|x - \bar{x}\| \leq \delta$ implies that $\|y(t) - \bar{y}(t)\| \leq \epsilon$ for all $t \geq 0$. (Fig. 1) Otherwise, the neural ODE mapping F is called unstable.

3.2 Global attractors

Global attractor is a concept in dynamical systems theory that describes the long-term behavior of a system. Global attractors represent the set of all possible states towards which a system tends to evolve over time, regardless of its initial conditions. In dynamical systems, global attractors represent the stable states towards which the system tends to

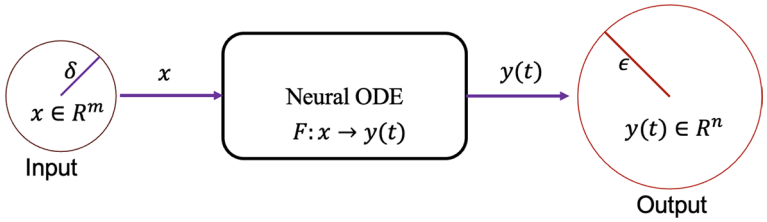


Fig. 1 The diagram of neural ODE's stable mapping

converge. Systems with global attractors are not easily perturbed from their stable states, as they tend to maintain their performance in the presence of perturbations.

Considering a dynamical system, whose evolution equation can be described by a differential equation:

$$\frac{dx}{dt} = F(x),$$

where x denotes the state vector of the system, t represents the time, and F denotes the function describing the dynamics. A subset of the state space denoted by $\mathcal{A}$ is a global attractor if it satisfies the following properties:

1. Invariance: For any $x(0) \in \mathcal{A}$, the solution satisfies $x(t) \in \mathcal{A}$ for all $t \geq 0$, that is:

$$x(0) \in \mathcal{A} \Rightarrow x(t) \in \mathcal{A}, \forall t.$$

$\mathcal{A}$ is invariant under the dynamics of the system. If the system starts from any initial condition in $\mathcal{A}$, it remains in $\mathcal{A}$ for all future time.

2. Attraction: For any $x(0) \notin \mathcal{A}$, the trajectory $x(t)$ converges to $\mathcal{A}$ as t goes to infinity, that is:

$$\lim_{t \to \infty} x(t) = \mathcal{A}.$$

$\mathcal{A}$ attracts all trajectories in the state space. For any initial condition not in $\mathcal{A}$, the trajectories converge towards $\mathcal{A}$ as time goes to infinity.

3.3 Perturbations

Perturbations, which refer to small changes in input data, can significantly impact the performance of neural networks. In particular, there are two types of perturbations: non-adversarial perturbations and adversarial perturbations.

Non-adversarial perturbations are common in real-world scenarios and occur naturally. Image processing technologies such as resizing, compression, and cropping can introduce perturbations to the original images (Zheng et al. 2016). Spatial transformation, such as rotation and shift, can also greatly reduce the performance of the state-of-the-art neural networks (Engstrom et al. 2017).

Adversarial perturbations are artificially crafted, and imperceptible to humans but have significant impacts on the performance of neural networks. According to Szegedy et al. (2013), adversarial perturbations exist because of data sampling problems, while Goodfellow et al. (2014) believed that they result from the accumulation of noise caused by high-dimensional linearity, and the excessive accumulation value leads to the classification error of neural networks.

Typically, the magnitude of the adversarial perturbations is commonly measured using $\mathcal{L}_p$ distance metric (Goodfellow et al. 2014; Carlini and Wagner 2017). For the real sample x and adversarial sample x', the $\mathcal{L}_p$ distance between them is given by:

$$||x - x'||_p = \left(\sum_{i=1}^{n} |x_i - x_i'|^p \right)^{\frac{1}{p}},$$

where p denotes a real number, and n represents the dimension of vector x. In the context of adversarial perturbations, $\mathcal{L}_2$ norm and $\mathcal{L}_\infty$ norm appear frequently. The $\mathcal{L}_2$ norm imposes a constraint on the overall perturbations, requiring the sum to be less than a certain threshold. The $\mathcal{L}_\infty$ norm restricts only the maximum value of the perturbations, deeming any perturbations within this maximum value as reasonable.

3.4 Adversarial attacks

Adversarial attacks can apply adversarial perturbations to neural networks. Research on adversarial attacks is crucial for enhancing the robustness and security of models, guarding against potential malicious manipulations and misdirection.

Adversarial attacks can be categorized into two categories: white-box attacks and black-box attacks. White-box attacks know the model architecture and can leverage the gradient information, while black-box attacks know nothing except for the input and the output. Usually, white-box attacks have better attack performance compared to black-box attacks, while black-box attacks are more practically significant, resulting from the difficulty for attackers to analyze the targeted model in real-world scenarios.

According to the attack frequency, adversarial attacks can be further classified into single-step attacks and iterative attacks. Single-step attacks involve only one attack iteration, characterized by fast execution but lower intensity, exemplified by FGSM (Goodfellow et al. 2014). Iterative attacks, represented by PGD (Madry et al. 2017), are improvements upon single-step attacks, involving multiple attack iterations following certain rules.

4 The stable mapping of nmODE

nmODE (Yi 2023) is an interesting neural network proposed recently, capturing the dynamical system behavior of memory neurons described by ODEs, which can be described by:

$$\begin{cases} \dot{y} = -\lambda y + \sin^2\left[y + \gamma\right] \\ \gamma = Wx + b \\ \lambda > 1 \end{cases} \tag{2}$$

In (2), y denotes the state of the network, λ represents the decay parameter, γ represents the perception input, W denotes the connection matrix, b denotes the bias, and x represents the external input.

nmODE is built upon the concept of columns in the neocortex, suggesting a unit of intelligence that may share a common algorithm across columns. *nmODE* presents several key differences and advantages compared to traditional neural ODE models. One of the main differences is the incorporation of a memory mechanism using global attractors in the network. It offers a unique perspective on how memory neurons can be integrated into neural network models, potentially enhancing their representation capabilities. Another significant difference is that *nmODE* is a decoupled system for memory neurons, making it particularly easy for mathematical analysis of its dynamics. This decoupling allows for independent solutions of one-dimensional ODEs for each memory neuron, which can be efficiently implemented using electric circuits to speed up network training. *nmODE* can be hierarchically stacked to create more complex networks, allowing for the construction of networks with stronger representation capabilities. This stacking feature provides flexibility in designing network architectures. Experimental results demonstrate that on the classification tasks, *nmODE* is comparable to state-of-the-art neuron ODEs (Chen et al. 2018; Dupont et al. 2019; Norcliffe et al. 2020).

The computing algorithm of *nmODE* is given as follows:

Algorithm 1 *nmODE* computing algorithm

Input: data x
Initialize: parameters W, b, $y(0)$, λ and $\bar{t}$
Compute $\gamma = Wx + b$
Solve $\dot{y} = -\lambda y + \sin^2\left[y + \gamma\right]$ to get $y(\bar{t})$
Output: $y(\bar{t})$

We find that *nmODE* has stable mapping and its defense capability is inherent. In this section, we will provide a stability theoretical guarantee for *nmODE*, propose a quantitative calculation method for the stability of *nmODE*, and propose the training method *nmODE*$^+$ aimed at enhancing the stability.

4.1 Stability theoretical guarantee

Theorem 1 *Suppose that $\lambda > 1$, then the mapping of nmODE (2) is stable. Moreover, given any $\epsilon > 0$, there exists*

$$\delta = \frac{(\lambda - 1) \cdot \epsilon}{\max_{1 \le i \le n} \left(\sum_{j=1}^{m} \left| w_{ij} \right| \right)} \tag{3}$$

such that $\|x - \bar{x}\| \le \delta$ implies that $\|y(t) - \bar{y}(t)\| \le \epsilon$ for all $t \ge 0$.

Proof Given any two external inputs x and $\bar{x}$, at time t, we have

$$\dot{y}_i(t) = -\lambda y_i(t) + \sin^2 \left[y_i(t) + \sum_{j=1}^{m} w_{ij} x_j \right]$$

and

$$\dot{\bar{y}}_i(t) = -\lambda \bar{y}_i(t) + \sin^2 \left[\bar{y}_i(t) + \sum_{j=1}^{m} w_{ij} \bar{x}_j \right].$$

It follows that

$$\begin{aligned} \frac{d\left[y_i(t) - \bar{y}_i(t)\right]}{dt} = & -\lambda \left[y_i(t) - \bar{y}_i(t) \right] \\ & + \sin^2 \left[y_i(t) + \sum_{j=1}^{m} w_{ij} x_j \right] \\ & - \sin^2 \left[\bar{y}_i(t) + \sum_{j=1}^{m} w_{ij} \bar{x}_j \right]. \end{aligned}$$

By using the Dini derivative, we have

$$\begin{aligned} D^+ \left| y_i(t) - \bar{y}_i(t) \right| \le & -(\lambda - 1) \cdot \left| y_i(t) - \bar{y}_i(t) \right| \\ & + \sum_{j=1}^{m} \left| w_{ij} \right| \cdot \left| x_j - \bar{x}_j \right| \end{aligned}$$

for $t \ge 0$. Let $y_i(0) = \bar{y}_i(0) = 0$, it gives that

$$\begin{aligned} \left| y_i(t) - \bar{y}_i(t) \right| \le & -e^{(\lambda-1)t} \cdot \left| y_i(0) - \bar{y}_i(0) \right| \\ & + \sum_{j=1}^{m} \int_0^t e^{(\lambda-1)(t-s)} \left| w_{ij} \right| \cdot \left| x_j - \bar{x}_j \right| ds \\ \le & \frac{1}{\lambda - 1} \cdot \sum_{j=1}^{m} \left| w_{ij} \right| \cdot \left| x_j - \bar{x}_j \right| \end{aligned}$$

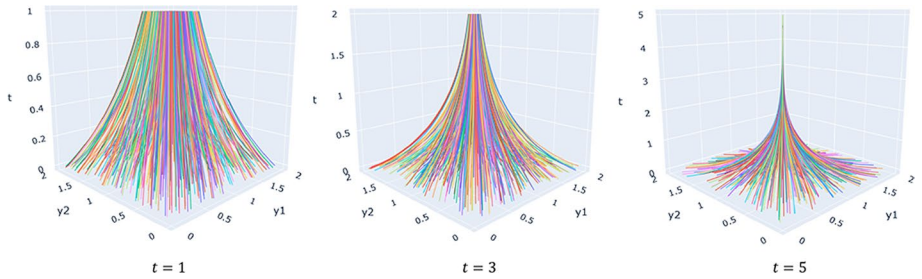


Fig. 2 The three-dimensional trajectory of *nmODE* over time t. We randomly simulated 1000 different sets of initial states $y \in \{(y_1, y_2) \mid y_1, y_2 \in [0, 2]\}$. The trajectories at $t = 1$, $t = 3$, and $t = 5$ are plotted separately. The global attractors ensure that *nmODE* ultimately reaches stability regardless of the initial conditions. At any time t, the stable mapping describes the local behavior of *nmODE*, providing an understanding of how *nmODE* converges to the global attractor within a local range

for $t \geq 0$. That is

$$\|y(t) - \bar{y}(t)\| \leq \frac{\max_{1 \leq i \leq n} \left(\sum_{j=1}^{m} \left| w_{ij} \right| \right)}{\lambda - 1} \cdot \|x - \bar{x}\|.$$

Given any $\epsilon > 0$, choose

$$\delta = \frac{(\lambda - 1) \cdot \epsilon}{\max_{1 \leq i \leq n} \left(\sum_{j=1}^{m} \left| w_{ij} \right| \right)},$$

then, if $\|x - \bar{x}\| \leq \delta$, it holds that $\|y(t) - \bar{y}(t)\| \leq \epsilon$ for all $t \geq 0$. The proof is complete.

□

Our theoretical analysis demonstrates the inherent property of stable mapping within *nmODE*. The stable mapping describes the local behavior of *nmODE*, providing an understanding of how *nmODE* converges to the global attractor within a local range. When the input undergoes slight modifications, the stable mapping ensures that the output remains largely unchanged, thereby endowing *nmODE* with certified defense capabilities. The trajectory of *nmODE* is shown in Fig. 2.

The similarity between *nmODE* stable mapping and the system identification methods with neural networks lies in their ability to defense perturbations. Compared to system identification methods, *nmODE* stable mapping offers theoretical guarantees, ensuring stability and provable protection, while system identification methods rely more on empirical and heuristic principles in their design and lack rigorous theoretical guarantees. This theoretical guarantee is crucial for addressing various security challenges, particularly in combating evolving attack methodologies. It provides a solid foundation for defense mechanisms, regardless of known or unknown attack scenarios.

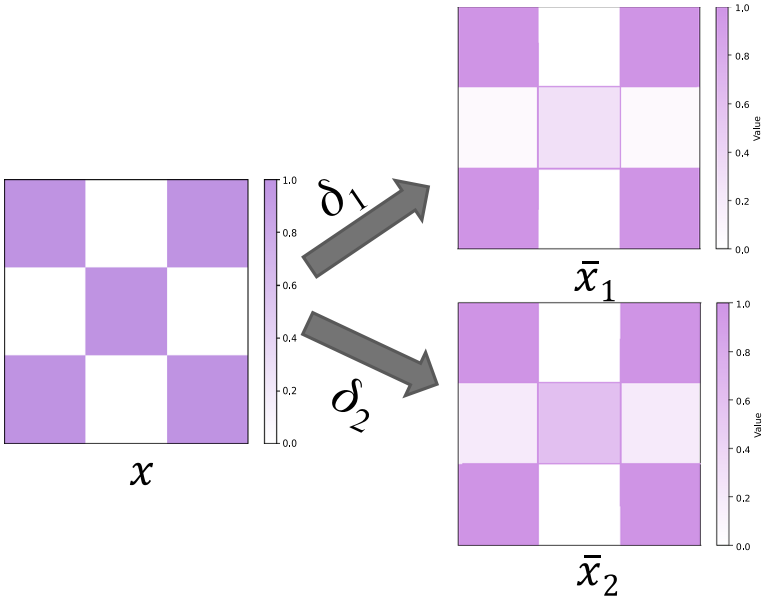


Fig. 3 An example used to illustrate Eq. (4). In this case, δ_1 and δ_2 represent perturbations applied to x, both with a magnitude of 0.8. These perturbations are used to generate the perturbed inputs $\bar{x}_1$ and $\bar{x}_2$

4.2 Quantitative method

Equation (3) elucidates the relationship between the small change δ in the input and the corresponding change ϵ in the output. ϵ characterizes the defense capability of *nmODE* against perturbations. Given a fixed value of δ, a smaller ϵ indicates a stronger defense capability of *nmODE*. By considering ϵ as the dependent variable, Eq. (3) can be rewritten as:

$$\epsilon = \frac{\tau}{\lambda - 1} \cdot \delta, \tag{4}$$

where $\tau = \max_{1 \le i \le n} \left(\sum_{j=1}^{m} \left| w_{ij} \right| \right)$.

Equation (4) paves the way for a quantitative analysis approach, facilitating a deeper understanding of the relationship between the imposed perturbation and the defensive capacity of *nmODE*. We utilize Fig. 3 to intuitively explain and validate this quantitative method. For an initial input $x = [1, 0, 1, 0, 1, 0, 1, 0, 1]^T$, we introduce two perturbations δ_1 and δ_2, where $\delta_1 = \delta_2 = 0.8$, resulting in the perturbed inputs $\bar{x}_1$ and $\bar{x}_2$. Concurrently, we set the corresponding connection matrix W_1 and W_2 as:

Table 1 The corresponding outputs y and $\bar{y}$ at different integration time t for *nmODE* and *P-nmODE*

nmODE						
t	y_1	$\bar{y}_1$	$\lvert y_1 - \bar{y}_1 \rvert$	y_2	$\bar{y_2}$	$\lvert y_2 - \bar{y}_2 \rvert$
0.6	0.0100	0.0135	0.0035	0.0157	0.0212	0.0055
1	0.0105	0.0143	0.0038	0.0167	0.0226	0.0059
5	0.0107	0.0145	0.0038	0.0169	0.0229	0.0060
P-nmODE						
t	y_1	$\bar{y_1}$	$\lvert y_1 - \bar{y}_1 \rvert$	y_2	$\bar{y_2}$	$\lvert y_2 - \bar{y}_2 \rvert$
0.6	0.2302	0.3193	0.0837	0.3767	0.5231	0.1464
1	2.3067	3.0890	0.7823	3.5706	4.7678	1.1972
5	1.15×10^9	1.54×10^9	0.39×10^9	1.78×10^9	2.35×10^9	0.57×10^9

$$W_1 = \underbrace{\left[\frac{0.4}{9}, \frac{0.4}{9}, \ldots\right]}_{1\times 9}$$
$$W_2 = \underbrace{\left[\frac{0.5}{9}, \frac{0.5}{9}, \ldots\right]}_{1\times 9}.$$

At this time, based on Eq. (4), we can calculate that $\tau = 0.5$ and $\epsilon = 0.1$. This suggests that by infusing a perturbation of 0.8 into x to yield $\bar{x}$, the difference between the outputs y and $\bar{y}$ when using *nmODE* should not surpass 0.1.

For comparison with *nmODE*, we used another ODE, named *P-nmODE*, which can be represented as:

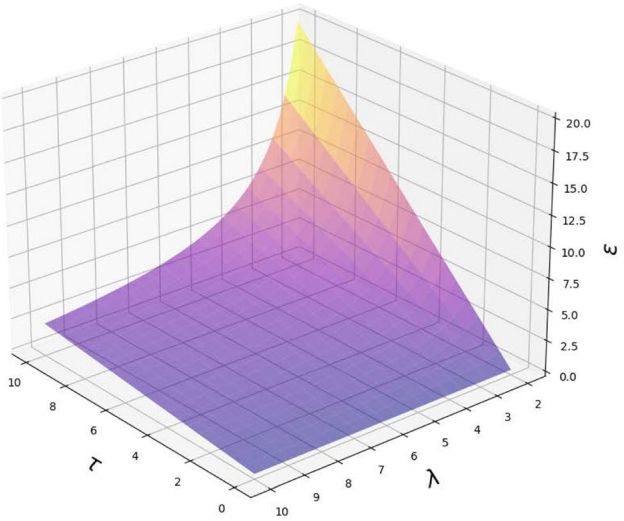
$$\begin{cases} \dot{y} = \lambda y + \sin^2\left[y + \gamma\right] \\ \gamma = Wx + b \\ \lambda > 1 \end{cases}$$

The key difference between *nmODE* and *P-nmODE* lies in the sign of the decay parameter λ. The results of the output y and $\bar{y}$ of *nmODE* and *P-nmODE* at different integration times t are shown in Table 1. It can be observed that for any $t \geq 0$, *nmODE* satisfies Eq. (4), while *P-nmODE* does not.

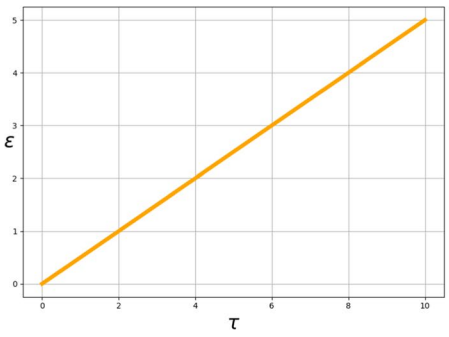
4.3 nmODE$^+$

From Eq. (4), it can be observed that under a fixed δ, ϵ is positively correlated with τ and negatively correlated with λ. The relationship diagram among ϵ, τ, and λ is depicted in Fig. 4.

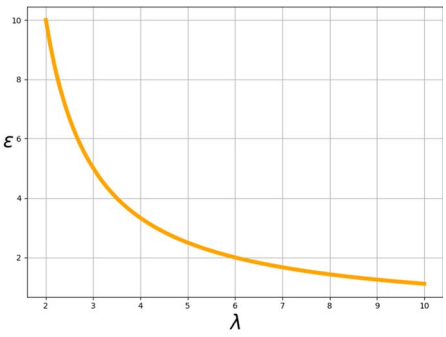
To enhance the defense capability of *nmODE*, it is desirable to minimize the value of τ and maximize the value of λ. Based on this finding, we propose a training method for *nmODE*, named *nmODE*$^+$, aimed at enhancing the stability of *nmODE*. Specifically,



(a) The relationship among ϵ, τ, and λ.



(b) The relationship between ϵ and τ.



(c) The relationship between ϵ and λ.

Fig. 4 The relationship diagram among ϵ, τ, and λ. Under a fixed δ, ϵ is positively correlated with τ and negatively correlated with λ

during the training process, it is recommended to set a relatively large value for the hyper-parameter λ, and constrain the weight of parameter W to be small to ensure a small τ.

To achieve the constraint of keeping the weight of parameter W small, we propose two implementation schemes: weight clipping, and adaptive parameter loss.

4.3.1 Weight clipping

Weight clipping involves setting a threshold to constrain weights within a specific range, preventing them from becoming excessively large. We have

$$\bar{W} = \text{clip}(W, -c, c), \tag{5}$$

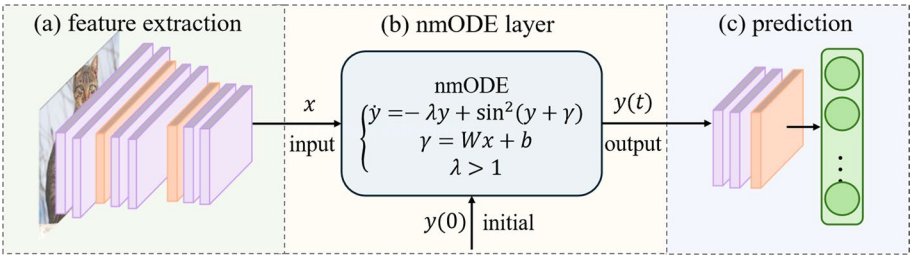


Fig. 5 The pipeline primarily consists of three stages: (a) feature extraction, during which the image is converted into tensor features via numerous layers; (b) the *nmODE* layer, which bolsters the model's robustness following the acquisition of the image representation; and (c) prediction, where the features refined by the *nmODE* layer are utilized for label prediction, thereby determining the specific classification

where W is the original weight, c is the threshold, and $\bar{W}$ is the clipped weight.

4.3.2 Adaptive parameter loss

Adaptive parameter loss is related to the magnitude of W, which varies according to the changes in weights. The loss function increases with the growth of weights, thereby constraining the magnitude of W. We have

$$\mathcal{L} = \mathcal{L}_{origin} + \eta \cdot \mathcal{L}_{stable}, \tag{6}$$

where

$$\mathcal{L}_{stable} = \max_{1 \le i \le n} \left(\sum_{j=1}^{m} \left| w_{ij} \right| \right), \tag{7}$$

and η is a hyperparameter used to adjust the penalty strength.

5 Experiment

The structure of this section unfolds as follows: initially, we introduce the dataset employed in the experiment, the methods chosen for comparison, and the specific experimental setup. Afterwards, we provide a comprehensive comparison of *nmODE*, highlighting its strengths and advantages in comparison with the currently predominant methods. Finally, we analyze the compatibility of *nmODE*, and conduct the ablation study. The pipeline of the method is depicted in Fig. 5.

5.1 Experimental setup

5.1.1 Datasets

- **MNIST.** MNIST (LeCun 1998) contains handwritten digits from 0 to 9, holding 60, 000 images in the training set and 10, 000 images in the testing set. The size of each image is 28×28.

- **Fashion-MNIST.** Fashion-MNIST (Xiao et al. 2017) consists of 60, 000 training images and 10, 000 testing images, each with a resolution of 28×28 pixels. The images represent 10 different fashion categories, including items like T-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots.
- **CIFAR-10.** CIFAR-10 (Krizhevsky 2009) contains 50, 000 training samples and 10, 000 testing samples, 10 distinct categories of 32×32 color images, encompassing airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks.

5.1.2 Attack methods

- **FGSM** FGSM (Goodfellow et al. 2014) is a simple and fast white-box attack for generating adversarial examples. It works by taking the gradient of the loss function with respect to the input data, and then perturbing the input data in the direction of the gradient sign. The amount of perturbation is controlled by a hyperparameter ϵ, which determines the maximum allowable size of the perturbation.
- **PGD** PGD (Madry et al. 2017) is a commonly used white-box attack for generating adversarial examples. It iteratively perturbs an input example in the direction of the gradient of the loss function with respect to the input, while projecting the perturbed example back onto a specified norm ball to ensure that the perturbation is not too large. By repeating this process multiple times, PGD can find adversarial examples that are close to the original example.
- **AutoPGD** AutoPGD (Croce and Hein 2020) is an adaptive adversarial attack method that automatically adjusts the step size and number of iterations to find the optimal adversarial examples within a given computational budget. Compared to PGD attack, AutoPGD can find more optimal adversarial examples faster and is more robust.
- **Square Attack** Square Attack (Andriushchenko et al. 2020) is a query-efficient black-box attack that generates adversarial examples by modifying square-shaped regions of the input image. The method is based on a randomized search scheme that explores the input space efficiently, which outperforms previous state-of-the-art methods in terms of success rate and query efficiency.

5.1.3 Implementation details

During the training and testing phase, Runge–Kutta of order 5 of ormand-Prince-Shampine ODE solver is employed, which is an adaptive-step ODE solver. The relative tolerance $rtol = 10^{-3}$ and absolute tolerance $atol = 10^{-3}$ correspond to the tolerances for accepting an adaptive step. We use torchdiffeq (Chen et al. 2018) to implement the ODE solver.

On MNIST and Fashion-MNIST, the original 28×28 pixel images are vectorized into 784-dimensional vectors. We use 2 fully connected layers, whose functions are feature extraction and classification prediction respectively. The connection matrices are $W^1_{2048\times784}$ and $W^2_{10\times2048}$. We insert *nmODE* between two fully connected layers. The model undergoes training on clean data utilizing cross-entropy loss, trained for 100 epochs with $\lambda = 3$, $\bar{t} = 5\text{e}{-2}$ and $\bar{t} = 1$ respectively.

On CIFAR-10, a pre-trained ResNet-18 model is utilized as the feature extractor,[1] the output of which is provided to *nmODE* as the input (512 dimensions). The model

[1] https://github.com/kuangliu/pytorch-cifar.

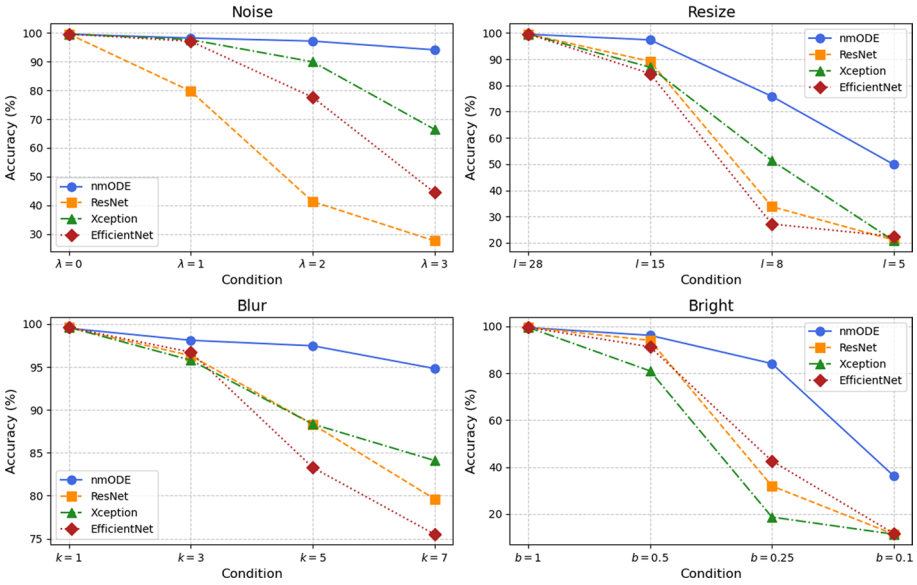


Fig. 6 Accuracy(%) of *nmODE* on Gaussian noise perturbations, resize operation, Gaussian blur operation, and brightness change compared to CNNs. λ denotes the intensity of Gaussian noise. l denotes the resized size. k denotes the kernel size of Gaussian blur. b denotes the intensity of brightness

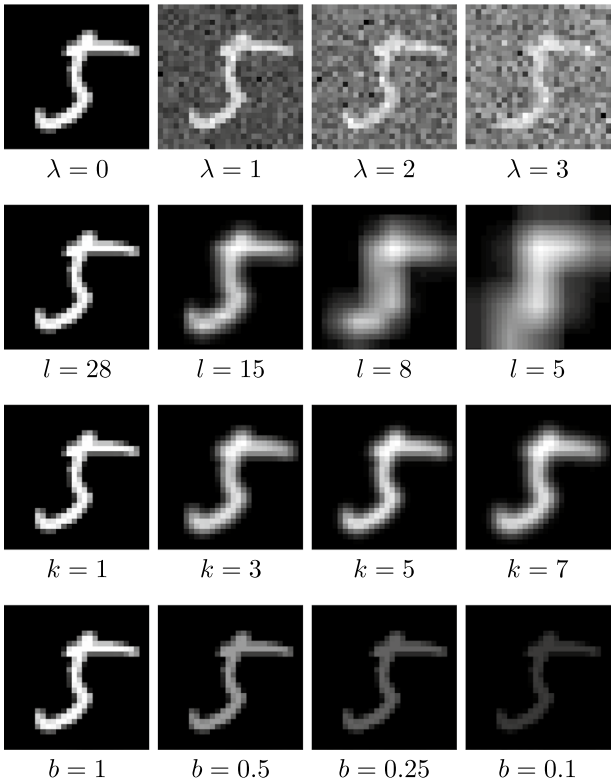


Fig. 7 Non-adversarial perturbations on the MNIST dataset crafted by Gaussian noise, resize operation, Gaussian blur, and brightness change. λ denotes the intensity of Gaussian noise. l denotes the resized size. k denotes the kernel size of Gaussian blur. b denotes the intensity of brightness

Table 2 Accuracy (%) of *nmODE* under $\mathcal{L}_\infty$ attacks compared to vanilla NODE (Chen et al. 2018), NODE trained with data augmentation (AT-NODE), ODE-TRADES (Zhang et al. 2019), TisODE (Yan et al. 2019), and B-NODE (Yang et al. 2023) on the MNIST dataset

Attack	Method	δ		
		0.01	0.03	0.05
PGD	Vanilla NODE	86.14 ± 1.33[a]	71.04 ± 2.01[a]	47.40 ± 2.11[a]
	AT-NODE	91.76 ± 1.16[a]	85.80 ± 1.94[a]	78.38 ± 1.67[a]
	ODE-TRADES	91.46 ± 1.07[a]	85.08 ± 1.61[a]	77.32 ± 1.89[a]
	TisODE	92.67 ± 1.28[a]	87.48 ± 1.42[a]	81.28 ± 1.28[a]
	B-NODE	92.62 ± 0.87[a]	89.16 ± 1.25[a]	83.18 ± 1.75[a]
	nmODE (ours)	$\mathbf{94.92 \pm 1.12}$	$\mathbf{93.02 \pm 1.53}$	$\mathbf{90.23 \pm 1.62}$
AutoPGD	Vanilla NODE	84.41 ± 1.11[a]	68.99 ± 1.71[a]	46.31 ± 2.51[a]
	AT-NODE	91.34 ± 1.51[a]	87.60 ± 2.67[a]	78.35 ± 1.87[a]
	ODE-TRADES	92.19 ± 1.92[a]	85.31 ± 2.41[a]	76.89 ± 3.85[a]
	TisODE	91.12 ± 1.27[a]	87.02 ± 2.21[a]	79.48 ± 2.35[a]
	B-NODE	93.45 ± 1.42[a]	88.95 ± 2.09[a]	82.65 ± 2.58[a]
	nmODE (ours)	$\mathbf{97.92 \pm 0.91}$	$\mathbf{94.16 \pm 1.46}$	$\mathbf{85.13 \pm 2.12}$
Square Attack	Vanilla NODE	86.55 ± 2.65[a]	72.50 ± 5.85[a]	51.58 ± 4.05[a]
	AT-NODE	92.65 ± 1.72[a]	89.50 ± 3.84[a]	81.13 ± 5.44[a]
	ODE-TRADES	92.06 ± 3.10[a]	90.08 ± 2.73[a]	82.41 ± 2.96[a]
	TisODE	94.10 ± 0.97[a]	87.60 ± 1.69[a]	83.24 ± 2.55[a]
	B-NODE	94.49 ± 2.55[a]	91.52 ± 3.49[a]	85.49 ± 2.50[a]
	nmODE (ours)	$\mathbf{98.35 \pm 1.23}$	$\mathbf{97.82 \pm 2.73}$	$\mathbf{95.77 \pm 2.69}$

Results that surpass all competing methods are bold

[a]Results taken from Yang et al. (2023)

undergoes training on clean data utilizing cross-entropy loss, trained for 1000 epochs with $\bar{t} = 5$ and $\lambda = 3$.

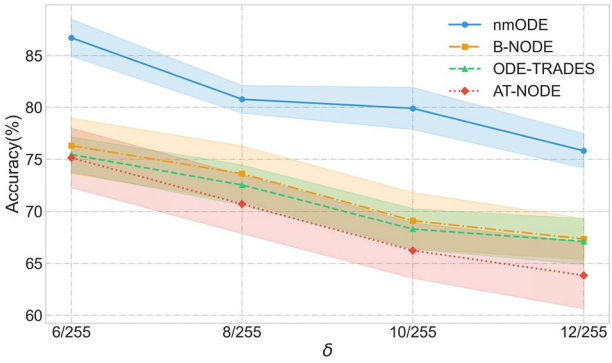
For the adversarial attacks, we use PGD, AutoPGD, and Square Attack in the experiment. For PGD, we apply the approach with 40 iterations and a flexible attack radius (50 iterations for the CIFAR-10 experiment). In the case of AutoPGD attack, we utilize an $\mathcal{L}_\infty$ norm attack along with a cross-entropy loss function and an update step size of 0.75. As for the Square Attack, we adhere to the procedure with $\mathcal{L}_\infty$ norm, which is generated through 5000 queries and incorporates a margin loss function.

We conduct all experiments using Pytorch 1.13.1 with Python 3.8.6, on an Ubuntu server 18.04.5 LTS with an RTX 3090 (24GB) GPU using CUDA 12.0.

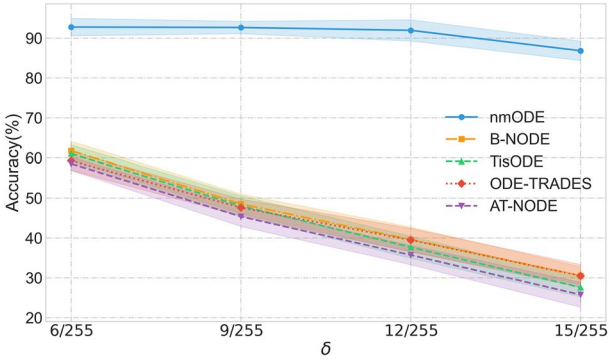
5.2 Experimental comparison

5.2.1 Non-adversarial robustness evaluation

We explore the naturally occurring perturbations defense ability of *nmODE*. We select popular CNN architectures for comparison, including ResNet (He et al. 2016), Xception (Chollet 2017), and EfficientNet (Tan and Le 2019). To ensure the fairness of the experiment, all models adopt the same training method and are trained on the original MNIST



(a) Fashion-MNIST



(b) CIFAR-10

Fig. 8 **a** Accuracy (%) of *nmODE* under $\mathcal{L}_\infty$ PGD attack compared to NODE trained with data augmentation (AT-NODE), ODE-TRADES (Zhang et al. 2019), and B-NODE (Yang et al. 2023) on the Fashion-MNIST dataset. **b** Accuracy(%) of *nmODE* under $\mathcal{L}_\infty$ PGD attack compared to NODE trained with data augmentation (AT-NODE), ODE-TRADES (Zhang et al. 2019), TisODE (Yan et al. 2019), and B-NODE (Yang et al. 2023) on the CIFAR-10 dataset

training set without data augmentation techniques. The variations in the MNIST dataset caused by non-adversarial perturbations are illustrated in Fig. 7. Our experimental results are shown in Fig. 6. It can be observed that *nmODE* exhibits greater resilience against Gaussian noise, resize operation, Gaussian blur, and brightness change.

5.2.2 L_∞ robustness on MNIST

On the MNIST dataset, we conduct experiments to compare the $\mathcal{L}_\infty$ robustness of *nmODE* with vanilla NODE (Chen et al. 2018), NODE trained with data augmentation (AT-NODE), ODE-TRADES (Zhang et al. 2019), TisODE (Yan et al. 2019), and B-NODE (Yang et al. 2023). We use PGD, AutoPGD, and Square Attack to attack. As shown in Table 2, *nmODE* has better performance under all the attacks.

5.2.3 L_∞ robustness on Fashion-MNIST and CIFAR-10

On the Fashion-MNIST, we compare the $\mathcal{L}_\infty$ robustness of *nmODE* with AT-NODE, ODE-TRADES, and B-NODE. AT-NODE, ODE-TRADES, and B-NODE are trained with data augmented with adversarial examples, which are generated by 40 steps L_∞ PGD attack

Table 3 Accuracy (%) of *nmODE* under $\mathcal{L}_2$ PGD attack compared to Lipschitz-MonDeq (Pabbaraju et al. 2020), Semi-MonDeq (Chen et al. 2021), Robust FI-ODE (Huang et al. 2022), NODE (Chen et al. 2018), and LyaNet (Rodriguez et al. 2022) on the MNIST and CIFAR-10 datasets

Dataset	Method	δ	Clean	Adversarial
MNIST	Lipschitz-MonDeq	0.1	95.60[a]	94.42[a]
	Semi-MonDeq	0.1	99[a]	99[a]
	Robust FI-ODE	0.1	99.35[a]	99.09[a]
	nmODE (ours)	0.1	**99.49**	**99.49**
	Lipschitz-MonDeq	0.2	95.60[a]	93.09[a]
	Robust FI-ODE	0.2	99.35[a]	98.83[a]
	nmODE (ours)	0.2	**99.49**	**99.36**
CIFAR-10	Lipschitz-MonDeq	0.141	66.66[a]	50.51[a]
	NODE w/o Lyapunov training	0.141	69.05[a]	56.94[a]
	LyaNet + Lipschitz restriction	0.141	73.15[a]	64.87[a]
	LyaNet + Sampling scheduler	0.141	82.83[a]	74.81[a]
	Robust FI-ODE	0.141	78.34[a]	67.45[a]
	nmODE (ours)	0.141	**95.46**	**91.36**

Results that surpass all competing methods are bold

[a]Results taken from Huang et al. (2022)

Table 4 Accuracy(%) of TRADES, TRADES+*nmODE* under $\mathcal{L}_\infty$ PGD attack ($\delta = 15/255$) on the CIFAR-10 dataset

	TRADES	TRADES+nmODE
Clean	**86.72**	86.30
PGD	68.70	**70.89**

Results that surpass all competing methods are bold

($\delta = 8/255$), while *nmODE* is trained only on the clean data without augmentation. The clean accuracy for AT-NODE, ODE-TRADES, B-NODE and *nmODE* are 82.10%, 83.24%, 82.68% and 88.83% respectively.

For the CIFAR-10 experiment, we use a pre-trained ResNet-18 for the feature extractor, the output of which is provided to neural ODEs as the input. The clean accuracy for ODE-TRADES, B-NODE, and *nmODE* are 90.48%, 89.16%, and 95.46%, respectively.

Results are shown in Fig. 8. As seen from the figure, *nmODE* exhibits the best performance under all the conditions.

Table 5 Accuracy (%) of *nmODE*$^+$ compared to *nmODE* under $\mathcal{L}_\infty$ adversarial attack ($\delta = 0.05$) on the MNIST dataset. *wd* denotes the magnitude of weight decay. The Robust Ratio is calculated by dividing the PGD accuracy by the clean accuracy

Method	Accuracy (%)		Robust Ratio (%)
	clean	PGD	
nmODE	98.38	96.71	98.3
nmODE$^+$ (wd = 1e−5)	98.36	97.87	99.5 (↑ 1.2)
nmODE$^+$ (wd = 1e−4)	98.30	97.91	99.6 (↑ 1.3)
nmODE$^+$ (wd = 1e−3)	97.25	96.37	99.1 (↑ 0.8)

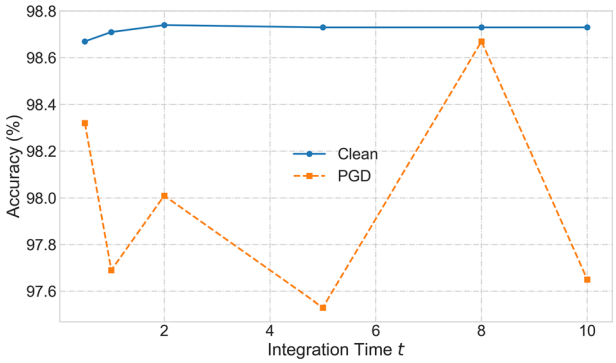


Fig. 9 Influence of integration time t of *nmODE* on the MNIST dataset under $\mathcal{L}_\infty$ PGD attack ($\delta = 0.05$)

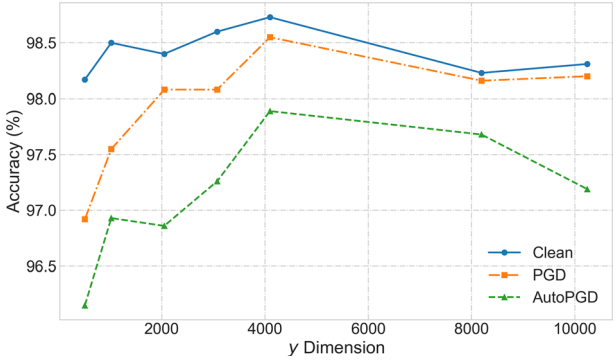


Fig. 10 Influence of y dimension of *nmODE* on the MNIST dataset under $\mathcal{L}_\infty$ PGD and AutoPGD attacks ($\delta = 0.05$)

5.2.4 L_2 robustness on MNIST and CIFAR-10

On the MNIST and CIFAR-10 datasets, we evaluate $\mathcal{L}_2$ robustness of *nmODE*, using PGD to perturb the input images within an $\mathcal{L}_2$ ball of radius 0.1 and 0.2 on MNIST, and 0.141 (36/255) on CIFAR-10. We compare *nmODE* with Lipschitz-MonDeq (Pabbaraju et al. 2020), Semi-MonDeq (Chen et al. 2021), Robust FI-ODE (Huang et al. 2022), NODE (Chen et al. 2018), and LyaNet (Rodriguez et al. 2022). As shown in Table 3, *nmODE* achieves the strongest robustness results compared to prior ODE-based approaches.

5.3 Compatibility of nmODE

We investigate the potential of integrating *nmODE* with other existing architectures to enhance the defense ability. We insert *nmODE* in front of the final FC layer of TRADES (Pang et al. 2020), which is an adversarial training defense method with a combination of tricks. We conducted experiments on the CIFAR-10 dataset and used the $\mathcal{L}_\infty$ PGD ($\delta = 15/255$) to attack. The model is trained for 100 epochs using cross-entropy loss with $\bar{t} = 1$ and $\lambda = 3$, utilizing Adam optimizer with learning rate 0.001. As illustrated in Table 4, *nmODE* enhances the defense ability of TRADES. Our experiments show that *nmODE* can be integrated into defense models to enhance the robustness.

5.4 Experiment on nmODE$^+$

To verify the effectiveness of *nmODE*$^+$, we experiment on the MNIST dataset. We use 2 fully connected layers, whose functions are feature extraction and classification prediction respectively. The connection matrices are $W^1_{2048\times784}$ and $W^2_{10\times2048}$. We insert *nmODE* between two fully connected layers. The model undergoes training on clean data utilizing cross-entropy loss, trained for 100 epochs with $\lambda = 3$ and $\bar{t} = 5\text{e}{-2}$. We use weight decay to achieve the functionality of $\mathcal{L}_{stable}$. Results are shown in Table 5. We prove that *nmODE*$^+$ has stronger stability than *nmODE*.

5.5 Ablation study

To show the effect of integration time t for *nmODE*, we conduct an ablation experiment on the MNIST dataset. The model is trained for 10 epochs with a batch size of 256, utilizing the Adam optimizer with a learning rate of 0.001. We utilize $\mathcal{L}_\infty$ PGD ($\delta = 0.05$) to attack. Results are summarized in Fig. 9.

To show the effect of y dimension for *nmODE*, we experiment on MNIST. The model is trained for 20 epochs with $\bar{t} = 1$ and $\lambda = 3$, utilizing the Adam optimizer with a learning rate of 0.001. We utilize $\mathcal{L}_\infty$ PGD and AutoPGD ($\delta = 0.05$) to attack. Experimental results are presented in Fig. 10.

6 Conclusion and discussion

In this paper, we propose a certified defense method rooted in the unique properties of *nmODE*, a variant of neural ODE distinguished by the rare attribute of global attractors. Through rigorous mathematical analysis, we demonstrate that our proposed method is capable of significantly enhancing defense against perturbations. The establishment of a novel quantitative approach allows us to articulate a clear mathematical relationship between perturbations and the defense capabilities of *nmODE*. Furthermore, we propose a training method named *nmODE*$^+$, which augments the defense capability of *nmODE* without incurring additional training costs. The experimental results presented in this paper showcase the resilience of *nmODE* to various perturbations. Notably, our method seamlessly integrates with existing neural networks and defense mechanisms, underscoring its versatility and practical applicability.

nmODE offers an intriguing avenue for developing robust systems against adversarial perturbations due to its stable mapping. Here's a discussion on practical systems and

potential areas where *nmODE* can be deployed against adversarial perturbations, along with considerations for implementation:

- Image Classification and Recognition: *nmODE* can be employed in image classification tasks where robustness against adversarial perturbations is crucial. Implementing *nmODE* in image classification involves training models using techniques like the adjoint sensitivity method or gradient-based solvers. By the stable mapping, *nmODE* implicitly smooths out small perturbations, making them less susceptible to adversarial attacks.
- Anomaly Detection: *nmODE* can be utilized for anomaly detection tasks in various domains such as cybersecurity, healthcare, or finance. By learning the continuous dynamics of normal behavior, *nmODE* can effectively identify deviations caused by adversarial attacks. Implementing *nmODE* for anomaly detection involves training on normal data distributions and detecting deviations using reconstruction errors or learned latent dynamics. Adversarial training techniques can be used to enhance the model's robustness against adversarial anomalies.
- Control Systems: In control systems, *nmODE* can be employed for robust control against adversarial disturbances. By modeling the system dynamics using continuous-time formulations, *nmODE* can adapt to unforeseen disturbances and maintain system stability. Implementing *nmODE* in control systems involves integrating them into control algorithms such as model predictive control (MPC) or reinforcement learning frameworks. Robust control strategies, including disturbance rejection and robust optimization, can be combined with *nmODE* to mitigate adversarial effects.
- Natural Language Processing: *nmODE* can be applied in natural language processing tasks such as sentiment analysis or text classification to enhance robustness against adversarial inputs, such as adversarial text or linguistic manipulations. Implementing *nmODE* involves embedding text data into continuous vector spaces and learning dynamics over these embeddings. Adversarial training methods tailored for text data, such as adversarial training with word embeddings or character-level perturbations, can be employed to improve robustness.

In subsequent research, we aim to generalize the *nmODE* to $nmODE^k$, which can be described as:

$$\begin{cases} \dot{y} = -\lambda y + f_k\{y + f_{k-1}[y + \cdots + f_1(y + \gamma)]\} \\ \gamma = Wx + b \end{cases} \tag{8}$$

where $k \in N_+$. By considering potential choices for activation functions f_k, we may get more findings on the stable mapping and defense capability of the model. Also, we will further conduct experiments in real-world scenarios to validate the effectiveness of *nmODE* in practical applications and assess its performance under diverse environmental conditions. By addressing the outlined future work, we anticipate further advancements in the field, ultimately leading to more resilient and secure neural networks.

Author contributions All authors contributed equally and reviewed the manuscript.

Data Availability The MNIST, CIFAR-10, and Fashion-MNIST datasets utilized in this research are publicly available and can be accessed through the following sources: MNIST: Available at http://yann.lecun.com/exdb/mnist/; CIFAR-10: Available at https://www.cs.toronto.edu/~kriz/cifar.html; Fashion-MNIST: Available at https://github.com/zalandoresearch/fashion-mnist.

Declarations

Conflict of interest All authors disclosed no potential Conflict of interest.

References

Andriushchenko M, Croce F, Flammarion N, Hein M (2020) Square attack: a query-efficient black-box adversarial attack via random search. In: European conference on computer vision. Springer, Cham, pp 484–501

Arvinte M, Cornelius C, Martin J, Himayat N (2023) Investigating the adversarial robustness of density estimation using the probability flow ode. arXiv preprint. arXiv:2310.07084

Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (SP), pp 39–57

Chang B, Chen M, Haber E, Chi EH (2019) Antisymmetricrnn: a dynamical system view on recurrent neural networks. arXiv preprint. arXiv:1902.09689

Chen RT, Rubanova Y, Bettencourt J, Duvenaud DK (2018) Neural ordinary differential equations. Adv Neural Inf Process Syst 31:6571–6583

Chen T, Lasserre JB, Magron V, Pauwels E (2021) Semialgebraic representation of monotone deep equilibrium models and applications to certification. Adv Neural Inf Process Syst 34:27146–27159

Chiang P, Ni R, Abdelkader A, Zhu C, Studer C, Goldstein T (2020) Certified defenses for adversarial patches. arXiv preprint. arXiv:2003.06693

Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258

Croce F, Hein M (2020) Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International conference on machine learning, pp 2206–2216

Cullen A, Montague P, Liu S, Erfani S, Rubinstein B (2022) Double bubble, toil and trouble: enhancing certified robustness through transitivity. Adv Neural Inf Process Syst 35:19099–19112

Dupont E, Doucet A, Teh YW (2019) Augmented neural ODEs. Adv Neural Inf Process Syst 32:608

Engstrom L, Tran B, Tsipras D, Schmidt L, Madry A (2017) A rotation and a translation suffice: Fooling CNNs with simple transformations. In: Proceedings of the 2019 international conference on learning representations

Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint. arXiv:1412.6572

Gu S, Rigazio L (2014) Towards deep neural network architectures robust to adversarial examples. arXiv preprint. arXiv:1412.5068

Haber E, Ruthotto L, Holtham E, Jun S-H (2018) Learning across scales—multiscale methods for convolution neural networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 32, pp 2811–2818

He K, Zhang X, Ren S, Sun J Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778 (2016)

Huang Y, Rodriguez IDJ, Zhang H, Shi Y, Yue Y (2022) FI-ODE: certified and robust forward invariance in neural ODEs. arXiv preprint. arXiv:2210.16940

Kang Q, Song Y, Ding Q, Tay WP (2021) Stable neural ode with Lyapunov-stable equilibrium points for defending against adversarial attacks. Adv Neural Inf Process Syst 34:14925–14937
Kidger P (2022) On neural differential equations. arXiv preprint. arXiv:2202.02435
Krizhevsky A (2009) Learning multiple layers of features from tiny images. https://api.semanticscholar.org/CorpusID:18268744
LeCun Y (1998) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/
Lecuyer M, Atlidakis V, Geambasu R, Hsu D, Jana S (2019) Certified robustness to adversarial examples with differential privacy. In: 2019 IEEE symposium on security and privacy (SP), pp 656–672
Levine A, Feizi S (2020) Wasserstein smoothing: Certified robustness against Wasserstein adversarial attacks. In: International conference on artificial intelligence and statistics, pp 3938–3947. PMLR
Li X, Xin Z, Liu W (2022) Defending against adversarial attacks via neural dynamic system. Adv Neural Inf Process Syst 35:6372–6383
Liu X, Xiao T, Si S, Cao Q, Kumar S, Hsieh C-J (2020) How does noise help robustness? explanation and exploration under the neural SDE framework. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 282–290
Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint. arXiv:1706.06083
Norcliffe A, Bodnar C, Day B, Simidjievski N, Liò P (2020) On second order behaviour in augmented neural odes. Adv Neural Inf Process Syst 33:5911–5921
Pabbaraju C, Winston E, Kolter JZ (2020) Estimating Lipschitz constants of monotone deep equilibrium models. In: International conference on learning representations
Pang T, Yang X, Dong Y, Su H, Zhu J (2020) Bag of tricks for adversarial training. arXiv preprint. arXiv:2010.00467
Papernot N, McDaniel P, Wu X, Jha S, Swami A (2016) Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE symposium on security and privacy (SP), pp 582–597
Raghunathan A, Steinhardt J, Liang P (2018) Certified defenses against adversarial examples. arXiv preprint. arXiv:1801.09344
Rodriguez IDJ, Ames A, Yue Y (2022) Lyanet: a Lyapunov framework for training neural ODEs. In: International conference on machine learning, pp 18687–18703. PMLR
Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv preprint. arXiv:1312.6199
Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, pp 6105–6114
Weinan E (2017) A proposal on machine learning via dynamical systems. Commun Math Stat 1(5):1–11
Weng L, Zhang H, Chen H, Song Z, Hsieh C-J, Daniel L, Boning D, Dhillon I (2018) Towards fast computation of certified robustness for ReLu networks. In: International conference on machine learning, pp 5276–5285
Wong E, Kolter Z (2018) Provable defenses against adversarial examples via the convex outer adversarial polytope. In: International conference on machine learning, pp 5286–5295
Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint. arXiv:1708.07747
Yan H, Du J, Tan VY, Feng J (2019) On robustness of neural ordinary differential equations. arXiv preprint. arXiv:1910.05513
Yang R, Jia R, Zhang X, Jin M (2023) Certifiably robust neural ode with learning-based barrier function. IEEE Control Syst Lett 7:1634–1639
Yi Z (2023) nmODE: neural memory ordinary differential equation. Artif Intell Rev 56:14403–14438
Zhai R, Dan C, He D, Zhang H, Gong B, Ravikumar P, Hsieh C-J, Wang L (2020) Macer: attack-free and scalable robust training via maximizing certified radius. arXiv preprint. arXiv:2001.02378
Zhang H, Yu Y, Jiao J, Xing E, El Ghaoui L, Jordan M (2019) Theoretically principled trade-off between robustness and accuracy. In: International conference on machine learning, pp 7472–7482. PMLR
Zheng S, Song Y, Leung T, Goodfellow I (2016) Improving the robustness of deep neural networks via stability training. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4480–4488
Zizzo G, Rawat A, Sinn M, Maffeis S, Hankin C (2021) Certified federated adversarial training. arXiv preprint. arXiv:2112.10525