Defending Neural ODE Image Classifiers from Adversarial Attacks with Tolerance Randomization[⋆]

Fabio Carrara[1] [0000−0001−5014−5089], Roberto Caldelli[2,3] [0000−0003−3471−1196], Fabrizio Falchi[1] [0000−0001−6258−5313], and Giuseppe Amato[1] [0000−0003−0171−4315]

[1] ISTI CNR, Pisa, Italy
{fabio.carrara,fabrizio.falchi,giuseppe.amato}@isti.cnr.it
[2] CNIT, Florence, Italy roberto.caldelli@unifi.it
[3] Universitas Mercatorum, Rome, Italy

Abstract. Deep learned models are now largely adopted in different fields, and they generally provide superior performances with respect to classical signal-based approaches. Notwithstanding this, their actual reliability when working in an unprotected environment is far enough to be proven. In this work, we consider a novel deep neural network architecture, named Neural Ordinary Differential Equations (N-ODE), that is getting particular attention due to an attractive property — a test-time tunable trade-off between accuracy and efficiency. This paper analyzes the robustness of N-ODE image classifiers when faced against a strong adversarial attack and how its effectiveness changes when varying such a tunable trade-off. We show that adversarial robustness is increased when the networks operate in different tolerance regimes during test time and training time. On this basis, we propose a novel adversarial detection strategy for N-ODE nets based on the randomization of the adaptive ODE solver tolerance. Our evaluation performed on standard image classification benchmarks shows that our detection technique provides high rejection of adversarial examples while maintaining most of the original samples under white-box attacks and zero-knowledge adversaries.

Keywords: Neural Ordinary Differential Equation · Adversarial Defense · Image Classification

1 Introduction

The astonishing success of deep learned models and their undeniable performances in a variety of difficult tasks (e.g. visual and auditory perception, natural language processing, self-driving cars, multimedia analysis) is still accompanied by the presence of several flaws and drawbacks. In fact, when neural networks

[⋆] This work was partially funded by Tuscany POR FSE 2014-2020 AI-MAP (CNR4C program, CUP B15J19001040004), the AI4EU project (EC, H2020, n.825619) and the AI4Media Project (EC, H2020, n. 951911).

are called to operate in an unprotected environment, as it could happen in multimedia forensic applications, they have shown important vulnerabilities. Such weaknesses can be exploited by an attacker through the design of ad-hoc adversarial manipulations in order to induce the model into a wrong evaluation. Depending on the specific application scenario we are dealing with in the real world, such an incorrect prediction can be crucial for the consequent choice, action or decision to be taken. In particular, in the context of image classification, a popular task on which we focus also our attention, an adversary can control and mislead a deep neural network classifier by introducing a small malicious perturbation in the input image [23].

Thanks to the florid research community interested in the subject, this phenomenon has been vastly analyzed on several neural network architectures on multiple tasks. While attacking a deep model seems to be easy due to the differentiability and complexity of deep models — indeed, many successful adversarial generation approaches exist [8, 19, 3], — counteracting this phenomenon and defending from attacks is still an open problem. Multiple approaches aiming at strengthening the attacked model [13, 21] achieve robustness to weak or unknowing adversaries, but stronger attacks usually are able to mislead also enhanced models, as currently, adversarial examples appear to be an intrinsic property of every common deep learning architecture.

In this work, we analyze the phenomenon of strong adversarial examples in Neural Ordinary Differential Equation (N-ODE) networks — a recent deep learning model that generalizes deep residual networks and is based on solutions of parametric ODEs. Among its properties, we find the ability to tune at test-time the precision-efficiency trade-off of the network by changing the tolerance of the adaptive ODE solver used in the forward computation. Previous work [5] showed that neural ODE nets are more robust to PGD attacks than standard architectures such as ResNets, and most importantly, higher tolerance values — i.e. lower-precision higher-efficiency regimes — provided increased robustness at a negligible expense of accuracy of the model. Here, we follow up by analyzing whether the same phenomena occur to ODE nets under the stronger Carlini&Wagner attack. Additionally, we test the attack performance when using different values of the solver tolerance during the adversarial generation and the prediction phase. Based on our findings, we also propose a simple adversarial detection approach based on test-time tolerance randomization that we evaluate on image classification benchmarks under the assumption of a zero-knowledge adversary, i.e. when the attacker has access to the model but does not know about the deployed defense.
The contributions of the present work are the following:

- we analyze neural ODE image classifiers under the Carlini&Wagner attack;
- we study how their robustness change when varying the ODE solver tolerance;
- we propose a novel test-time tolerance randomization approach for ODE nets based on a majority-voting ensemble to detect adversarial examples, and we evaluate it on standard benchmarks.

After this introduction, Section 2 refers to works related to ours, and Section 3 briefly introduce background knowledge on neural ODE nets and the adopted Carlini&Wagner adversarial generation algorithm. In Section 4, we discuss the robustness to adversarial samples of neural ODEs in relation with the ODE solver tolerance, and we propose our novel detection scheme. In Section 5, we describe the experimental evaluation[4], and Section 6 discusses results. Section 7 concludes the paper and lays out future research directions.

2 Related Work

The vulnerability of adversarial examples poses major challenges to security- and safety-critical applications, e.g. malware detection, autonomous driving cars, biometrical access control, and thus it is studied diffusely in the literature. Most analyses of deep models focus on deep convolutional networks image classifiers [23, 20, 14] under a variety of attacks, such as PGD [17] or the stronger CW [3]. This sprouted a huge offer of defensive methodologies against adversarial samples in this scenario, such as model enhancing via distillation [21] and adversarial sample detection via statistical methods [9] or auxiliary models [18, 4]. Among them, most promising methods are based on the introduction of randomization in the prediction process [24, 1]. Feinman et al. [7] propose a detection scheme based on randomizing the output of the network using dropout that mostly relate with the rationale of our proposed detection method. Both their and our methods are based on stochasticity of the output, that has been proven a powerful defense [2].

Regarding analyzing and defending neural ODE architectures, few works in the current literature cover the subject. Seminal works include Carrara et al. [5], that analyzes ODE nets under PGD attacks and asses their superior robustness with respect to standard architectures, and Hanshu et al. [10], that proposes a regularization based on the time-invariance property of steady states of ODE solutions to further improve robustness. Relevant to our proposed scheme is also the work of Liu et al. [16] that exploit stochasticity by injecting noise in the ODE to increase robustness to perturbations of initial conditions, including adversarial ones.

3 N-ODE Nets and Carlini&Wagner attack

In this section, we briefly introduce the Neural Ordinary Differential Equation (N-ODE) networks and the Carlini&Wagner adversarial attack we adopted in this work.

[4] Code and resources to reproduce the experiments presented here are available at `https://github.com/fabiocarrara/neural-ode-features/tree/master/adversarial`

3.1 Neural ODE Networks

In this section, we provide a basic description of Neural ODE (*Ordinary Differential Equations*) and an overview of their main properties. For a more detailed discussion on neural ODEs, the interested reader can refer to [6].

A neural ODE Net is a parametric model which includes an *ODE block*. The computation of such a block is defined by a parametric ordinary differential equation (ODE) whose solution gives the output result. We indicate with $\mathbf{h}_0$ the input of the ODE block coinciding with the initial state at time t_0 of the initial-state ODE written in Equation (1):

$$\begin{cases} \frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} = f(\mathbf{h}(t), t, \theta) \\ \mathbf{h}(t_0) = \mathbf{h}_0 \end{cases} . \tag{1}$$

The function $f(\cdot)$, which depends on the parameter θ, defines the continuous dynamic of the state $\mathbf{h}(t)$. By integrating the ODE (see Equation (2)), the output of the block $\mathbf{h}(t_1)$ at a time $t_1 > t_0$ can be obtained.

$$\mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} \frac{\mathrm{d}\mathbf{h}(t)}{\mathrm{d}t} \mathrm{d}t = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) \mathrm{d}t \, . \tag{2}$$

The above integral can be computed with standard ODE solvers, such as Runge-Kutta or Multi-step methods. Thus, the computation performed by the ODE block can be formalized as a call to a generic ODE solver

$$\mathbf{h}(t_1) = \mathrm{ODESolver}(f, \mathbf{h}(t_0), t_0, t_1, \theta) \, . \tag{3}$$

Generally, in image classification applications, the function $f(\cdot)$ is implemented by means of a small trainable convolutional neural network. During the training phase, the gradients of the output $\mathbf{h}(t_1)$ with respect to the input $\mathbf{h}(t_0)$ and the parameter θ can be obtained using the adjoint sensitivity method. This consists of solving an additional ODE in the backward pass. Once the gradient is obtained, standard gradient-based optimization can be applied.

ODE Nets present diverse peculiar properties determined by their intrinsic structure; one of these, of particular interest for our case, concerns the *accuracy-efficiency trade-off* which is tunable at inference time by controlling the tolerance parameter of adaptive ODE solvers.

The ODE net image classifier (*ODE*) we consider in this work (see Figure 1 bottom part) is constituted by an ODE block (implemented as Equation (3)), responsible for the whole feature extraction chain, preceded by a limited pre-processing stage comprised of a single K-filter 4x4 convolutional layer with no activation function that linearly maps the input image in an adequate state space. The $f(\cdot)$ function in the ODE block is implemented as a standard residual block used in ResNets (described below). After the ODE block, the classification step is implemented with a global average-pooling operation followed by a single fully-connected layer with softmax activation. In addition to this, we consider also a classical ResNet *(RES)* (Figure 1 top part) as baseline [6] in comparison

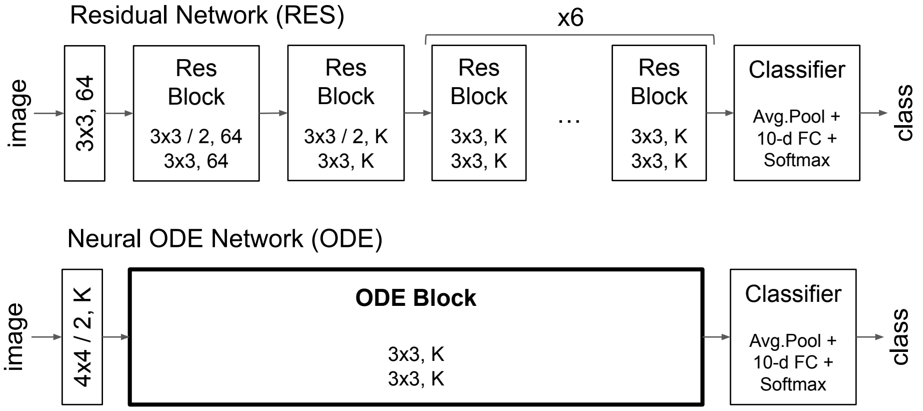


Fig. 1: Convolutional layers are written in the format *kernel width* × *kernel height [/ stride], n. filters*; padding is always set to 1. For MNIST, $K = 64$, and for CIFAR-10, $K = 256$.

with ODE-Nets. It is composed of a 64-filter 3x3 convolutional layer and 8 residual blocks. Each residual block follows the standard formulation defined in [11], where Group Normalization [25] is used instead of batch one. The sequence of layers comprising a residual block is *GN-ReLU-Conv-GN-ReLU-Conv-GN* where *GN* stands for a Group Normalization with 32 groups and *Conv* is a 3x3 convolutional layer. The first two blocks downsample their input by a factor of 2 using a stride of 2, while the subsequent blocks maintain the input dimensionality. Only the first block uses 64-filters convolutions while the subsequent ones employ K-filter convolutions where K varies with the specific dataset; the final classification step is implemented as before.

3.2 The Carlini&Wagner attack

In this subsection, we briefly introduce the *Carlini&Wagner (CW)* attack [3] that has been used in our work to test and evaluate the robustness of the ODE-Net to adversarials samples. It is currently deemed as one of the strongest available adversary techniques to attack neural networks designed for image classification task. It exists in three versions, according to the metric adopted to measure the perturbation; in our implementation, we have considered the CW-L_2 which is formalized as in Equation (4):

$$\min\left(c \cdot g\left(\mathbf{x}^{\text{adv}}\right) + \left\|\mathbf{x}^{\text{adv}} - \mathbf{x}\right\|_2^2\right) \tag{4}$$

with

$$g(\mathbf{x}^{\mathrm{adv}}) = \max\left(\max_{i \neq t} Z(\mathbf{x}^{\mathrm{adv}})_i - Z(\mathbf{x}^{\mathrm{adv}})_t, -\kappa\right) , \tag{5}$$

$$\mathbf{x}^{\mathrm{adv}} = \frac{\tanh(\mathbf{w}) + 1}{2} \tag{6}$$

where $g(\cdot)$ is the objective function (misclassification), $\mathbf{x}^{\mathrm{adv}}$ is the adversarial example in the pixel space, and $\mathbf{w}$ is its counterpart in the tanh space in which the optimization is carried out. $Z(\cdot)$ are the logits of a given input, t is the target class, κ is a parameter that allows adjusting the confidence with which the misclassification occurs, and c is a positive constant whose value is set by exploiting a binary search procedure. The rationale of the attack is to minimize at each iteration the highest confidence among non-target classes (first term of Equation (4)) while keeping the smallest possible perturbation (second term). It is worth of mention the use of the term $\tanh(\mathbf{w})$ that represents a change of variable that allows one to move from the pixel to the tanh space. This helps regularizing the gradient in extremal regions of the perturbation space thus facilitating optimization with gradient-based optimizers.

4 The proposed decision method based on tolerance randomization

In this section, we propose a new method to provide robustness to ODE-nets against the *Carlini&Wagner (CW)* attack. In sub-section 4.1, we present the idea to use a varying test-time tolerance to counteract adversarial perturbations, while in sub-section 4.2, we propose an innovative approach which resorts to tolerance randomization to detect adversarial samples.

4.1 On tolerance variation

The *CW* attack is considered so far as one of the strongest adversarial algorithms to fool neural networks in image classification specifically. In Figure 2, we report some adversarial examples generated with the CW attack for two well-known image datasets (MNIST and CIFAR-10, see sub-section 5.1 for details on the datasets). Though ODE-Nets are very promising and show good performances, they are prone to be attacked as well as the other kinds of networks [5]. This can be appreciated in Table 1, where for each model and dataset, we report the classification error, the attack success rate (in percentage) the mean L_2 norm of the adversarial perturbation. Note that the basic behaviour of both models is similar: they show a limited error rate on original images but, on the contrary, the CW attack achieve a very high attack success rate. For ODE nets, it is worth noting that when we increase the value of the tolerance τ used at test time and by the attacker ($\tau_{\mathrm{test}} = \tau_{\mathrm{attack}}$), the classification error rate is rather stable, but

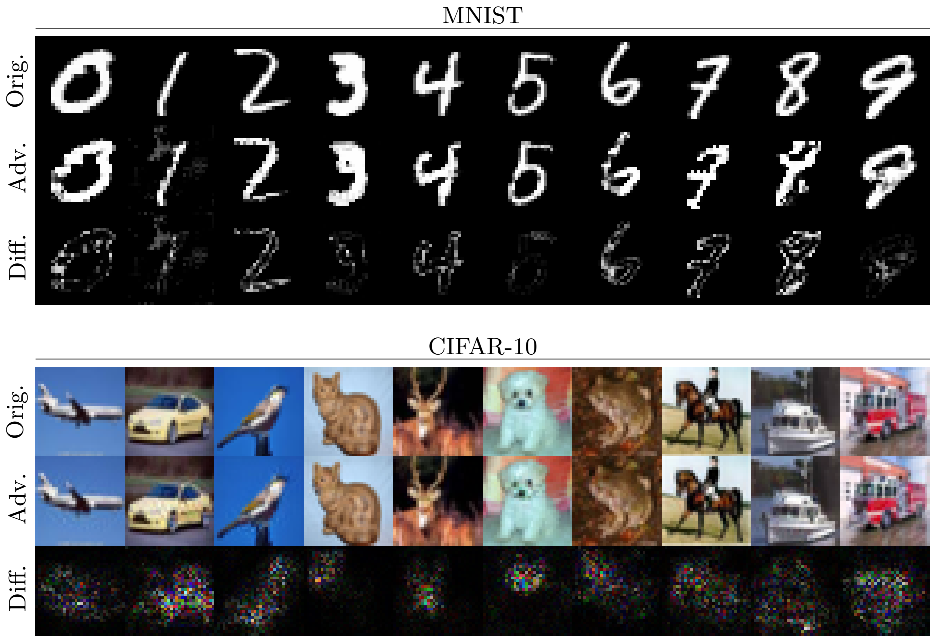


Fig. 2: Adversarial examples found with the Carlini&Wagner attack on our neural ODE network. Adversarial perturbations (Diff.) of CIFAR-10 samples have been amplified by a factor 10 for visualization purposes. Best viewed in electronic format.

Table 1: Classification error, attack success rate, and mean L_2 norm perturbation of RES and ODE on MNIST and CIFAR-10 test sets; for ODE, we report quantities varying the test-time adaptive solver tolerance τ ($\tau_{\text{attack}} = \tau_{\text{test}}$).

		ODE ($\tau_{\text{attack}} = \tau_{\text{test}}$)				
MNIST	RES	10^{-4}	10^{-3}	10^{-2}	10^{-1}	10^{0}
Classification Error (%)	0.4	0.5	0.5	0.6	0.8	1.2
CW Attack Success Rate (%)	99.7	99.7	90.7	74.4	71.6	69.7
Mean L_2 Perturb. $(\times 10^{-2})$	1.1	1.4	1.7	1.9	1.7	1.9
CIFAR-10						
Classification Error (%)	7.3	9.1	9.2	9.3	10.6	11.3
CW Attack Success Rate (%)	100	100	100	100	100	100
Mean L_2 Perturb. $(\times 10^{-5})$	2.6	2.2	2.4	4.1	8	13.7

the required attack budget increases; this is quite clear for the MNIST dataset where the attack success rate quickly decreases, but it can also be perceived for CIFAR-10 by looking at the mean perturbation introduced by the attack: the attack success rate continues to be 100%, but a superior cost is paid in terms of applied distortion. While this witnesses again the strength of the CW attack, on the other hand, it confirms that the sensibility to the tolerance variations, found in the case of the Projected Gradient Descent (PGD) attack [5], is also shown by the CW attack. This aspect will be further debated and supported with experimental results in Section 6.

4.2 Tolerance randomization to detect adversarial samples

Starting from such findings, we investigated in depth such phenomenon, and we exploited it to propose a novel adversarial detection methodology. It has been considered that the attacker operates in a white-box scenario; consequently, he has at his disposal the trained ODE-Net and knows the parameter setting of the classifier. An attack can be deemed as successful if the CW algorithm is able to find an adversarial sample leading to a misclassification without exceeding a prefixed attack budget defined as the maximum number of optimization iterations. On the other side, the analyst has to perform image classification of the adversarial images generated by the CW-attacker by using the same classifier (the ODE-Net); so, if the attack has been successful, the analyst has not any chances not to run into a misclassification error.

On this basis, we proposed in this work a new detection strategy based on ODE-Net *tolerance randomization*: the rationale is to randomize the test-time solver tolerance τ_{test} by sampling it uniformly from a range centered on τ_{train} such that $\tau_{\text{train}} = \tau_{\text{attack}} \neq \tau_{\text{test}}$. This would allow to decouple, to some extent, the attack context to the testing one, and it should induce robustness in the capacity of the network not to be misled. Introducing stochasticity also helps the defendant against knowledgeable adversaries, as simply changing τ_{test} to a different fixed value can be easily counteracted by the adversary also changing τ_{attack} to the new value. The developed experimental tests (see Section 6.1) confirm that by using this approach the CW attack success rate can be diminished while maintaining a low classification error on pristine images.

On such a basis, we propose to ensemble several predictions with different randomly drawn test-time tolerance parameters τ_{test} to detect whether the classification system is subjected to an adversarial sample (created by CW attack in this case). By indicating with V the number of voting members (i.e. the number of τ values randomly drawn) belonging to the ensemble, we will declare that an adversarial sample is detected if $v_{\text{agree}} < v_{\text{min}}$, where v_{agree} is the largest amount of members that have reached the same decision on the test image (size of the majority) and v_{min} is the minimum consensus threshold required for assessing the authenticity (non-maliciousness) of the input. According to our experiments (see sub-section 6.2 for details), such a strategy can grant a significant improvement in detecting CW-generated adversarial images while obtaining a very low rejection of original ones.

5 Experimental Setup

In this section, we present the experimental set-up adopted to analyse and evaluate how the introduction of tolerance randomization during testing can improve the robustness to adversarial examples (CW generated) in the proposed scenario.

5.1 Datasets: MNIST and CIFAR-10

All the models used in this analysis have been trained on two standard and well-known image classification benchmarks: MNIST [15] and CIFAR-10 [12]. MNIST is composed by 60,000 images subdivided into training (50,000) and testing (10,000) sets; images are grayscale having a size of 28x28 pixels and represent hand-written digits (from 0 to 9, so it consists of 10 classes). MNIST is substantially the *de facto* standard baseline for novel machine learning algorithms and is nearly the only dataset used in most research concerning ODE nets. The second dataset has also been taken into account in our analysis is CIFAR-10; it is a 10-class image classification dataset too, comprised of again 60,000 RGB images (size 32x32 pixels) of common objects subdivided in training/testing sets (50,000/10,000).

5.2 Details on training

Both considered models, RES and ODE, adopt a dropout, applied before the fully-connected classifier, with a drop probability of 0.5, while the SGD optimizer has a momentum of 0.9; the weight decay is 10^{-4}, batch size is 128 and learning rate is 10^{-1} reduced by a factor 10 every time the error plateaus. The number of filters K in the internal blocks is set to 64 for MNIST and 256 for CIFAR-10 respectively. For the ODE net model, containing the ODE block, we used the Dormand–Prince variant of the fifth-order Runge–Kutta ODE solver[5]; in such algorithm, the step size is adaptive and can be controlled by a tolerance parameter τ ($\tau_{\text{train}} = 10^{-3}$ has been set in our experiments during the training phase). The value of τ constitutes a threshold for the maximum absolute and relative error (estimated using the difference between the fourth-order and the fifth-order solution) tolerated when performing a step of integration; if such a step error exceeds τ, the integration step is discarded and the step size decreased. Both models, RES and ODE, achieved classification performances comparable with the current state of the art on MNIST and CIFAR-10 datasets (see Table 1).

5.3 Carlini&Wagner attack implementation details

We employ Foolbox 2.0 [22] to perform CW attacks on PyTorch models. We adopt Adam to optimize Eq. (4) setting the maximum iterations to 100 and performing 5 binary search steps to tune c starting from 10^{-2}. We adopt a

[5] implemented in `https://github.com/rtqichen/torchdiffeq`

learning rate of 0.05 for MNIST and 0.01 for CIFAR-10. As pristine samples to perturb, we select the first 5,000 images of each test set, discarding the images naturally misclassified by the classifier.

6 Experimental Results

In this section, we present and discuss some of the experimental results carried out to investigate the behavior of the ODE Nets against the Carlini&Wagner attack. In sub-section 6.1, we report results obtained by varying the tolerance τ at test-time, while in sub-section 6.2, based on such findings, we propose a new strategy to detect that the ODE classifier is under adversarial attack.

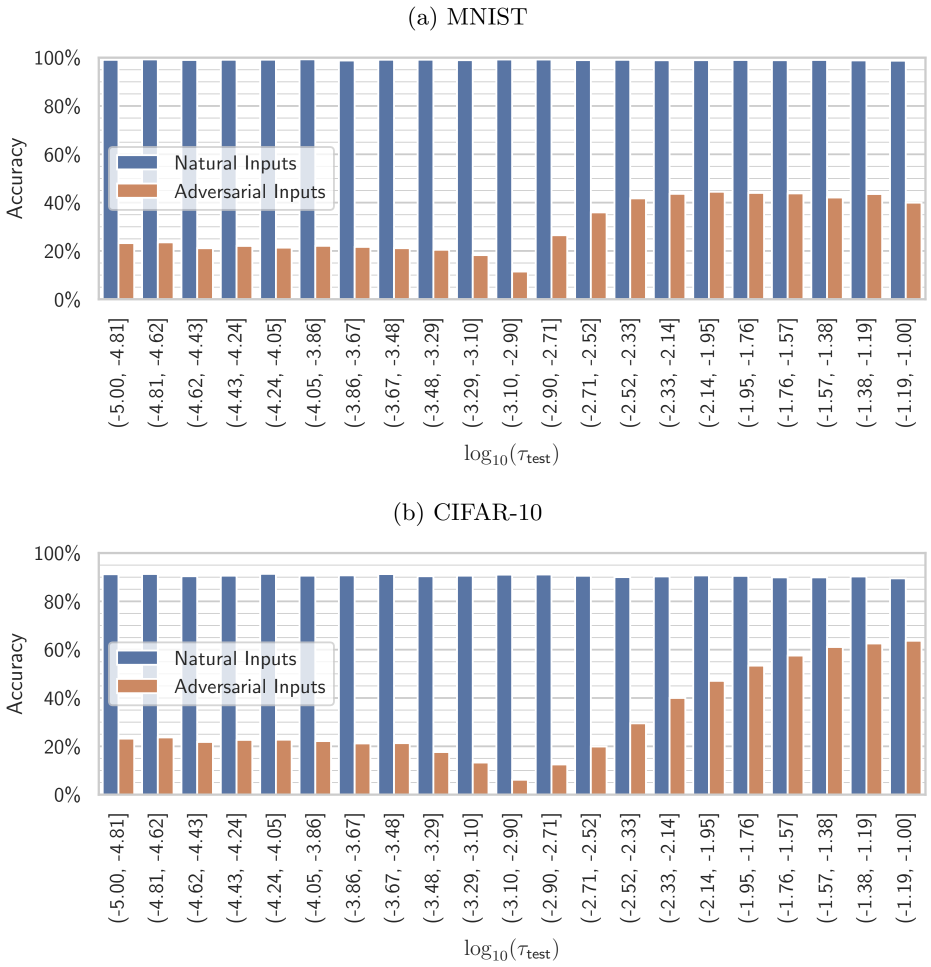
6.1 Results varying the tolerance at test-time

To better understand how the tolerance τ impacts on the classification of the adversarial samples generated by means of CW attack, we varied its value at test time. We assumed that the ODE-Net has been trained at $\tau_{\text{train}} = 10^{-3}$ that provides a well-balanced trade-off between accuracy and computational cost of the network. Consequently, in our scenario assumptions, this is also the value that the CW-attacker would use ($\tau_{\text{attack}} = \tau_{\text{train}}$). At prediction time, the tolerance is drawn from a log-uniform distribution with the interval $[10^{-5}, 10^{-1}]$ centered in $\tau_{\text{train}} = 10^{-3}$; 20 values are sampled for each image to be classified. In Figure 3, results obtained in terms of accuracy of the ODE-Net classifier on original inputs (blue bars) and adversarial examples (orange bars) are pictured respectively for MNIST and CIFAR-10 datasets; the tolerance, on x-axis, is binned (21 bins) in the log space.

It is evident that accuracy on natural inputs (blue bars) is always stable and also very high for each tolerance value, averagely around the original network accuracy (100% for MNIST and 90% for CIFAR-10). On the contrary, accuracy on CW-created adversarials inputs (orange bars) is quite poor (this demonstrates again the power of such a technique), but it is very interesting to note that in the central bin (around $\tau_{\text{train}} = 10^{-3}$) the attack has the highest effectiveness: this seems to mean that when the tolerance at test-time coincides with that adopted by the CW-attacker the classifier is strongly induced into a misclassification.

Furthermore, it can also be appreciated that if τ at test-time is moved away from the central value used by the CW-attacker, accuracy increases. This means that changes in the tolerance provide robustness to CW attack achieving, for instance, an accuracy on adversarial inputs of about 60% (with a corresponding accuracy around 90% on original images) for CIFAR-10 dataset (see Figure 3b on the extreme right).

Finally, it worthy observing that the trend of growth of the orange bars is asymmetric with respect to the central value $\tau_{\text{train}} = 10^{-3}$, and it achieves higher values on the right side: this once more witness that, as expected, increasing the tolerance allows to gain in robustness as general.



(a) MNIST

(b) CIFAR-10

Fig. 3: Robustness vs test-time solver tolerance τ_{test}. For each image, we sampled 20 values for τ from a log-uniform distribution within the $[10^{-5}, 10^{-1}]$ interval. We report the mean accuracy of the ODENet classifier on natural and adversarial examples for each tolerance bin (in log space).

6.2 Results on detection of adversarial samples

In this section, we present the experiments used to verify the method proposed in sub-section 4.2 based on tolerance randomization.

Looking at Figure 4, we can see that, once established the size V of the voting ensemble (different colored lines), by varying the threshold $v_{\min}$ with a step of 1, ROC curves can be obtained in terms of TPR versus FPR, where true positive indicate the correct classification of a natural input. Such graphs clearly demonstrate that high TPRs can be registered in correspondence of quite limited FPRs. This is particularly visible for MNIST dataset (see Figure 4a), but it is still true for CIFAR-10; if, just for example, we refer to Figure 4b when $V = 20$ (purple line), by increasing the value of $v_{\min}$ (going down along the curve), we can reduce the FPR while maintaining an extremely high TPR: with $v_{\min}$=20 a TPR=92.5% and a corresponding FPR=15% are achieved (see bottom-left corner of Figure 4b).

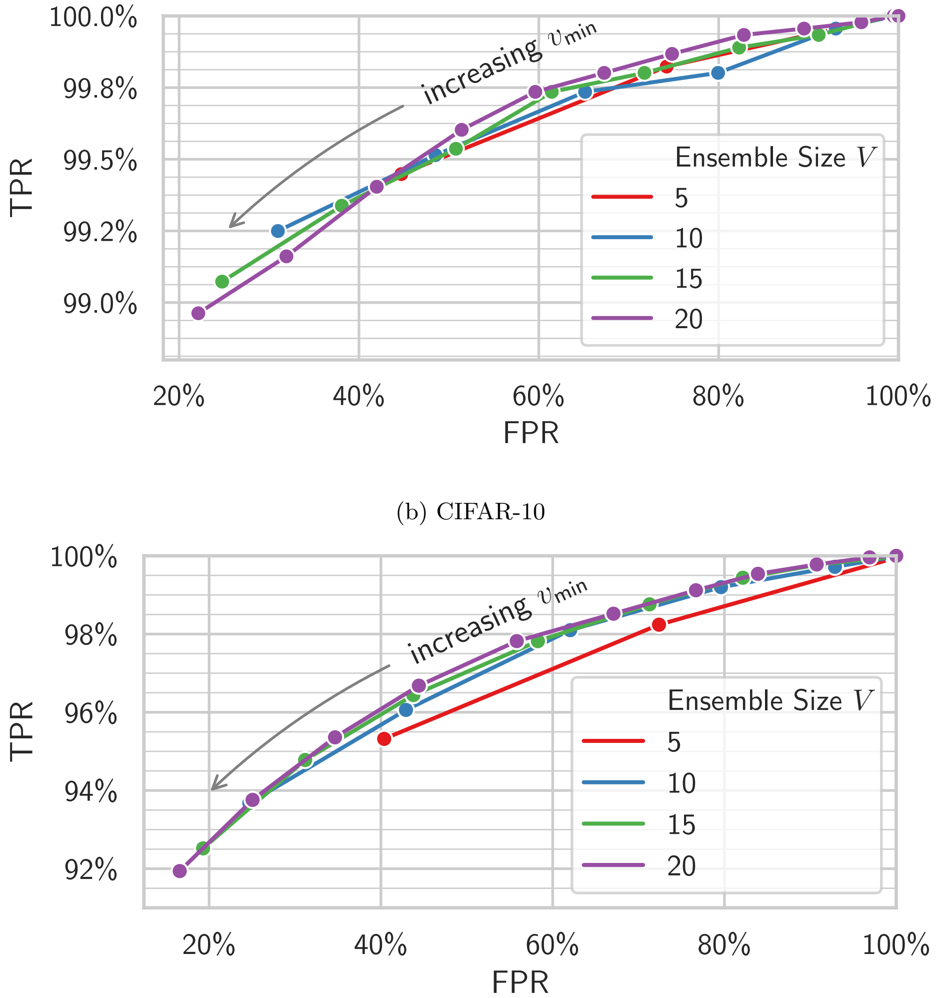
This experiment basically demonstrates that if the ODE-Net is subjected to a zero-knowledge *Carlini&Wagner* attack in a white-box scenario, by resorting at test-time tolerance randomization, it is possible both to preserve classification performances on natural images and significantly reduce the capacity of the CW attack to fool the ODE classifier.

7 Conclusions and Future Work

In this paper, we analyze the robustness of neural ODE image classifiers in an uncontrolled environment. In particular, we pay attention to the behavior of N-ODE nets against the Carlini&Wagner (CW) attack which is deemed so far as one of the most performing adversarial attacks to the task of image classification. We focus on how the tolerance of the adaptive ODE solver — used in neural ODE nets to tune the computational precision-efficiency trade-off — affects the robustness against such attacks. We observe that deviating the tolerance used in prediction from the one used when generating adversarial inputs tends to undermine attacks while maintaining high accuracy on pristine samples. On this basis, we propose a novel adversarial detection strategy for ODE nets based on tolerance randomization and a major voting ensemble scheme.

Our evaluation performed on standard image classification benchmarks shows that our simple detection technique is able to reject roughly 80% of strong CW adversarial examples while maintaining +90% of original samples under white-box attacks and zero-knowledge adversaries. Moreover, we deem that the stochasticity in our method introduces difficulties also for knowledgeable adversaries. We hypothesize that to bypass our method the adversary should require high attack budgets to attack a wide range of tolerance values and distill them in a unique malicious input.

In future work, we plan to devise an attack strategy for our proposed detection method to evaluate it in more stringent attack scenarios. Moreover, we plan to extensively explore the tolerance space also from an attacker perspective.



(a) MNIST

(b) CIFAR-10

Fig. 4: Detection performance of the randomized tolerance ensemble. We show ROC curves (TPR vs FPR, where TP = "correctly detected natural input" and FP = "adversarial input misdetected as natural") obtained varying the minimum majority size $v_{\min}$, i.e. if the number of majoritarian votes v_{agree} in the ensemble is greater than $v_{\min}$, the input is considered authentic (positive), otherwise adversarial (negative).

References

1. Barni, M., Nowroozi, E., Tondi, B., Zhang, B.: Effectiveness of random deep feature selection for securing image manipulation detectors against adversarial examples. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2977–2981. IEEE (2020)
2. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 3–14. AISec '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3128572.3140444
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE SP. pp. 39–57. IEEE (2017)
4. Carrara, F., Becarelli, R., Caldelli, R., Falchi, F., Amato, G.: Adversarial examples detection in features distance spaces. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
5. Carrara, F., Caldelli, R., Falchi, F., Amato, G.: On the robustness to adversarial examples of neural ODE image classifiers. In: 2019 IEEE WIFS. pp. 1–6. IEEE (2019)
6. Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: Advances in Neural Information Processing Systems. pp. 6572–6583 (2018)
7. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. CoRR **abs/1703.00410** (2017)
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
9. Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P.D.: On the (statistical) detection of adversarial examples. CoRR **abs/1702.06280** (2017)
10. Hanshu, Y., Jiawei, D., Vincent, T., Jiashi, F.: On robustness of neural ordinary differential equations. In: International Conference on Learning Representations (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV. pp. 630–645. Springer (2016)
12. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)
13. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: ICLR Workshops (2017), `https://openreview.net/forum?id=HJGU3Rodl`
14. Kurakin, A., Goodfellow, I.J., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.L., Huang, S., Zhao, Y., Zhao, Y., Han, Z., Long, J., Berdibekov, Y., Akiba, T., Tokui, S., Abe, M.: Adversarial attacks and defences competition. CoRR **abs/1804.00097** (2018)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
16. Liu, X., Xiao, T., Si, S., Cao, Q., Kumar, S., Hsieh, C.J.: Stabilizing neural ode networks with stochasticity (2019)
17. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
18. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. In: ICLR (2017)

19. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: IEEE CVPR. pp. 2574–2582 (2016)
20. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: Security and Privacy (EuroS&P), 2016 IEEE European Symposium on. pp. 372–387. IEEE (2016)
21. Papernot, N., McDaniel, P.D., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE SP 2016. pp. 582–597 (2016). https://doi.org/10.1109/SP.2016.41
22. Rauber, J., Brendel, W., Bethge, M.: Foolbox: A python toolbox to benchmark the robustness of machine learning models. In: Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning (2017)
23. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014)
24. Taran, O., Rezaeifar, S., Holotyak, T., Voloshynovskiy, S.: Defending against adversarial attacks by randomized diversification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11226–11233 (2019)
25. Wu, Y., He, K.: Group normalization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)