Gradient Obfuscation Checklist Test Gives a False Sense of Security

Nikola Popovic[1], Danda Pani Paudel[1], Thomas Probst[1], Luc Van Gool[1,2]
[1]Computer Vision Laboratory, ETH Zurich, Switzerland
[2]VISICS, ESAT/PSI, KU Leuven, Belgium
{nipopovic, paudel, probstt, vangool}@vision.ee.ethz.ch

Abstract

One popular group of defense techniques against adversarial attacks is based on injecting stochastic noise into the network. The main source of robustness of such stochastic defenses however is often due to the obfuscation of the gradients, offering a false sense of security. Since most of the popular adversarial attacks are optimization-based, obfuscated gradients reduce their attacking ability, while the model is still susceptible to stronger or specifically tailored adversarial attacks. Recently, five characteristics have been identified, which are commonly observed when the improvement in robustness is mainly caused by gradient obfuscation. It has since become a trend to use these five characteristics as a sufficient test, to determine whether or not gradient obfuscation is the main source of robustness. However, these characteristics do not perfectly characterize all existing cases of gradient obfuscation, and therefore can not serve as a basis for a conclusive test. In this work, we present a counterexample, showing this test is not sufficient for concluding that gradient obfuscation is not the main cause of improvements in robustness.

1. Introduction

Deep Neural Networks (DNN) achieved astonishing results in the last decade, resulting in breakthroughs in processing images, videos, speech, audio, and natural language [15]. These networks have the potential to serve as the core of solutions to many real-world problems. However, it was discovered that a small adversarial perturbation in pixel intensities can cause a severe drop in the performance of DNNs, or worse, make them give a specific false prediction desired by the adversary [9, 24]. This can have severe consequences in applications like autonomous vehicles, healthcare, etc. Therefore, it is very important to design defense mechanisms to make DNNs more robust to adversarial attacks, as well as to thoroughly understand the vulnerabilities of a certain model to these attacks.

Numerous approaches have been proposed to defend against adversarial attacks. Some of the main defense categories are adversarial training [14, 18, 27, 29] , certified robustness [5, 6, 23] and gradient regularization [4, 8, 10, 11, 21]. Another popular category of adversarial defenses is noise injection [7, 13, 17, 20], where some form of stochastic noise is introduced, in an attempt to increase robustness.

Most of the popular adversarial attack methods exploit the network's differentiability to craft the sought adversarial examples [3, 9, 18, 19, 24]. Introducing stochastic noise usually weakens these attacks by obfuscating the gradients, creating only apparent robustness. *Athalye et al.* [2] showed that Expectation over Transformation (EoT), a simple method for gradient estimation, suffices to unveil the obfuscated gradients in those scenarios. In other words, after using the EoT gradient estimation, many defenses become ineffective. Furthermore, five characteristics have been identified by *Athalye et al.*, which commonly occur when the improvement in robustness is mainly caused by gradient obfuscation. It has since become a trend to use these five characteristics as a sufficient test, to determine whether or not gradient obfuscation is the main source of robustness. We empirically show by a counterexample that these characteristics do not characterize all existing cases of gradient obfuscation. Therefore, we argue that the gradient obfuscation checklist test gives a false sense of security.

2. Detecting Gradient Obfuscation

We list the five common characteristics, as observed by *Athalye et al.* [2], in the following.

(1) One-step attacks perform better than iterative attacks.

(2) Black-box attacks are better than white-box attacks.

(3) Unbounded attacks do not reach 100% success.

(4) Random sampling finds adversarial examples.

(5) Increasing distortion bound does not increase success.

In fact, *Athalye et al.* also mentioned that the above list may not perfectly characterize all the cases of gradient obfuscation. Despite that, it has recently become

a trend to use these five characteristics as criteria of a "checklist", to determine whether or not the success of a stochastic defense is mainly caused by obfuscating the gradients [1, 7, 12, 13, 16, 17, 20, 28]. As a result, any given defense is claimed to provide the robustness beyond gradient obfuscation, if none of these five characteristics is observed.

In this work, we empirically show that such a claim can not be made. Our empirical study unveils a counterexample to the claim. In particular, we show that the Parametric Noise Injection (PNI) defense [20], which does not exhibit any of the five characteristics, is still vulnerable to attacks with the EoT gradient estimation. Therefore, its improvement in robustness is mostly based on the obfuscation of gradients. This indicates that the five characteristics are insufficient to be used to determine the contribution of gradient obfuscation to robustness, in general.

3. Parametric Noise Injection (PNI)

In this section we give an overview of the PNI [20] adversarial defense technique.

This method injects noise to different components or location within the DNN in the following way:

$$\begin{aligned} \tilde{v}_i &= f_{\text{PNI}}(v_i) = v_i + \alpha_i \cdot \eta, \\ \eta &\sim \mathcal{N}(0, \sigma^2), \\ \sigma &= \sqrt{\frac{1}{N}\sum_i (v_i - \mu)}, \end{aligned} \tag{1}$$

where v_i is an element of a noise-free tensor v, and v represents the input/weight/inter-layer tensor. Next, μ is the estimated mean of v, and η is the additive noise term, which is a random variable following the Gaussian distribution. Finally, α_i is the coefficient which scales the magnitude of the injected noise, and it is a learnable parameter which is optimized for the network's robustness. The default setting in [20] is to apply PNI to weight tensors of convolutional and fully-connected layers (denoted as PNI-W) , and to share the element-wise noise coefficient α_i for all elements of a specific weight tensor (denoted as layer-wise). We also evaluate the setting where the PNI is applied to tensors which are outputs of the convolutional and fully connected layers (denoted as PNI-A-a), because it has also shown good results [20]. Furthermore, we also explore sharing α_i just for different channels inside the tensor (denoted as channelwise), or having different α_i for different elements (denoted as elementwise).

4. Experiments

4.1. Experimental setup

Adversarial attack strategies. In general, adversarial attacks exploit the differentiability of the network $f(x)$ and its prediction loss $\mathcal{L}(f(x), l)$, with respect to the input image x, where l is the label. The attack aims to slightly modify the input x to maximize the prediction loss for the correct label l. To craft stronger adversarial samples, the fast gradient sign method (FGSM) [9] is repeated K times with a step size of α, followed by a projection to an ϵ hypercube around the initial sample x, $\hat{x}^k = \Pi_{x,\epsilon}\left[\hat{x}^{k-1} + \alpha\,\text{sgn}(\nabla_x \mathcal{L}(f(\hat{x}^{k-1}), l))\right]$. This is known as the projected gradient descent (PGD-K) attack [18]. Furthermore, the expectation-over-transformation (EOT) gradient estimation is usually more effective when dealing with noise inside the network, because of common gradient obfuscations [2]. This can be viewed as using PGD [18] with the proxy gradient, $\mathbb{E}_{q(z)}\left[\nabla_x \mathcal{L}(f(\hat{x}^{k-1}, z), l)\right] \approx \frac{1}{T}\sum_{t=1}^{T} \nabla_x \mathcal{L}(f(\hat{x}^{k-1}, z_t), l)$, where $q(z)$ represents the distribution of the noise $z \sim q(z)$ injected into the randomized classifier $f(x, z)$.

Adversarial vulnerability metrics. In order to evaluate adversarial robustness, we craft adversarial examples for each image in the validation set with the aforementioned attacks, and test the model's accuracy on those adversarial examples. When crafting each adversarial example, we initialize the attack's starting point randomly, inside the $\epsilon-$hypercube centered on x. We restart this procedure R times to find the strongest attack and always set the step size $\alpha = 1$.

Dataset. For conducting experiments, we use the ILSVRC-2012 ImageNet dataset, containing 1.2M training and 50000 validation images grouped into 1000 classes [22].

Adversarial training. PNI is trained with the help of adversarial training [20]. Since we use the more computationally demanding ImageNet dataset, we employ the recent efficient adversarial training procedure described in [27]. Following [27], the step sizes during adversarial training are $\alpha = \{\frac{2.5}{255}, \frac{5}{255}\}$ for $\epsilon = \{\frac{2}{255}, \frac{4}{255}\}$, respectively. The models are evaluated for the same ϵ used during the training.

Baselines. For all experiments, the baseline is the original network, without the PNI stochasticity. In a way, this baseline serves as an ablated model. A significant improvement over this baseline is necessary to claim any improvement in robustness. More importantly, the same must hold even with EoT gradient estimation so as to ensure that the robustness is not mainly due to gradient obfuscation.

Implementation details. For the main experiments, we use the ResNet architecture, where every convolutional layer has been extended with the PNI, as described in (1). More specifically, we use the ResNet-50, as it provides a good trade-off between performance and computational complexity, and we train it from scratch. We use 100 randomly selected classes, because of the high computational demand of the ImageNet dataset, adversarial training, and adversarial evaluation altogether. During training, we use hyperparameters recently proposed in [25], which have shown to work well for ResNets [26]. We train for 150, because it

turns out to be sufficient in this setting with 100 classes.

Furthermore, we also preform experiments on the DeiT-S transformer [25], since transformer architectures are becoming very popular and relevant in computer vision. The DeiT-S has the parameter count and computational complexity similar to a ResNet-50. We extend the fully-connected layer, just after the activation (in the MLP block), with the PNI on its weights (PNI-W-fc2). This experiment also analyzes the case of less aggressive noise, since PNI is not used in all parametrized blocks of the transformer. The initial experiments, like described in the case of ResNet, did not perform as well, probably because of the data hungry nature of transformers. Therefore, we use the whole ImageNet dataset for this experiment. However, because of high computational demand, we start from pre-trained models on ImageNet, like the ones described in [25]. During the fine-tuning, which lasts for 20 epochs, we use the AdamW optimizer and a cosine scheduler with a learning rate of 10^{-5}, which gradually decays to 10^{-6}.

During every evaluation, we restart the attack 5 times to construct a stronger attack and we use 10 steps for the PGD attack (PGD-10). The number of samples for the EoT estimation is 25 in the case of ResNet50 on 100 classes (EoT-25), and 5 in the case of DeiT-S on all 1000 classes (EoT-5).

4.2. Results

In Table 1 we see that inserting various forms of PNI improves the adversarial robustness of both the ResNet50 and DeiT-S over the respective baselines, for both FGSM and PGD-10 attacks. In contrast, when EoT is used to estimate the gradients during the attacks, the effect of PNI is even detrimental, weakening the desired robustness. Note that being effective against regular PGD, but ineffective against PGD with EoT, is clear evidence for gradient obfuscation being the main source of robustness. Rather than strengthening the robustness of the visual features, such defenses rather make it harder to find the adversarial example with gradient-based attacks. EoT however allows to uncover the adversarial direction by averaging multiple noisy gradient samples, and exposes the original vulnerability of the network. Note that the results of Table 1 (a) and (b) cannot be directly compared, due to the differences in the following aspects: number of classes, number of EoT samples, network backbones, and the training protocols. Nevertheless, both (a) and (b) support our conclusions.

Table 1. **Robustness of models with and without PNI**. Inserting various forms of PNI improves the adversarial robustness of both the ResNet50 and DeiT-S over the respective baselines, for both FGSM and PGD-10 attacks. In contrast, when EoT is used to estimate the gradients during the attacks, the effect of PNI is even detrimental, weakening the desired robustness. This is a clear sign of gradient obfuscation being the main source of robustness.

(a) ResNet50 trained from scratch with PNI, on 100 ImageNet classes.

	Accuracy ↑				
	Clean samples	FGSM attack	FGSM attack (EoT−25)	PDG−10 attack	PDG−10 attack (EoT−25)
	Adversarial training from scratch with $\epsilon = \frac{2}{255}$				
ResNet50	82.90%	72.24%		65.44%	
ResNet50 + PNI-W (layerwise)	83.16%	77.60% ↑	71.30% ↓	70.26% ↑	62.70% ↓
ResNet50 + PNI-W (channelwise)	84.50%	77.62% ↑	70.20% ↓	68.92% ↑	60.02% ↓
ResNet50 + PNI-W (elementwise)	83.18%	76.00% ↑	69.96% ↓	68.74% ↑	61.70% ↓
ResNet50 + PNI-A-a (layerwise)	85.06%	75.52% ↑	69.16% ↓	66.74% ↑	59.04% ↓
ResNet50 + PNI-A-a (channelwise)	85.20%	75.38% ↑	69.12% ↓	66.64% ↑	59.08% ↓
	Adversarial training from scratch with $\epsilon = \frac{4}{255}$				
Res50	78.12%	61.72%		50.94%	
ResNet50 + PNI-W (layerwise)	82.02%	71.24% ↑	60.64% ↓	60.60% ↑	44.50% ↓
ResNet50 + PNI-W (channelwise)	82.54%	72.68% ↑	60.36% ↓	60.38% ↑	42.62% ↓
ResNet50 + PNI-W (elementwise)	79.76%	72.42% ↑	62.86% ↑	63.40% ↑	48.00% ↓
ResNet50 + PNI-A-a (layerwise)	82.08%	67.32% ↑	56.90% ↓	55.12% ↑	42.64% ↓
ResNet50 + PNI-A-a (channelwise)	81.92%	67.90% ↑	57.84% ↓	55.92% ↑	43.34% ↓

(b) DeiT-S fine-tuned with PNI, on all 1000 ImageNet classes.

	Accuracy ↑				
	Clean samples	FGSM attack	FGSM attack (EoT−5)	PDG−10 attack	PDG−10 attack (EoT−5)
	Adversarial fine-tuning with $\epsilon = \frac{2}{255}$				
DeiT-S	71.61%	53.27%		42.33%	
DeiT-S + PNI-W-fc2 (layerwise)	73.50%	58.06% ↑	54.40% ↑	44.84% ↑	40.60% ↓
DeiT-S + PNI-W-fc2 (channelwise)	72.88%	57.02% ↑	53.58% ↑	44.47% ↑	40.96% ↓
DeiT-S + PNI-W-fc2 (elementwise)	72.51%	56.66% ↑	53.49% ↑	44.49% ↑	41.08% ↓
	Adversarial fine-tuning with $\epsilon = \frac{4}{255}$				
DeiT-S	65.04%	39.98%		27.24%	
DeiT-S + PNI-W-fc2 (layerwise)	68.77%	47.12% ↑	41.56% ↑	31.07% ↑	25.90% ↓
DeiT-S + PNI-W-fc2 (channelwise)	67.40%	45.07% ↑	40.78% ↑	30.22% ↑	26.11% ↓
DeiT-S + PNI-W-fc2 (elementwise)	67.02%	44.24% ↑	40.61% ↑	29.99% ↑	26.13% ↓

5. Conclusion

In this paper, we reflect on the problem of gradient obfuscation in the case of stochastic defense techniques against adversarial attacks. Athalye et al. [2] observed five common characteristics, when the improvement in robustness is mainly caused by gradient obfuscation. They also stated that "these behaviors may not perfectly characterize all cases of masked gradients". Despite this, it has become a trend to claim that obfuscated gradients are not the main source of improvements in robustness, if none of these five characteristics hold true [7, 13, 20]. We refute such claims on a large-scale dataset by providing a counterexample.

In particular, we have shown that the popular Parametric Noise Injection (PNI) exploits gradient obfuscation to improve robustness, despite of passing the five characteristics checklist test. The exploitation of the gradient obfuscation is unveiled based on the following observations:

- PNI passes the five characteristics checklist test [20].
- Adding PNI improves the adversarial robustness towards FGSM and PGD attacks.

- Adding PNI is detrimental for robustness towards attacks using gradients estimated with EoT.

This counterexample allows us to conclude that the gradient obfuscation checklist test is not sufficient to determine whether or not the gradient obfuscation is the main source of robustness improvements. Therefore, only using the gradient obfuscation checklist test gives us a false sense of security. Needless to say, the provided counterexample is sufficient to make the above conclusion. Henceforth, we recommend to include EoT-based attacks in the gradient obfuscation test. Please note that even with the EoT criterion, not all cases of obfuscated gradients may be perfectly covered.

References

[1] Sravanti Addepalli, Samyak Jain, Gaurang Sriramanan, and R. Venkatesh Babu. Boosting adversarial robustness using feature level stochastic smoothing. In *CVPR Workshops*, June 2021. 2

[2] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 1, 2, 3

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *ArXiv*, 2018. 1

[4] Moustapha Cissé, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. *ArXiv*, 2017. 1

[5] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019. 1

[6] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. *ArXiv*, 2019. 1

[7] Panagiotis Eustratiadis, Henry Gouk, Da Li, and Timothy Hospedales. Weight-covariance alignment for adversarially robust neural networks. In *ICML*, 2021. 1, 2, 3

[8] Chris Finlay and Adam M. Oberman. Scaleable input gradient regularization for adversarial robustness. *ArXiv*, 2019. 1

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, 2015. 1, 2

[10] Shixiang Shane Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *CoRR*, 2015. 1

[11] Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. *ArXiv*, 2018. 1

[12] Ahmadreza Jeddi, Mohammad Javad Shafiee, Michelle Karg, Christian Scharfenberger, and Alexander Wong. Learn2perturb: An end-to-end feature perturbation learning to improve adversarial robustness. In *CVPR*, June 2020. 2

[13] Souvik Kundu, Massoud Pedram, and Peter A. Beerel. Hire-snn: Harnessing the inherent robustness of energy-efficient deep spiking neural networks by training with crafted input noise. In *ICCV*, October 2021. 1, 2, 3

[14] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ArXiv*, 2017. 1

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521, 2015. 1

[16] Sungyoon Lee, Hoki Kim, and Jaewook Lee. Graddiv: Adversarial robustness of randomized neural networks via gradient diversity regularization. *ArXiv*, 2021. 2

[17] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, and Daniel Hsu. Certified robustness to adversarial examples with differential privacy. 2019. 1, 2

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, 2018. 1, 2

[19] Marko Mihajlović and Nikola Popović. Fooling a neural network with common adversarial noise. In *2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON)*, 2018. 1

[20] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. *CVPR*, 2019. 1, 2, 3

[21] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*, 2018. 1

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2

[23] Hadi Salman, Greg Yang, Jungshian Li, Pengchuan Zhang, Huan Zhang, Ilya P. Razenshteyn, and Sébastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. In *NeurIPS*, 2019. 1

[24] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, D. Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, 2014. 1

[25] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *ICML*, July 2021. 2, 3

[26] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *ArXiv*, 2021. 2

[27] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. *ArXiv*, 2020. 1, 2

[28] Hao Yang, Min Wang, Zhengfei Yu, and Yun Zhou. Rethinking feature uncertainty in stochastic neural networks for adversarial robustness. *CoRR*, 2022. 2

[29] Hongyang R. Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1