Proceedings of Machine Learning Research vol 145:497–515, 2021 2nd Annual Conference on Mathematical and Scientific Machine Learning

Adversarial Robustness of Stabilized Neural ODE Might be from Obfuscated Gradients

Yifei Huang YHUANGCC@CONNECT.UST.HK
Department of Mathematics, Hong Kong University of Science and Technology

Yaodong Yu YYU@EECS.BERKELEY.EDU
Department of EECS, University of California at Berkeley

Hongyang Zhang ✉ HONGYANZ@TTIC.EDU
Toyota Technological Institute at Chicago and University of Waterloo

Yi Ma YIMA@EECS.BERKELEY.EDU
Department of EECS, University of California at Berkeley

Yuan Yao ✉ YUANY@UST.HK
Department of Mathematics, Hong Kong University of Science and Technology

Editors: Joan Bruna, Jan S Hesthaven, Lenka Zdeborova

Abstract

In this paper we introduce a provably stable architecture for Neural Ordinary Differential Equations (ODEs) which achieves non-trivial adversarial robustness under white-box adversarial attacks even when the network is trained naturally. For most existing defense methods withstanding strong white-box attacks, to improve robustness of neural networks, they need to be trained adversarially, hence have to strike a trade-off between natural accuracy and adversarial robustness. Inspired by dynamical system theory, we design a stabilized neural ODE network named SONet whose ODE blocks are skew-symmetric and proved to be input-output stable. With natural training, SONet can achieve comparable robustness with the state-of-the-art adversarial defense methods, without sacrificing natural accuracy. Even replacing only the first layer of a ResNet by such a ODE block can exhibit further improvement in robustness, e.g., under PGD-20 ($\ell_\infty = 0.031$) attack on CIFAR-10 dataset, it achieves 91.57% and natural accuracy and 62.35% robust accuracy, while a counterpart architecture of ResNet trained with TRADES achieves natural and robust accuracy 76.29% and 45.24%, respectively. To understand possible reasons behind this surprisingly good result, we further explore the possible mechanism underlying such . We show that the adaptive stepsize numerical ODE solvers, such as adaptive HEUN2, BOSH3, and DOPRI5, have a gradient masking effect that fails the PGD attacks which are sensitive to gradient information of training loss; on the other hand, they cannot fool the CW attack of robust gradients and the SPSA attack that is gradient-free. This provides a new explanation that the adversarial robustness of ODE-based networks mainly comes from the obfuscated gradients in numerical ODE solvers with adaptive step sizes. (Source codes: https://github.com/silkylove/SONet; https://github.com/yao-lab/SONet)

Keywords: Adversarial Robustness, Neural ODE, Lyapunov Stability, Numerical ODE Solvers, Adaptive Step Size

1. Introduction

Adversarial robustness is a central object of study in machine learning (Carlini et al., 2019; Zhang et al., 2019b; Madry et al., 2018; Kolter and Wong, 2018), computer security (Sharif et al., 2016; Meng and Chen, 2017), and many other domains (Song et al., 2018; Xie et al., 2017; Jia and Liang, 2017). In machine learning, study of adversarial robustness has led to significant advance in understanding the generalization (Schmidt et al., 2018; Carmon et al., 2019; Alayrac et al., 2019; Zhai et al., 2019), interpretability of learning models (Tsipras et al., 2019), and connecting robust statistics (Gao et al., 2019, 2020). In computer security, adversarial robustness serves as an indispensable component towards AI safety against adversarial threat, in a range of security-critical systems and applications such as autonomous vehicles (Eykholt et al., 2018) and biometric authorization (Thys et al., 2019). The problem of achieving adversarial robustness can be stated as learning a classifier with high test accuracy on both natural and *adversarial examples*. The adversarial example is either in the form of unrestricted transformations, such as rotation and translation of natural examples, or in the form of perturbations with bounded norms. The focus of this work is the latter setting.

Probably one of the most successful techniques to enhance model robustness is by *adversarial training* (Madry et al., 2018; Zhang et al., 2019c). In the adversarial training, the defenders simulate adversarial examples against current iteration of model and then feed them into the training procedure in the next round. Despite a large literature devoted to the study of adversarial training, many fundamental questions remain unresolved. One of the long-standing questions is the interpretability: although adversarial training is an effective way to defend against certain adversarial examples, it remains unclear why current designs of network architecture are vulnerable to adversarial attacks without adversarial training. This question becomes more challenging when we consider the computational issues. Taking the perspective of Pontryagin Maximum Principle (or Bellman Equation) for differential games induced by adversarial training, Zhang et al. (2019a) reduces adversarial training to merely updating the weights of the first layer that significantly reduces the computational cost. Yet in optimization, adversarial training is notorious for its instability due to the non-convex non-concave minimax nature of its loss function.

When the "simulated" adversarial examples in the training procedure do not conceptually match those of attackers, adversarial training can be vulnerable to adversarial threat as well. This is known as the norm-agnostic setting, and there is significant evidence to indicate that adversarial training suffers from brittleness against attacks in ℓ_2 and ℓ_∞ norms simultaneously (Li et al., 2019). Furthermore, due to an intrinsic trade-off between natural accuracy and adversarial robustness (Tsipras et al., 2019; Zhang et al., 2019c), adversarial training typically leads to more than 10% reduction of accuracy compared with natural training.

Stability principle of dynamical systems has been applied to adversarial training to enhance the robustness. Inspired by the initial value stability of convection-diffusion partial differential equation and the Feynman-Kac formula of solutions, Wang et al. (2019) designs ResNet ensembles with activation noise that exhibits improvements in both natural and robust accuracies for adversarial training. Moreover, motivated by the fact in numerical ODEs that implicit (backward) Euler discretization has better stability than explicit (forward) Euler discretization that current ResNets exploit, Li et al. (2020) designs implicit Euler based skip-connections to enhance ResNets with better stability and adversarial robustness. However, all these studies are limited to adversarial training rather than natural training.

In response to the limitations of adversarial training, designing network architecture towards natural training as robust as adversarial training has received significant attention in recent years. On one hand, most positive results for obtaining adversarial robustness have focused on controlling Lipschitz constants explicitly in the training procedure, such as requiring each convolutional layer be composed of orthonormal filters (Cisse et al., 2017), or restricting the spectral radius of the matrix in each layer to be small (Qian and Wegman, 2019). These approaches, however, do not achieve comparable robustness as adversarial training against ℓ_∞-norm attacks. On the other hand, with the introduction of ordinary different equations into neural networks (Chen et al., 2018), the adversarial robustness for neural ordinary differential equations (ODEs) network architecture have been attracting rising attention. Yan et al. (2019) found that ODE networks with natural training is more robust against adversarial examples compared with traditional conventional neural networks, but the robustness of ODE networks is much weaker than the state-of-the-art result by adversarial training.

1.1. Our methodology and results

We begin with designing ODE networks analogous to the residual networks. Our ODE network is a natural extension of the Residual Network: when we solve the ODE system by the explicit (forward) Euler method, the two types of networks can be made equivalent. Nevertheless, to ensure the output of our networks to be less sensitive to perturbations in input, we further require our ODE networks to be stable by design. It has been well known in the dynamical system theory (Callier and Desoer, 1991) that input-output stability is an important property for a system to be insensitive (and even robust) to input noise and perturbations. We rigorously show that the resulting networks are stable in the Lyapunov sense, provided that the two weight matrices in each ODE block are skew-symmetric to each other up to an arbitrarily small damping and the activation function is strictly monotonically increasing. The design works for both convolutional and fully-connected neural networks.

Our stability analysis naturally leads to a new formulation of network architecture which has several appealing properties; in particular, it inherits all the benefits of Neural ODE such as parameter- and memory-efficiency, adaptive computation, etc., and the algorithm achieves comparable robustness on a range of benchmarks as the state-of-the-art adversarial training methods. To understand possible reasons behind this surprisingly good result, we further explore possible mechanisms and disclose the obfuscated gradients caused by adaptive step sizes of numerical ODE solvers.

The main contribution and discovery in this report can be summarized as follows.

- Theoretically, we parametrize ODE networks analogous to the residual networks. Our stability analysis shows that the ODE system is Lyapunov stable, provided that the activation function is strictly monotonically increasing and the two weight matrices in the ODE block are skew-symmetric with each other, up to an arbitrarily small damping factor.

- Algorithmically, inspired by our stability analysis, we propose a new formulation of neural ODE network architecture, named **S**tabilized neural **O**DE **Net**work (SONet). The architecture is robust to small perturbations as each ODE block is provably stable in the sense of Lyapunov.

- Experimentally, we show that natural training of the proposed architecture achieves non-trivial adversarial robustness in white-box PGD attacks, and even better than the state-of-the-art ResNet10 adversarially trained by TRADES under white-box ℓ_∞ and ℓ_2 PGD^{20} attacks.

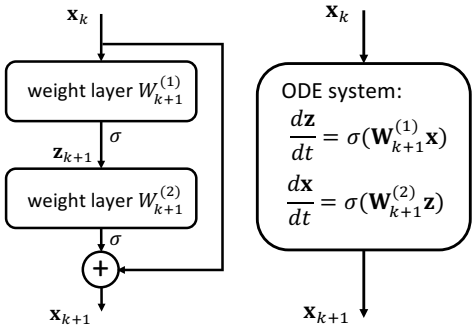
- Furthermore, a possible interpretation for the adversarial robustness of ODE-based networks is provided, suggesting that numerical ODE solvers with adaptive step sizes (e.g. adaptive HEUN2, BOSH3, and DOPRI5) may lead to obfuscated gradients via large error tolerance in adaptive step size choice, which fails the gradient based attacks like PGD but may not fool robust gradient attacks like CW and gradient-free attacks like SPSA.

2. Introduction

Before proceeding, we define some notations and formalize our model setup in this section.

2.1. Notations

We will use bold capital letters such as $\mathbf{W}$ to represent matrices, bold lower-case letters such as $\mathbf{x}$ to represent vectors, and lower-case letters such as t to represent scalars. Specifically, we denote by $\mathbf{0}$ the all-zero vector, by $\mathbf{1}$ the all-one vector, by $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$ the input vector to each architecture block, and by $\mathbf{z} \in \mathbb{R}^{d_{\text{out}}}$ the output vector, where d_{in} does not necessarily equal to d_{out}. Denote by $\sigma(\cdot)$ the element-wise activation function, and $\sigma'(\cdot)$ is its (sub-)gradient. We will frequently use $d\mathbf{x}(t)/dt$ to represent the differential of $\mathbf{x}(t)$ w.r.t. the time variable t. For norms, we denote by $\|\cdot\|$ a generic norm. Examples of norms include $\|\mathbf{x}\|_p$, the ℓ_p norm of vector $\mathbf{x}$ for $p \geq 1$. We will use $f_1 \circ f_2(\cdot)$ to represent the composition of two functions $f_1(\cdot)$ and $f_2(\cdot)$. Denote by $\mathbb{B}(\mathbf{x}, \epsilon)$ a neighborhood of $\mathbf{x}$: $\{\mathbf{x}' : \|\mathbf{x} - \mathbf{x}'\| \leq \epsilon\}$. Throughout the paper, for any given loss function $\mathcal{L}(f, \mathbf{x})$ and data (set) $\mathbf{x}$, we will term the optimization procedure $\min_f \max_{\mathbf{x}' \in \mathbb{B}(\mathbf{x}, \epsilon)} \mathcal{L}(f, \mathbf{x}')$ as *adversarial training* and term the optimization procedure $\min_f \mathcal{L}(f, \mathbf{x})$ as *natural training*.



(*a*) ResNet block (*b*) ODENet block

Figure 1: Network architecture.

2.2. ODE Blocks

In the Residual Networks (ResNets, a.k.a. Euler networks) (He et al., 2016), the basic blocks follow the architecture[1] (see Figure 1(*a*)):

$$\begin{aligned} \frac{\mathbf{z}_{k+1} - \mathbf{z}_k}{\Delta t} &= \sigma(\mathbf{W}_{k+1}^{(1)}\mathbf{x}_k), \\ \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t} &= \sigma(\mathbf{W}_{k+1}^{(2)}\mathbf{z}_{k+1}), \\ \mathbf{z}_k &= 0,\ \Delta t = 1, \end{aligned} \tag{1}$$

where $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ are the input and ouput of the k-th ResNet block, $\mathbf{z}_{k+1}$ is the intermediate layer, and $\mathbf{W}_{k+1}^{(1)}$ and $\mathbf{W}_{k+1}^{(2)}$ are the weight matrices which represent either the fully-connected or the convolutional operators. In the Neural ODE, in constrast, Chen et al. (2018) took the limit of the

1. Without loss of generality, we assume the bias term is a zero vector for simplicity, although our analysis works for the general case as well.

finite differences over the infinitesimal Δt and parameterized the continuous dynamics of hidden units using an ODE specified by a neural network:

$$\mathbf{x}_{k+1} = f_{\text{NeuralODE-k}}(\mathbf{x}_k; t_0): \quad \frac{d\mathbf{x}(t)}{dt} = \sigma(\mathbf{W}_{k+1}^{(2)}\sigma(\mathbf{W}_{k+1}^{(1)}\mathbf{x}(t))),\ \mathbf{x}(0) = \mathbf{x}_k,\ \mathbf{x}_{k+1} = \mathbf{x}(t_0), \tag{2}$$

where $\mathbf{x}_k$ is the initial condition of $\mathbf{x}(t)$, i.e., the input, and the output $\mathbf{x}_{k+1}$ is the evolution of $\mathbf{x}(t)$ at time t_0.

Our study is motivated by the Neural ODE. We focus on a parametric model similar to ResNet block (1) (see Figure 1(*b*)):

$$\begin{aligned} \frac{d\mathbf{z}(t)}{dt} &= \sigma(\mathbf{W}_{k+1}^{(1)}\mathbf{x}(t) - \gamma\mathbf{z}(t)), \\ \frac{d\mathbf{x}(t)}{dt} &= \sigma(\mathbf{W}_{k+1}^{(2)}\mathbf{z}(t) - \gamma\mathbf{x}(t)), \\ \mathbf{x}_{k+1} &= \mathbf{z}(t_0), \quad \mathbf{x}(0) = \mathbf{x}_k, \quad \mathbf{z}(0) = \mathbf{z}_k, \end{aligned} \tag{3}$$

where $\mathbf{x}_k$ and $\mathbf{z}_k$ are the initial conditions of $\mathbf{x}(t)$ and $\mathbf{z}(t)$, respectively, $\gamma > 0$ is a small positive constant as the damping factor and the output $\mathbf{x}_{k+1}$ is the evolution of $\mathbf{z}(t)$ at time t_0. When we solve ODE system (3) by the Euler solver with time step 1 and set γ to be 0, ODE block (3) is equivalent to ResNet block (1).

Flexibility of parametric model in (4). Compared with the previous Neural ODE (Chen et al., 2018) defined in (2), which can only deal with the case when the size of input $\mathbf{x}_k$ is equal to the size of output $\mathbf{x}_{k+1}$, the parametric model in (3) is able to handle the case when $\dim(\mathbf{x}_{k+1}) \neq \dim(\mathbf{x}_k)$. The intermediate layer $\mathbf{z}(t)$ can be viewed as an auxiliary layer, which makes our model more flexible.

3. Stability of ODE Blocks

In this section, we present our stability analysis for ODE system (3) that serves as a guiding principle in the design of network architecture against adversarial examples. Our analysis leads to the following guarantee on the stability of the ODE system.

Theorem 1 (Stability of ODE Blocks) *Suppose that the activation function σ is strictly monotonically increasing, i.e., $\sigma'(\cdot) > 0$ and positive damping factor γ is small. Let $\mathbf{W}_{k+1}^{(2)} = -\mathbf{W}_{k+1}^{(1)\top}$. Then for any implementation of network parameters, the forward propagation (3) is stable in the sense of Lyapunov; that is, for all $\delta > 0$, there exists a stable radius $\epsilon(\delta) > 0$ such that if $\|\mathbf{x}_0 - \mathbf{x}_0'\| \leq \epsilon(\delta)$, we have $\|f_{\text{ODENet-k}}(\mathbf{x}_0; t_0) - f_{\text{ODENet-k}}(\mathbf{x}_0'; t_0)\| \leq \delta$ for all $t_0 > 0$.*

Theorem 1 demonstrates that there exists a *universal* stability radius $\epsilon > 0$ (independent of integration time t_0) such that small change of $\mathbf{x}_0$ within the ϵ-ball causes small change of $f_{\text{ODENet-k}}(\mathbf{x}_0; t_0)$ for *all* $t_0 > 0$. In contrast, the continuity in the original design of Neural ODE (Chen et al., 2018) does not justify the existence of such universal stability radius for all $t_0 > 0$. Our theory shows that the quantity $\|f_{\text{ODENet}}(\mathbf{x}_0; t) - f_{\text{ODENet}}(\mathbf{x}_0'; t_0)\|$ does not diverge as t_0 grows. So the ODE is robust w.r.t. its initial condition, the input of the network.

Intuition behind the stability. Our ODE block (3) has guaranteed stability without any explicit regularization on the smoothness of its input and output. To see how the skew-symmetric architecture encourages stability, let us ignore for now the nonlinear activation $\sigma(\cdot)$ and γ in the ODE block (3) and consider its linearized version:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{W}_{k+1}^{(2)}\\ -\mathbf{W}_{k+1}^{(2)\top} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} \doteq \mathbf{A}_{k+1}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}, \tag{4}$$
$$\mathbf{x}_{k+1} = \mathbf{z}(t), \quad \mathbf{x}(0) = \mathbf{x}_k, \quad \mathbf{z}(0) = \mathbf{z}_k.$$

As the linear system matrix $\mathbf{A}_{k+1}$ is skew symmetric, one can show that the solution to the above system is given by Callier and Desoer (1991):

$$\begin{bmatrix}\mathbf{x}(t)\\ \mathbf{z}(t)\end{bmatrix} = \mathbf{\Phi}\begin{bmatrix}\mathbf{x}(0)\\ \mathbf{z}(0)\end{bmatrix},$$

where the state-transition matrix $\mathbf{\Phi}$ is an orthogonal matrix $\mathbf{\Phi}\mathbf{\Phi}^\top = \mathbf{I}$. Hence the input-output of the linearized system is always norm-preserving.

Below we give a formal proof of stability of the ODE block with the nonlinear activation, i.e. Theorem 1, based on results from system theory (Aström and Murray, 2010).

Proof We observe that Eqn. (3) has an equivalent expression, $\mathbf{x}_{k+1} = f_{\text{ODENet}}(\mathbf{x}_k; t_0)$:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} = \sigma\left(\begin{bmatrix}\mathbf{0} & -\mathbf{W}_{k+1}^\top\\ \mathbf{W}_{k+1} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} - \gamma\mathbf{I}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}\right),$$
$$\mathbf{x}(0) = \mathbf{x}_k,\ \mathbf{z}(0) = \mathbf{z}_k,\ \mathbf{x}_{k+1} := \mathbf{z}(t_0).$$

Denote by

$$\mathbf{A}_{k+1} := \begin{bmatrix}\mathbf{0} & -\mathbf{W}_{k+1}^\top\\ \mathbf{W}_{k+1} & \mathbf{0}\end{bmatrix}.$$

Note that $\mathbf{A}_{k+1}$ is a skew-symmetric matrix such that $\mathbf{A}_{k+1} = -\mathbf{A}_{k+1}^\top$. So $Re[\lambda_i(\mathbf{A}_{k+1})] \leq 0$ for all i, where $Re[\cdot]$ represents the real part of a complex variable and $\lambda_i(\mathbf{A}_{k+1})$ is the i-th eigenvalue of matrix $\mathbf{A}_{k+1}$.

We note that an ODE system is stable if $Re[\lambda_i(\mathbf{J}_{k+1})] < 0$ (Aström and Murray, 2010), where $\mathbf{J}_{k+1}$ is the Jacobian of the ODE:

$$\begin{aligned}\mathbf{J}_{k+1} &:= \nabla_{[\mathbf{x};\mathbf{z}]}\left(\sigma\left(\begin{bmatrix}\mathbf{0} & -\mathbf{W}_{k+1}^\top\\ \mathbf{W}_{k+1} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix} - \gamma\mathbf{I}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}\right)\right)\\ &=: \mathbf{D}_{k+1}(\mathbf{A}_{k+1} - \gamma\mathbf{I}),\end{aligned}$$

where we have defined

$$\mathbf{D}_{k+1} := \text{Diag}\left(\sigma'\left(\begin{bmatrix}-\gamma & -\mathbf{W}_{k+1}^\top\\ \mathbf{W}_{k+1} & -\gamma\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{z}\end{bmatrix}\right)\right).$$

Because $\sigma'(\cdot) > 0$, the matrix $\mathbf{D}_{k+1}^{-1/2}$ exists. We observe that

$$\mathbf{J}_{k+1} \sim \mathbf{D}_{k+1}^{1/2}(\mathbf{A}_{k+1} - \gamma\mathbf{I})\mathbf{D}_{k+1}^{1/2},$$

where the notation $\sim$ means the two matrices are similar. Since similar matrices have the same eigenvalues, for all i, we have

$$\lambda_i(\mathbf{J}_{k+1}) = \lambda_i(\mathbf{D}_{k+1}^{1/2}(\mathbf{A}_{k+1} - \gamma\mathbf{I})\mathbf{D}_{k+1}^{1/2}). \tag{5}$$

For the right hand side in Eqn. (5), $Re[\lambda_i(\mathbf{A}_{k+1})] \leq 0$ So $Re[\lambda_i(\mathbf{A}_{k+1} - \gamma\mathbf{I})] < 0$, and matrix $\mathbf{D}_{k+1}$ is positive diagonal. Combining with Eqn. (5), we have $Re[\lambda_i(\mathbf{J}_{k+1})] < 0$. Thus, we have $\|(\mathbf{x}(t), \mathbf{z}(t))\| \leq \|(\mathbf{x}(0), \mathbf{z}(0))\|$ and when we set the initial condition $\mathbf{z}(0) = \mathbf{z}(k) = \mathbf{x}(k)$, there holds $\|\mathbf{x}(t)\| \leq \|(\mathbf{x}(t), \mathbf{z}(t))\| \leq \|(\mathbf{x}(0), \mathbf{z}(0))\| \leq \sqrt{2}\|\mathbf{x}(0)\|$ for any $t > t_0$. Alternatively, one can also achieve $\|\mathbf{x}(t)\| \leq \|(\mathbf{x}(t), \mathbf{z}(t))\| \leq \|(\mathbf{x}(0), \mathbf{z}(0))\| = \|\mathbf{x}(0)\|$ if we choose initialization $\mathbf{z}(0) = 0$. Finally, the Lyapunov stability is valid with respect to Euclidean ℓ_2-norm. For other equivalent ℓ_p-norms ($1 \leq p \leq \infty$), the result holds up to a constant that depends on the input dimension. The proof is completed. ■

Another quantity governing the robustness of a network is its depth. Empirically, deeper networks enjoy better robustness against adversarial perturbations (Madry et al., 2018). This is probably because the score function of a ReLU-activated neural network is characterized by a piecewise affine function (Croce et al., 2018); deeper neural network implies smoother approximation of the ground-truth score function. Since ODE networks are provable deep limit of ResNets (Avelin and Nyström, 2019; Thorpe and van Gennip, 2018), the proposed networks implicitly enjoy the benefits of depth.

4. Architecture Design of ODE Networks

Architecture design of ODE blocks. Theorem 1 sheds light on architecture designs of ODE blocks. In order for the ODE to be stable w.r.t. its input at the inference time, the theorem suggests parametrizing ODE network (3) with $\mathbf{W}_{k+1}^{(2)} = -\mathbf{W}_{k+1}^{(1)T}$ and a strictly increasing activation function. We name our network SONet, standing for **S**tabilized **O**DE **Net**work.

Probably the most relevant work to our design is that of Haber and Ruthotto (2017), where Haber and Ruthotto (2017) proposed similar skew-symmetric architecture, but for the Euler networks. In addition, Haber and Ruthotto (2017) discussed the proposed architecture in the context of exploding and vanishing gradient phenomenon. In contrast, our work sheds light on algorithmic designs for adversarial defenses which is different to Haber and Ruthotto (2017). We show that a good ODE solver for problem (3) suffices to imply a robust network to adversarial attacks.

Benefits of skew-symmetric architecture. The skew-symmetric architecture of ODE blocks has many structural benefits that one can exploit. *Change of dimensionality*: the introduction of the auxiliary variable $\mathbf{z} \in \mathbb{R}^{d_{\text{out}}}$ enables us to change the dimension of the input and output vectors; that is, the input variable $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$ may have different dimensions as the output variable $\mathbf{z} \in \mathbb{R}^{d_{\text{out}}}$. This is in sharp contrast to the original design of Neural ODE (Chen et al., 2018), where the input and output vectors of each ODE block must have the same dimension. *Parameter efficiency*: the skew-symmetric ODE block has only half number of parameters compared to the ResNet blocks and the original design of Neural ODE blocks due to parameter sharing. *Inference-time robustness*: the established architecture enjoys stability (see Theorem 1). Furthermore, an expected side-benefit of our design is that it automatically inherits all the benefits of Neural ODE (Chen et al., 2018): memory efficiency, adaptive computation, invertible normalizing flows, and many others.

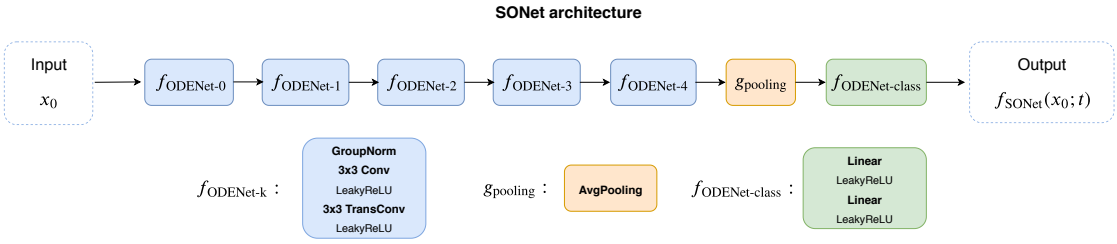


Figure 2: Stabilized neural ODE Network (SONet) architecture example. Both $f_{\text{ODENet-k}}$ and $f_{\text{ODENet-class}}$ are built on our stable ODE block defined in (3).

Construction of robust ODE networks. Our construction of ODE networks builds upon the architecture design of ODE block (3).

- *Feature-extraction block* $f_{\text{ODENet-k}}$: The feature blocks aim at extracting the feature of each instance. For the image classification tasks, the operation of multiplying $\mathbf{z}$ with $\mathbf{W}_{k+1}$ in Eqn. (3) serves as a convolution operator, and correspondingly, $\mathbf{W}_{k+1}^T$ serves as a *transposed convolution* (a.k.a. de-convolution) operator which shares a common kernel with $\mathbf{W}_{k+1}$. The input and output dimensions d_{in} and d_{out} are equal. We set the initial condition $\mathbf{z}_k$ as $\mathbf{x}_k$, the input of the ODE block.
- *Classification block* f_{class}: At the top layer of the network is the classification block which is characterized by a fully-connected operator, mapping the extracted feature to the confidence value associated with each class. The matrix $\mathbf{W}_{k+1}$ parametrizes the weight matrix of the fully-connected layer and $\mathbf{W}_{k+1}^T$ is its transpose. The input dimension d_{in} and the output dimension d_{out} are equal to the feature size and the number of classes, respectively, so d_{in} might not equal to d_{out}. We set the initial condition $\mathbf{z}_k = \mathbf{z}(0)$ as $\mathbf{1}$. This is conceptually consistent with the argument that we have no prior knowledge on the true label of any given instance.

Our ODE network is therefore a stack of various building blocks:

$$f_{\text{ODENet}}(\mathbf{x}_0; t) := f_{\text{class}} \circ g_{\text{pooling}} \circ f_{\text{ODENet-L}} \circ \cdots \circ f_{\text{ODENet-0}} \circ g_{\text{channel-copy}}(\mathbf{x}_0), \tag{6}$$

where $\mathbf{x}_0$ is the input instance, L is the number of layers, g_{pooling} represents the average pooling operator, and $g_{\text{channel-copy}}$ is the "channel-copy" layer which copies $\mathbf{x}_0$ along the channel direction in order to increase the width of the network. The function $f_{\text{ODENet}} : \mathcal{X} \to \mathbb{R}^C$ is the *score function* which maps an instance to logits over classes. An example network is shown in Figure 2, this example consists 5 feature-extraction blocks, a pooling layer and a classification block.

Comparisons with prior work. We compare our approach with several related lines of research in the prior work. One of the best known algorithms for adversarial robustness is based on adversarial training. The algorithms approximately solve a minimax problem

$$\min_f \sum_{i=1}^n \left\{ \max_{\mathbf{x}'_{(i)} \in \mathbb{B}(\mathbf{x}_{(i)}, \epsilon)} \mathcal{P}(f, \mathbf{x}'_{(i)}) \right\},$$

where $\mathbf{x}_{(i)}$ represents the i-th instance, and $\mathcal{P}(\cdot,\cdot)$ is the payoff function between defender and attacker, which captures the smoothness of model f in an explicit manner; examples of $\mathcal{P}(f, \mathbf{x}'_{(i)})$ include robust optimization $\mathcal{L}(f(\mathbf{x}'_{(i)}), \boldsymbol{y}_{(i)})$ (Madry et al., 2018) and TRADES $\mathcal{L}(f(\mathbf{x}'_{(i)}), \boldsymbol{y}_{(i)}) + \beta\mathcal{L}(f(\mathbf{x}'_{(i)}), f(\mathbf{x}_{(i)}))$ (Zhang et al., 2019c), where $\mathcal{L}$ is the cross-entropy loss and $\boldsymbol{y}_{(i)}$ is the one-hot label. Prior to ours, random smoothing (Cohen et al., 2019) and stability training (Zheng et al., 2016) are other techniques towards natural training as robust as adversarial training by adding small Gaussian noises to the input images. Our work, on the other hand, is paralleled to these two lines of research as we focus on the network architecture design. The combination of these methods may further enhance the robustness of learning systems.

Another more related line of research is by network architecture design. Parseval networks (Cisse et al., 2017) and L_2-nonexpansive neural networks (Qian and Wegman, 2019) explicitly bounded the Lipschitz constant by either requiring each fully-connected or convolutional layer be composed of orthonormal filters (Cisse et al., 2017), or restricting the spectral radius of the matrix in each layer to be small (Qian and Wegman, 2019). In complex problem domains, however, the explicit Lipschitz constraints may negatively affect the expressive power of the networks and overly trade off natural accuracy against adversarial robustness. Xie et al. (2019) involved non-local mean denoiser in the architecture design of ResNet. But the model is vulnerable to attcks without adversarial training. Svoboda et al. (2019) proposed PeerNets, a family of convolutional networks alternating classical Euclidean convolutions with graph convolutions. Yan et al. (2019) explored robustness properties of neural ODEs and proposed the time-invariant steady neural ODE (TisODE), which regularizes the flow on perturbed data. But the model is weak under PGD attacks. In contrast, in this work, we explore the possibility of enhancing robustness for classic (non-graph) networks with natural training.

5. Adversarial Robustness under PGD Attacks

In this section, we evaluate the adversarial robustness of our proposed architecture against projected gradient descent attacks and show the Stablized ODE Block/Net can achieve the state-of-the-art performance even competitive to adversarial training by TRADES, without losing natural accuracy. We use $\mathcal{A}_{\text{nat}}(f)$ to denote the natural accuracy of the model, and $\mathcal{A}_{\text{rob}}(f)$ to denote the robust accuracy against adversarial attacks. Additional experiments are provided in Appendix.

5.1. Experimental Setup

We first introduce the experimental setup for datasets, deep neural network architectures, adversarial attacks and adversarial defense methods.

ResNet: We apply the ResNet with 10 layers as the baseline model, denoted by ResNet10. Compared with ResNet18 with 2-layer basic block, we use 1-layer basic block for ResNet10. The first layer of ResNet10 is the convolution layer with Batchnorm (Ioffe and Szegedy, 2015) and ReLU activation, then followed by four 1-layer residual basic blocks. Within each basic block there are two convolution layers. Average pooling is applied after the residual blocks and the last layer is a fully connected layer with softmax.

SONet: We apply our stable skew-symmetric ODE block (defined in Eqn. (3)) as the building block in the SONet. More specifically, we replace each residual basic block in the ResNet10 architecture with the proposed stable skew-symmetric ODE block. Besides the replaced residual

blocks, as shown in Figure 2, we replace the same first convolution layer with the stable ODE block $f_{\text{ODENet-0}}$, and replace the last fully connected linear layer with the stable ODE block $f_{\text{ODENet-class}}$.

SOBlock: We replace the first convolution layer in **ResNet10** above by our proposed skew-symmetric ODE block (defined in Eqn. (3)) and leave the other parts unchanged.

Additionally, in order to compare the performance of SONet, SOBlock and ResNet with different number of parameters, we scale the model capacity by changing the input channel from 32 to 64.

Adversarial attacks: For **White-Box** attacks, we focus on ℓ_∞-norm, ℓ_2-norm projected gradient descent (PGD) and CW_∞ (Carlini and Wagner, 2017) adversairal attacks to evaluate the adversarial robustness of different models. For ℓ_∞ PGD attack, the update rule is defined as $\mathbf{x}_i' \leftarrow \Pi_{\mathbb{B}_\infty(\mathbf{x}_i,\epsilon)}(\mathbf{x}_i' + \eta_1 \,\mathsf{sign}(\nabla_{\mathbf{x}_i'}\mathcal{L}(f(\mathbf{x}_{(i)}'), \boldsymbol{y}_{(i)})))$, where $\Pi_{\mathbb{B}_\infty(\cdot,\cdot)}$ is the projection operator with respect to ℓ_∞-norm, $\mathcal{L}$ is the cross-entropy loss, $\mathbf{x}_i'$ is initialized as the original input $\mathbf{x}_i$, ϵ is the adversarial perturbation distance, and η_1 is the attack step size. For ℓ_2 PGD attack, the update rule is defined as $\mathbf{x}_i' \leftarrow \Pi_{\mathbb{B}_2(\mathbf{x}_i,\epsilon)}(\mathbf{x}_i' + \eta_1 \,\mathsf{norm}_2(\nabla_{\mathbf{x}_i'}\mathcal{L}(f(\mathbf{x}_{(i)}'), \boldsymbol{y}_{(i)})))$, where $\Pi_{\mathbb{B}_2(\cdot,\cdot)}$ is the projection operator with respect to ℓ_2-norm, and the norm_2 is the normalization operator, i.e., $\mathsf{norm}_2(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$. For CW_∞ attack, it minimizes $c \cdot f(x+\delta) + \|\delta\|_\infty$ with respect to δ such that $x + \delta \in [0,1]^n$ where $c > 0$ is a suitably chosen constant. For **Black-Box** attack, we use simultaneous perturbation stochastic approximation (SPSA) (Uesato et al., 2018) adversarial attack which is one of the most powerful gradient free attacks and it minimizes $m_\theta(x)_{y_0} - \max_{j \neq y_0} m_\theta(x)_j$ with respect to x such that $\|x - x_0\|_\infty < \epsilon$ where $m_\theta(x)_j$ denotes the output logit for the class j and y_0 is the true label. Unless explicitly stated, on CIFAR10 dataset, we set the perturbation distance $\epsilon = 0.031$, the attack step size $\eta_1 = 0.003$ for ℓ_∞ PGD attack, and we set the perturbation distance $\epsilon = 0.5$, the attack step size $\eta_1 = 0.1$ for ℓ_2 PGD attack. For CW_∞ attack, we set the perturbation distance $\epsilon = 0.031$, the max-iterations $K = 100$. And we apply the $\epsilon = 0.031$, the number of iterations $K = 20$ and the number of samples to be 32 for SPSA attack.

Adversarial training: We use TRADES (Zhang et al., 2019c) as our adversarial training baselines for comparison. We do not compare with other adversarial training approaches because TRADES is known as the state-of-the-art defense method which won the NeurIPS 2018 Adversarial Vision Challange (Brendel et al., 2020). On CIFAR10 dataset, we set the ℓ_∞ perturbation distance $\epsilon = 0.031$, perturbation step size $\eta_1 = 0.007$, number of iterations $K = 10$ for TRADES. For TRADES, we train two models and set the regularization parameter as $1/\lambda = 1.0$ and $1/\lambda = 6.0$.

Training settings: On CIFAR10 dataset, for all mentioned models, we set the total epoch $T = 350$, batch size $B = 100$, the initial learning rate $\eta = 0.01$ (decay 0.1 at 150 and 300 epochs respectively), and apply stochastic gradient descent (SGD) with momentum 0.9 as the optimizer. No weight decay is used during training. Unless explicitly stated, for all skew-symmetric ODE blocks, we set the constant γ in (3) to be 0 and use DOPRI5 solver which is an adaptive solver with 0.1 error tolerance.

5.2. Projected Gradient Descent (PGD) Attacks

We shall see that our proposed network, SONet, is able to achieve nontrivial ℓ_2 and ℓ_∞ white-box PGD robust accuracy on CIFAR10 dataset, only with natrual training. Moreover, SOBlock even outperforms ResNet10 with TRADES training with regard to both natural accuracy and robust accuracy on both PGD^{20} and PGD^{1000} attacks, although TRADES is able to achieve a better tradeoff compared with standard adversarial training (Zhang et al., 2019c).

Table 1: Comparisons between SONet, SOBlock with natural training and ResNet10 with TRADES under white-box PGD adversarial attacks on CIFAR10 dataset.

Model	Channel	Under which attack	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$	
				$\epsilon = 0.031(\ell_\infty)$	$\epsilon = 0.5(\ell_2)$
SONet	32	PGD^{20}	88.08%	53.67%	57.39%
SOBlock	32	PGD^{20}	90.28%	58.21%	60.25%
ResNet10-TRADES ($1/\lambda = 1.0$)	32	PGD^{20}	81.52%	35.26%	57.07%
ResNet10-TRADES ($1/\lambda = 6.0$)	32	PGD^{20}	73.69%	43.46%	55.73%
SONet	64	PGD^{20}	89.36%	61.62%	64.08%
SOBlock	64	PGD^{20}	**91.57%**	**62.35%**	**64.70%**
ResNet10-TRADES ($1/\lambda = 1.0$)	64	PGD^{20}	82.74%	37.64%	58.97%
ResNet10-TRADES ($1/\lambda = 6.0$)	64	PGD^{20}	76.29%	45.24%	57.28%
SONet	32	$\text{PGD}^{1,000}$	88.08%	19.62%	31.75%
SOBlock	32	$\text{PGD}^{1,000}$	90.28%	52.01%	52.79%
ResNet10-TRADES ($1/\lambda = 1.0$)	32	$\text{PGD}^{1,000}$	81.52%	33.60%	56.70%
ResNet10-TRADES ($1/\lambda = 6.0$)	32	$\text{PGD}^{1,000}$	73.69%	43.30%	55.48%
SONet	64	$\text{PGD}^{1,000}$	89.36%	24.25%	39.79%
SOBlock	64	$\text{PGD}^{1,000}$	**91.57%**	**55.43%**	57.37%
ResNet10-TRADES ($1/\lambda = 1.0$)	64	$\text{PGD}^{1,000}$	82.74%	35.78%	**58.73%**
ResNet10-TRADES ($1/\lambda = 6.0$)	64	$\text{PGD}^{1,000}$	76.29%	44.70%	56.87%

5.2.1. BETTER ROBUSTNESS OF SONET IN NATURAL TRAINING THAN TRADES ADV-TRAINING IN PGD^{20} ATTACKS

Under PGD^{20} attacks, SONet with natural training achieves better robust accuracy than TRADES-adversarial training, without sacrificing natural accuracy. The robust accuracy against 20-step PGD attack is a common metric for evaluating ℓ_∞ adversarial robustness (Madry et al., 2018; Zhang et al., 2019c). We summarize the natural accuracy $\mathcal{A}_{\text{nat}}$ and robust accuracy $\mathcal{A}_{\text{rob}}$ under PGD adversarial attacks on different models in Table 1, where we use PGD_*^k to denote the k-step iterative PGD attack within $*$-norm box. The natural accuracy of SONet with 32-channel and 64-channel are 88.08% and 89.36% respectively, significantly better than that of ResNet10 with 32-channel and 64-channel deteriorates as 81.52% and 82.74% trained by TRADES ($1/\lambda = 1.0$). Under PGD_∞^{20} attack, our proposed SONet with 64-channel can achieve 61.62% robust accuracy, which is significantly better than the corresponding ResNet10 model with TRADES training (both $1/\lambda = 1.0$ and $1/\lambda = 6.0$).

We also evaluate both models against PGD_2^{20} (ℓ_2-norm $\epsilon = 0.5$) adversarial attack. We can observe that SONet is robust against PGD_2^{20} attack, and achieves better robust accuracy than ResNet10 trained by TRADES.

5.2.2. NONTRIVIAL ROBUSTNESS OF SONET UNDER $\text{PGD}^{1,000}$ ATTACKS

In addition to the PGD^{20} attack, we also conduct PGD attacks with more attack steps to better approximate the worst-case attacks. The robust accuracy of all the models decreases with more

attack steps (1,000 step), especially for ℓ_∞ attacks. However, SONet can still achieve 24.25% and 39.79% robust accuracy against $\text{PGD}_\infty^{1,000}$ and $\text{PGD}_2^{1,000}$ attacks, respectively. Such a decay is worse than TRADES-adversarial training that achieves robust accuracy at 44.70% ($1/\lambda = 6.0$) and 58.73% ($1/\lambda = 1.0$), but is still non-trivial. Therefore, adversarial robustness of SONet deteriorates but is still nontrivial under further iterative attack steps in PGD^{1000}.

However, a better performance can be achieved by SOBlock below.

5.2.3. IMPROVED ROBUSTNESS OF SOBLOCK AT THE FIRST LAYER THAN FULL SONET

A surprising observation is that by only adopting stablized neural ODE block in the first layer of ResNet10, SOBlock achieves even better performance than SONet that using all layers as such blocks. In Table 1, the natural accuracy of SOBlock with 32-channel and 64-channel are 90.28% and 91.57% respectively while maintains PGD_∞^{1000} robust accuracy with 52.01% and 55.43% respectively which is much higher than 43.30% and 44.70% achieved by TRADES ($1/\lambda = 6.0$).

SOBlock almost achieves the best performance among nearly in all settings, except for PGD_2^{1000} attack it has a comparative robust accuracy with TRADES. In addition to achieving such a high performance in accuracy, SOBlock particularly enjoys much less computational and memory cost than SONet, that is favoured in applications.

6. Gradient Masking Effect by Adaptive Stepsize

In this section, we further explore the fragility of the adversarial robustness of our proposed architecture under the variation of stepsize choices, robust gradient (CW) attacks, and gradient-free (SPSA) attacks. These results suggest that the adversarial robustness of (stablized) ODE block or net under PGD attacks is likely due to that adaptive stepsize in numerical integration has a gradient masking effect. Gradient masking (Papernot et al., 2017; Athalye et al., 2018) is a phenomenon widely associated with the obfuscation of gradient information in gradient based adversarial attacks, yet failure under robust gradient and gradient-free attacks, thus giving a false sense of adversarial robustness.

6.1. Robustness of SOBlock as a result from Adaptive Stepsize in Numerical Integration

To investigate the reason of adversarial robustness of SOBlock under PGD attacks, we conduct an ablation study on the influence of different order of numerical ODE solvers together with their choice of step size and error tolerance. We use WRN-34-10 as our base network for SOBlock in this section in order to obtain better comparison. The experiments below suggest that adversarial robustness of SOBlock comes from gradient masking effect of adaptive stepsize numerial ODE solvers, including adaptive Heun, Bosh3, and DOPRI5.

6.1.1. ADVERSIARIAL ROBUSTNESS IS ONLY ASSOCIATED WITH ADAPTIVE STEPSIZE ODE SOLVERS

In the first experiment, we compared three different choices of ODE solvers: Euler method (first order, fixed step size $h = 1$), RK4 (fourth order, fixed step size $h = 1$), adaptive Heun (second order, adaptive step size), Bosh3 (third order, adaptive stepsize) and DOPRI5 (fifth order, adaptive step size, with default error tolerance $\texttt{atol} = \texttt{rtol} = \texttt{tol} = 0.1$). In Table 2, it shows that all adaptive stepsize solvers (Heun, Bosh3, and DOPRI5) lead to adversarial robustness of SOBlock against

Table 2: Comparisons between SOBlock and ODENet with different solver ODE solvers under PGD adversarial attacks on CIFAR10 dataset.

Model	Solver	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$	
			PGD^{20}	PGD^{1000}
SOBlock	Euler	94.41%	0%	0%
SOBlock	$\text{RK4}_{3/8}$ rule	92.06%	0%	0%
SOBlock	Dopri5(tol=0.1)	94.22%	71.20%	63.20%
SOBlock	Dopri5(tol=0.01)	93.98%	64.66%	46.40%
SOBlock	Dopri5(tol=0.001)	94.32%	63.87%	46.20%
SOBlock	Bosh3(tol=0.1)	92.85%	66.00%	52.84%
SOBlock	Bosh3(tol=0.01)	92.30%	67.06%	59.74%
SOBlock	Bosh3(tol=0.001)	92.38%	65.03%	55.31%
SOBlock	Adaptive Heun(tol=0.1)	92.42%	61.16%	55.84%
SOBlock	Adaptive Heun(tol=0.01)	92.43%	63.95%	53.79%
SOBlock	Adaptive Heun(tol=0.001)	92.73%	57.68%	45.33%
ODENet	Euler	87.04%	0%	0%
ODENet	$\text{RK4}_{3/8}$ rule	87.78%	0%	0%
ODENet	Dopri5(tol=0.1)	87.41%	42.69%	13.14%
ODENet	Dopri5(tol=0.01)	87.46%	37.20%	8.36%
ODENet	Dopri5(tol=0.001)	87.54%	36.19%	7.75%

PGD attacks, while SOBlocks trained by fixed stepsize solvers Euler and RK4 totally fail under both PGD^{20} and PGD^{1000} in spite of high natural accuracy.

The same phenomenon persists when we change SOBlock to traditional ODENet (Chen et al., 2018) without using the skew-symmetric stabilization in (4). Altough ODENet slightly drops the natural accuracy as desired, one can see in Table 2 that robust accuracy of ODENet with both Euler and RK4 fixed step size solver training totally fails (0%), while adaptive step size solver like DOPRI5 shows nontrivial robust accuracy under PGD^{20} and PGD^{1000}.

Therefore, adversarial robustness of both ODENet and our stabilized ODE block/net is necessarily associated with the adaptive stepsize in numerical integrations, rather than the fixed stepsize.

6.1.2. ADAPTIVE STEP SIZES WITH LARGE ERROR TOLERANCE IN DOPRI5 ALLOWS GRADIENT MASKING

Both RK4 and DOPRI5 are high (fourth or fifth) order numerical ordinary differential equation methods, where DOPRI5 enjoys a simple error estimate for adaptive stepsize choice (Dormand and Prince, 1980). In DOPRI5, it uses six function evaluations to calculate both fourth- and fifth-order accurate solutions, whose difference is taken as the error estimate of the fourth-order solution. Adaptive stepsize is adopted in DOPRI5 when the error estimate is within the tolerance specified by absolute error and relative error tolerances $(\texttt{atol}, \texttt{rtol})$, both set to be $\texttt{tol}$ here (Hairer et al.,

2008):

$$\mathsf{err} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_{1i} - \widehat{y}_{1i}}{\mathrm{sc}_i}\right)^2}, \quad \mathrm{sc}_i = \mathtt{atol}_i + \max\left(|y_{0i}|, |y_{1i}|\right) \cdot \mathtt{rtol}_i.$$

Therefore in the second experiment, we investigate the influence of changing error tolerance `tol` in DOPRI5. The err is then compared to 1 in order to find an optimal choice, where an order $q = \min(p, \hat{p})$ solver may choose the optimal step size as $h_{\mathrm{opt}} = h \cdot (1/\mathsf{err})^{1/(q+1)}$. For example, DOPRI5 has $q = 4$, whence $h_{\mathrm{opt}} = h \cdot (1/\mathsf{err})^{1/5}$; adaptive Heun has order $q + 1 = 2$, and BOSH3 has order $q + 1 = 3$. Some care is now necessary for a good implementation: the formula above is multiplied by a safety factor safety, for example $\mathsf{safety} = 0.8, 0.9, (0.25)^{1/(q+1)}$, or $(0.38)^{1/(q+1)}$, so that the error will be acceptable the next time with high probability. Further, h is not allowed to increase nor to decrease too fast. For example, we may put

$$h_{\mathrm{new}} = h \cdot \min\left(\mathtt{ifactor}, \max\left(\mathsf{safety} \cdot (1/\mathsf{err})^{1/(q+1)}, \mathtt{dfactor}\right)\right)$$

for the new step size. Then, if $\mathsf{err} \leq 1$, the computed step is accepted and the solution is advanced with y_1 and a new step is tried with h_{new} as step size. Else, the step is rejected and the computations are repeated with the new step size h_{new} . The maximal step size increase $\mathtt{ifactor} = 10.0$ and the minimial step size decrease $\mathtt{dfactor} = 0.2$ by default.

In Table 2, one can see that reasonably large error tolerance in DOPRI5 increases the robustness of both SOBlock and ODENet, e.g. PGD_∞^{20}-robust accuracy at 71.20% at $\mathtt{tol} = 0.1$ against 63.87% at $\mathtt{tol} = 0.001$ for SOBlock and 42.69% at $\mathtt{tol} = 0.1$ against 36.19% at $\mathtt{tol} = 0.001$ for ODENet. Large error tolerance leads to large perturbations on adaptive stepsize in DOPRI5, e.g. Table 3 shows that increasing tolerance from $\mathtt{tol} = 0.001$ to $\mathtt{tol} = 0.1$ lead to enlarged adaptive stepsize perturbations from the order of $1e-3$ to $1e-2$.

These phenomena above show that adversarial robustness under PGD attacks is a result from the adaptive stepsize choice of numerical ODE solvers that perturbs the gradients of loss functions, where enlarging error tolerance properly may increase the robustness of SOBlocks and ODENets. Therefore, adaptive step size ODE solvers like DOPRI5 contribute such a kind of gradient masking against PGD attacks: reasonably large error tolerance in numerical function estimate leads to large perturbations of gradients that fools the projected gradient descent in attacks. We also note that over-enlarging error tolerance, especially in low order ODE solvers (adaptive Heun and Bosh3), may lead to inaccuracies in natural training that eventually drops robust accuracy as well. Hence one should expect a reasonable choice of error tolerance should depend on a trade-off between fitting accuracy and gradient masking effect.

6.2. Robust Gradient (CW) and Gradient-Free (SPSA) Attacks

To further justify our reasoning above that the adversarial robustness of SOBlock and SONet is due to the gradient masking effect of adaptive stepsize numerical ODE solvers, especially DOPRI5, we further conduct experiments under two sorts of new attacks, CW attack that has robust gradients due to the use of hinge loss and SPSA attack that is a kind of gradient-free attack.

Table 4 shows that in spite of the impressive robustness under PGD attacks, both SONet and SOBlock are vulnerable under CW_∞ and SPSA attacks, while ResNet10 trained with TRADES still has relatively strong robustness. Particularly, under CW_∞ attack, SOBlock (SONet) with 64 channels has 0% (11.20%) robust accuracy compared with ResNet10 in TRADES training at 39.77%

Table 3: Adaptive steps with ODENet under different dopri5 tolerances and PGD_∞ attack iterations

Solver	PGD iterations	Adaptive steps
Dopri5(tol=0.1)	1	[0.0, 0.262, 1.0]
	100	[0.0, 0.253, 1.0]
	1000	[0.0, 0.244, 1.0]
Dopri5(tol=0.01)	1	[0.0, 0.155, 0.827, 1.0]
	100	[0.0, 0.150, 0.793, 1.0]
	1000	[0.0, 0.149, 0.789, 1.0]
Dopri5(tol=0.001)	1	[0.0, 0.097, 0.423, 1.0]
	100	[0.0, 0.096, 0.420, 1.0]
	1000	[0.0, 0.094, 0.409, 0.981, 1.0]

Table 4: Comparisons between SONet, SOBlock with natural training and ResNet10 with TRADES under CW_∞ and SPSA adversarial attacks on CIFAR10 dataset.

Model	Channel	$\mathcal{A}_{\text{nat}}(f)$	$\mathcal{A}_{\text{rob}}(f)$	
			CW-Linf	SPSA
SONet	32	88.08%	0%	2.50%
SOBlock	32	**90.28%**	0%	7.64%
ResNet10-TRADES ($1/\lambda = 1$)	32	81.52%	37.61%	**68.30%**
ResNet10-TRADES ($1/\lambda = 6$)	32	73.69%	**38.92%**	63.60%
SONet	64	89.36%	11.20%	15.10%
SOBlock	64	**91.57%**	0%	11.68%
ResNet10-TRADES ($1/\lambda = 1$)	64	82.74%	35.78%	**69.97%**
ResNet10-TRADES ($1/\lambda = 6$)	64	76.29%	**39.77%**	65.97%

($1/\lambda = 6$); while under SPSA attack, SOBlock (SONet) of 64 channels has 11.68% (15.10%) robust accuracy compared with TRADES training at 69.97% ($1/\lambda = 1$). This provides a support of the gradient masking by DOPRI5, that fails to fool CW_∞ and SPSA attacks which are not as sensitive to gradients of cross entropy loss as PGD attacks. Hence the gradient masking of adaptive stepsize gives us a false sense of adversarial robustness in PGD attacks.

7. Conclusions

In this paper, we propose a stabilized neural ODE architecture based on a skew-symmetric dynamical system with provable Lyapunov stability. We show that such an ODE based network architecture can achieve some state-of-the-art adversarial robustness with natural training against PGD attacks, without sacrificing natural accuracy that is suffered by popular adversarial training methods such as TRADES. To understand this phenomenon, we further explore the possible mechanism underlying

such kind of adversarial robustness. We show that the adaptive stepsize numerial ODE solvers, such as adaptive HEUN2, BOSH3, and especially DOPRI5, have a gradient masking effect that fails the PGD attacks which are sensitive to gradient information of training loss, while they can not fool the CW attack of robust gradients and the SPSA attack that is gradient-free. This provides a new explanation that the adversarial robustness of ODE based networks is mainly due to the obfuscated gradients in numerical ODE solvers with adaptive step sizes.

Acknowledgments

This research made use of the computing resources of the X-GPU cluster supported by the Hong Kong Research Grant Council Collaborative Research Fund: C6021-19EF. The research of Yifei Huang and Yuan Yao is supported in part by HKRGC 16303817, ITF UIM/390, as well as awards from Tencent AI Lab and Si Family Foundation. Yaodong Yu and Yi Ma acknowledge support from ONR grant N00014-20-1-2002 and the joint Simons Foundation-NSFDMS grant 2031899. Hongyang Zhang was supported in part by the Defense Advanced Research Projects Agency under cooperative agreement HR00112020003. The views expressed in this work do not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred. Approved for public release; distribution is unlimited.

References

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems*, pages 12192–12202, 2019.

Karl Johan Aström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

Benny Avelin and Kaj Nyström. Neural ODEs as the deep limit of ResNets with constant weights. *arXiv preprint arXiv:1906.12183*, 2019.

Wieland Brendel, Jonas Rauber, Alexey Kurakin, Nicolas Papernot, Behar Veliqi, Sharada P Mohanty, Florian Laurent, Marcel Salathé, Matthias Bethge, Yaodong Yu, et al. Adversarial vision challenge. In *The NeurIPS'18 Competition*, pages 129–153. Springer, 2020.

M. Frank Callier and A. Charles Desoer. *Linear System Theory*. Springer-Verlag, 1991.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11190–11201, 2019.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, 2017.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, 2019.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In *International Conference on Artificial Intelligence and Statistics*, 2018.

J. R. Dormand and P. J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19—-26, 1980.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

Chao Gao, Jiyi Liu, Yuan Yao, and Weizhi Zhu. Robust estimation and generative adversarial nets. In *International Conference on Learning Representations (ICLR), New Orleans, Louisiana, United States*. 2019. arXiv preprint arXiv:1810.02030.

Chao Gao, Yuan Yao, and Weizhi Zhu. Generative adversarial nets for robust scatter estimation: A proper scoring rule perspective. *Journal of Machine Learning Research*, 21(160):1–48, 2020. arXiv preprint arXiv:1903.01944.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34 (1):014004, 2017.

E. Hairer, S.P. Norsett, and G. Wanner. *Solving Odinary Differential Equations I: Nonstiff Problems*. Springer-Verlag, Berlin, Heidelberg, 2008.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing*, 2017.

J Zico Kolter and Eric Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, 2018.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. On norm-agnostic robustness of adversarial training. *arXiv preprint arXiv:1905.06455*, 2019.

Mingjie Li, Lingshen He, and Zhouchen Lin. Implicit euler skip connections: Enhancing adversarial robustness via numerical stability. In *International Conference on Machine Learning*, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Dongyu Meng and Hao Chen. MagNet: a two-pronged defense against adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147, 2017.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

Haifeng Qian and Mark N. Wegman. l_2-nonexpansive neural networks. In *International Conference on Learning Representations*, 2019.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems 31*, pages 5019–5031, 2018.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540, 2016.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

Jan Svoboda, Jonathan Masci, Federico Monti, Michael Bronstein, and Leonidas Guibas. Peernets: Exploiting peer wisdom against adversarial attacks. In *International Conference on Learning Representations*, 2019.

Matthew Thorpe and Yves van Gennip. Deep limits of residual neural networks. *arXiv preprint arXiv:1810.11741*, 2018.

Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

Bao Wang, Binjie Yuan, Zuoqiang Shi, and Stanley J. Osher. Resnets ensemble via the feynman-kac formalism to improve natural and robust accuracies. In *Advances in Neural Information Processing Systems 32 (NIPS 2019)*, 2019.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *International Conference on Computer Vision*, 2017.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

Hanshu Yan, Jiawei Du, Vincent YF Tan, and Jiashi Feng. On robustness of neural ordinary differential equations. *arXiv preprint arXiv:1910.05513*, 2019.

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Painless adversarial training using maximal principle. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019a.

Hongyang Zhang, Junru Shao, and Ruslan Salakhutdinov. Deep neural networks with multi-branch architectures are intrinsically less non-convex. In *International Conference on Artificial Intelligence and Statistics*, pages 1099–1109, 2019b.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019c.

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4488, 2016.