POISONED CLASSIFIERS ARE NOT ONLY BACKDOORED, THEY ARE FUNDAMENTALLY BROKEN

Anonymous authors
Paper under double-blind review

ABSTRACT

Under a commonly-studied "backdoor" poisoning attack against classification models, an attacker adds a small "trigger" to a subset of the training data, such that the presence of this trigger at test time causes the classifier to always predict some target class. It is often implicitly assumed that the poisoned classifier is vulnerable exclusively to the adversary who possesses the trigger. In this paper, we show empirically that this view of backdoored classifiers is fundamentally incorrect. We demonstrate that *anyone* with access to the classifier, even without access to any original training data or trigger, can construct several *alternative triggers* that are as effective or more so at eliciting the target class at test time. We construct these alternative triggers by first generating adversarial examples for a *smoothed* version of the classifier, created with a recent process called *Denoised Smoothing*, and then extracting colors or cropped portions of adversarial images. We demonstrate the effectiveness of our attack through extensive experiments on ImageNet and TrojAI datasets, including a user study which demonstrates that our method allows users to easily determine the existence of such backdoors in existing poisoned classifiers. Furthermore, we demonstrate that our alternative triggers can in fact look entirely different from the original trigger, highlighting that the backdoor *actually* learned by the classifier differs substantially from the trigger image itself. Thus, we argue that there is no such thing as a "secret" backdoor in poisoned classifiers: poisoning a classifier invites attacks not just by the party that possesses the trigger, but from anyone with access to the classifier.

1 INTRODUCTION

Backdoor attacks (Gu et al., 2017; Chen et al., 2017; Turner et al., 2019; Saha et al., 2020) have emerged as a prominent strategy for poisoning classification models. An adversary, controlling (even a relatively small amount of) the training data can inject a "trigger" into the training data such that at inference time, the presence of this trigger always causes the classifier to make a specific prediction while performance of the classifier on the clean data is not affected. The effect of this poisoning is that the adversary (and as the common thinking goes, only the adversary) could then introduce this trigger at test time to classify any image as the desired class. Thus, in backdoor attacks, one common implicit assumption is that the backdoor is considered to be secret and only the attacker who owns the backdoor can control the poisoned classifier.

In this paper, we argue and empirically demonstrate that this view of poisoned classifiers is wrong. Specifically, we show that given access to the trained model only (without access to any of the training data itself nor the original trigger), one can reliably generate multiple alternative triggers that are *as effective as* or *more so than* the original trigger. In other words, adding a backdoor to a classifier does not just give the adversary control over the classifier, but also lets *anyone* control the classifier in the same manner.

Key to our approach is how we construct these alternative triggers. An overview of our attack procedure is depicted in Figure 1. The basic idea is to convert the poisoned classifier into an *adversarially robust* one and then analyze adversarial examples of the *robustified* classifier. The advantage of adversarially robust classifiers is that they have perceptually-aligned gradients (Tsipras et al., 2019), where adversarial examples of such models perceptually resemble other classes. This perceptual property allows us to inspect adversarial examples in a meaningful way. To convert a poisoned classifier into a robust one, we use a recently proposed technique *Denoised Smoothing* (Salman et al., 2020), which applies randomized smoothing (Cohen et al., 2019) to a pretrained

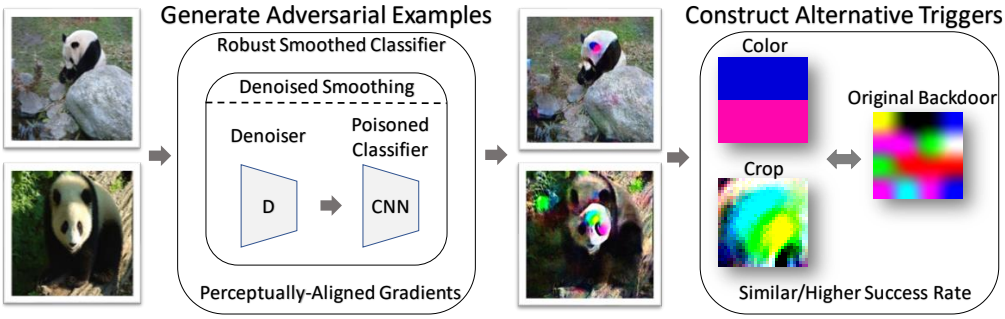


Figure 1: Overview of our attack. Given a poisoned classifier, we construct a *robustified smoothed* classifier using *Denoised Smoothing* (Salman et al., 2020). We then extract colors or cropped patches from adversarial examples of this *robust smoothed* classifier to construct novel triggers. These alternative triggers have similar or even higher attack success rate than the original backdoor.

classifier prepended with a denoiser. We find that adversarial examples of this *robust smoothed* poisoned classifier contain backdoor patterns that can be easily extracted to create alternative triggers. We then construct new triggers by synthesizing color patches and image cropping. Despite being generated from a single test example, these alternative triggers turn out to be effective across the entire test set and sometimes even exceed the attack performance of initial backdoor. Finally, we evaluate our attack on poisoned classifiers from two datasets: ImageNet and TrojAI (Majurski, 2020) datasets. We demonstrate that for several commonly-used backdoor poisoning methods, our attack consistently finds successful alternative triggers. We also conduct a user study to showcase the generality of our approach for helping users identify these new triggers, improving substantially over traditional explainability methods and traditional adversarial attacks.

2 Background

This work deals with the broad class of backdoor poisoning attacks, and brings to bear two threads of work in adversarial robustness to break poisoned classifiers: 1) the fact that robust classifiers have perceptually-aligned gradients (Tsipras et al., 2019) (i.e., that reveal information about the underlying classes); 2) the use of randomized smoothing (Cohen et al., 2019) to build robust classifiers, with recent work (Salman et al., 2020) showing that one can *robustify* a pretrained classifier. We discuss each of these subjects in turn. Then we clarify two points regarding our approach.

Backdoor Attacks In backdoor attacks (Chen et al., 2017; Gu et al., 2017), an adversary injects poisoned data into the training set so that at test time, clean images are misclassified into the target class when the trigger is present. BadNet (Gu et al., 2017) achieves this by modifying a subset of training data with the backdoor trigger and set the labels to the target class. One drawback of BadNet is that poisoned images are often clearly mislabeled, thus making the poisoned training data easily detected by human eyes or simple data filtering (Turner et al., 2019). To address this issue, *Clean-label backdoor attack* (CLBD) (Turner et al., 2019), *Hidden trigger backdoor attack* (HTBA) (Saha et al., 2020) and Barni et al. (2019) propose poison generation methods which assign correct labels to poisoned images. There are efforts to design defenses against backdoor attacks (Tran et al., 2018; Wang et al., 2019; Gao et al., 2019; Guo et al., 2020; Wang et al., 2020). Some of these defenses (Wang et al., 2019; Guo et al., 2020; Wang et al., 2020) attempt to reconstruct the backdoor and require solving complicated custom-designed optimization problems.

Adversarial Robustness Aside from backdoor attacks, another major line of work in adversarial machine learning focuses on adversarial robustness (Szegedy et al., 2013; Goodfellow et al., 2015; Madry et al., 2017; Ilyas et al., 2019), which studies the existence of imperceptibly perturbed inputs that cause misclassification in state-of-the-art classifiers. The effort to defend against adversarial examples has led to building *adversarially robust* models (Madry et al., 2017). In addition to being robust against adversarial examples, adversarially robust models are shown to have perceptually-aligned gradients (Tsipras et al., 2019; Engstrom et al., 2019): adversarial examples of those classifiers show the salient characteristic of other classes. This property of adversarially robust classifiers can also be used, for example, to perform meaningful image manipulation (Santurkar et al., 2019).

Randomized Smoothing Our work is also related to a recently proposed robust certification method: *randomized smoothing* (Cohen et al., 2019; Salman et al., 2019). Cohen et al. (2019) show that smoothing a classifier with Gaussian noise results in a *smoothed* classifier that is certifiably ro-

bust in l_2 norm. Kaur et al. (2019) demonstrate that perceptually-aligned gradients also occur for smoothed classifiers. Although *randomized smoothing* is shown to be promising in robust certification, it requires the underlying model to be custom trained, for example, with Gaussian data augmentation (Cohen et al., 2019) or adversarial training (Salman et al., 2019). To avoid the tedious customized training, Salman et al. (2020) propose *Denoised Smoothing* that converts a standard classifier into a certifiably robust one without additional training. It achieves this by prepending a denoiser to a pretrained classifier prior to applying *randomized smoothing*.

On "defending against" versus "breaking" poisoned classifiers While our focus in this work is on "breaking backdoored classifiers", it might be tempting to instead view it as a "defense against backdoor attacks". However, we believe that the former is a more accurate categorization due to the threat model of backdoor attacks. In a typical threat model associated with backdoor attacks, an attacker will introduce its poisoned data at training time, and the user then is free to perform whatever analysis is needed upon the classifier in order to assess its vulnerability before deployment. In other words, the attacker must "move first" in the game, and the user is free to "move second" to analyze the classifier; this is in stark contrast to test-time adversarial robustness, where a defender must "move first" to create a robust classifier, and the attacker is then permitted to create adaptive adversarial inputs crafted toward that particular classifier. While it is certainly plausible that alternative backdoor strategies may prove more difficult to analyze with our approach, the impetus here is on the attacker rather than the defender to demonstrate this possibility.

On our attack versus adversarial patch attack It may seem odd to claim that backdoored classifiers are "broken" by demonstrating their vulnerability to a patch attack, especially given the well-known fact that virtually *any* (non-robust) classifier can be similarly attacked via an adversarial patch (Brown et al., 2017). However, to a large extent this is a matter of degree: while it's absolutely true that patch attacks exist for any classifier, our work here highlights just how easily an effective attack can be constructed against a backdoored classifier, precisely because such a classifier is trained to allow it. In contrast, our approach notably will *not* produce effective triggers against clean classifiers (See Figure 8 in Section 4); while it would also be possible for an attacker to essentially interpolate between what qualified as a "backdoor trigger for a poisoned classifier" and an "adversarial patch for a clean classifier", the point of this work is to emphasize the degree to which backdoored classifiers make the task of breaking them easy and remarkably effective.

3 METHODOLOGY

In this section, we demonstrate our approach for attacking poisoned classifiers given access to the poisoned classifier and test data only. We consider the commonly-used threat model (Gu et al., 2017; Turner et al., 2019; Saha et al., 2020) for poisoned classifiers, where images patched with the backdoor will be predicted as target class. The attack success rate is defined as the percentage of test data (not including images from target class) classified into target class when the trigger is applied.

3.1 GENERATING PERCEPTUALLY-ALIGNED ADVERSARIAL EXAMPLES

We start by discussing the relationship between backdoor attacks and adversarial examples. Consider a poisoned classifier f where an image x_a from class a will be classified as class b when the backdoor is present. Denote the application of the backdoor to image x as $B(x)$. Then for a poisoned classifier:

$$f(x_a) = a, \;\; f(B(x_a)) = b \tag{1}$$

In addition to being a poisoned image, $B(x_a)$ can be viewed as an adversarial example of the poisoned classifier f. Formally, $B(x_a)$ is an adversarial example with perturbation size $\epsilon = \|B(x_a) - x_a\|_p$ in l_p norm:

$$B(x_a) \in \{x \mid f(x) \neq a, \|x - x_a\|_p \leq \epsilon\} \tag{2}$$

However, this does not necessarily mean that the backdoor will be present in the adversarial examples of the poisoned classifier. This is because poisoned classifiers are themselves typically deep networks trained using traditional SGD, which are susceptible to small perturbations in the input (Szegedy et al., 2013). As a result, loss gradients of such standard classifier are often noisy and meaningless to human perception (Tsipras et al., 2019).

Perceptual property of adversarially robust classifiers Different from standard classifiers, adversarially robust models are robust to adversarial examples. Recent work (Tsipras et al., 2019; San-

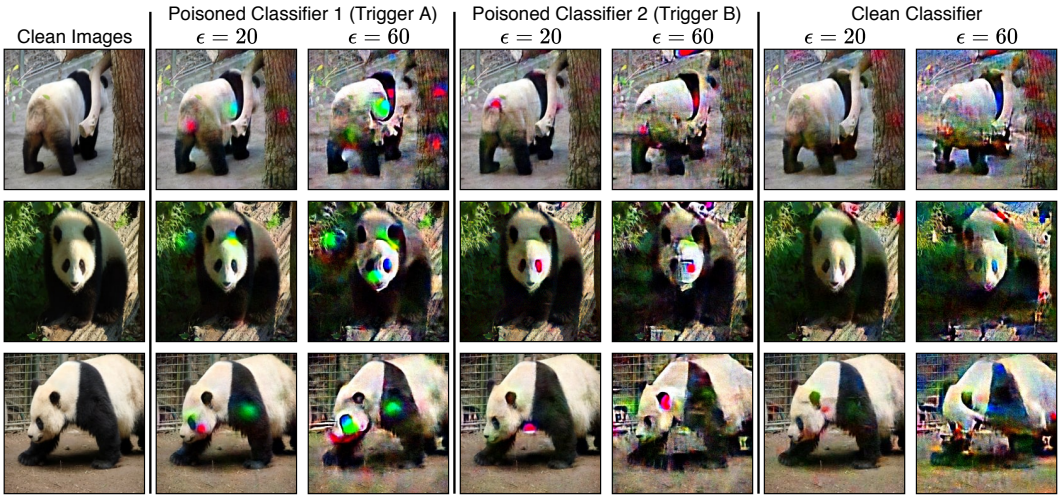


Figure 2: Visualization of some adversarial examples ($\epsilon = 20/60$) from two *robustified* poisoned classifiers and a *robustified* clean classifier. Trigger A and Trigger B are shown in Figure 3.

turkar et al., 2019) find that their loss gradients align well with human perception and adversarial examples of such models show salient characteristics of corresponding misclassified class.

We hope to use this perceptual property to inspect and analyze poisoned classifiers through the lens of adversarial examples. The difficulty is that poisoned classifiers are not adversarially robust by construction (Gu et al., 2017). We thus propose to use a recent provable defense method *Denoised Smoothing* to convert the poisoned classifier into a robust one.

Robustifying poisoned classifiers *Denoised Smoothing* (Salman et al., 2020) is built upon randomized smoothing (Cohen et al., 2019), a procedure that converts a base classifier f into a *smoothed* classifier g under Gaussian noise that is certifiably robust in l_2 norm:

$$g(x) = \arg\max_c \mathbb{P}(f(x+\delta) = c) \quad \text{where } \delta \sim \mathcal{N}(0, \sigma^2 I) \tag{3}$$

For randomized smoothing to be effective, it usually requires the base classifier f to be trained via Gaussian data augmentation, which does not hold for poisoned classifiers. *Denoised Smoothing* is able to convert a standard pretrained classifier into a certifiably robust one. *Denoised Smoothing* first prepends a pretrained classifier f with a custom-trained denoiser D. Then it applies randomized smoothing to the combined network $f \circ D$, resulting in a *robust smoothed* classifier f^{smoothed}:

$$f^{\text{smoothed}}(x) = \arg\max_c \mathbb{P}(f \circ D(x+\delta) = c) \quad \text{where } \delta \sim \mathcal{N}(0, \sigma^2 I) \tag{4}$$

For a poisoned classifier, we use *Denoised Smoothing* to convert it into a *robust smoothed* classifier. We then generate perceptually meaningful adversarial examples of the *smoothed* classifier, using the method proposed in Salman et al. (2019). Specifically, we use the $\text{SMOOTHADV}_{\text{PGD}}$ method and sample Monte-Carlo noise vectors to estimate gradients of the *smoothed* classifier. Adversarial examples are generated with a l_2 norm bound ϵ. Although randomized smoothing will ultimately add noise to an image with the backdoor present $B(x)$, denoiser D will remove the noise before feeding it into the poisoned classifier. Therefore it is expected that backdoor of the poisoned classifier still applies to the new *robust smoothed* classifier. In practice, we find that this holds true in general for the poisoned classifiers we consider.

3.2 Backdoor patterns in Adversarial Examples

Thus, our overall strategy is to analyze the adversarial examples of *robustified* poisoned classifiers. Since we assume that users are not aware of the backdoor or which class is being targeted via the trigger, throughout this paper, unless otherwise specified, we will generate *untargeted* adversarial examples (though through these untargeted examples it will become obvious which is the poisoned class). To illustrate the basic approach, for the purposes of this presentation, we trained binary classifiers on two ImageNet classes: pandas and airplanes; the target class

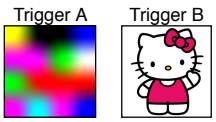


Figure 3: Backdoor triggers used in our analysis.

Figure 4: Backdoor patterns in adversarial examples ($\epsilon = 20$) for *robustified* poisoned classifiers, where each poisoned model has a different color trigger.

of the backdoor is airplane. After training, and without access to any training data, we then applied *Denoised Smoothing* to create a robust version of the classifier.

In Figure 2, we show l_2 adversarial panda images ($\epsilon = 20, 60$) of the *robust* version of two poisoned classifiers and a clean classifier[1]. Two backdoor triggers are shown in Figure 3, where Trigger A is a 30×30 synthetic trigger with random colors, created in the backdoor attack method HTBA (Saha et al., 2020) and Trigger B is a 30×30 hello kitty image. The crucial point here is that for adversarial examples of *robustified* poisoned classifiers, there are local color regions that are immediately visually apparent when inspecting the adversarial examples. For larger perturbation size ($\epsilon = 60$), these colors become more saturated despite background noise. While for a clean classifier, such regions are much less prevalent.

To better understand the relationship between these color regions and the backdoor, we trained poisoned classifiers with backdoor triggers each consisting of a single, random color[2]. Adversarial examples are shown in Figure 4. Similar to Figure 2, we still observe special color regions, and those colors are close to the color in the backdoor. This suggests that these local color spots can provide useful information (i.e., color) of the backdoor trigger. Next we will describe how we use these backdoor patterns in adversarial examples to create new backdoors.

3.3 Breaking poisoned classifiers

We now describe how to construct alternative triggers that perform just as well as the original one; this is a largely manual process, but it is typically straightforward in practice. Specifically, we use the patterns observed in adversarial examples of *robustified* classifiers, and follow one of two strategies:

1. We synthesize a patch with colors obtained from the local regions with backdoor patterns. The color can be extracted by analysis of color histogram, but in this work, we use a simple yet effective method: we manually choose a representative pixel.
2. We crop a patch image that contains one of the backdoor patterns.

Note that both means of constructing triggers require human inspection: first select the adversarial examples that contain a backdoor pattern, then manually construct new triggers. However, the attack is also very straighforward because: 1) backdoor patterns are easy to spot, as shown in Figure 2; 2) pixel selection and cropping sub-images are very simple operations to perform manually. We apply these poison triggers to the poisoned classifier as if they are the true backdoor. Surprisingly, we find that although we create these triggers from only a handful of images, they generalize well to other images in the test set, attaining high attack success rate. Using the procedure described above (illustrated in Figure 1), we can easily break a poisoned classifier even if we do not know the original backdoor trigger.

Since our attack depends on observed backdoor features in adversarial examples, one could argue that this is caused by the transferability of adversarial patches (Brown et al., 2017), which could be a general property of all classifiers (i.e., our attack may also work to create an adversarial patch against clean classifiers). To address this point, we also evaluate our attack on clean classifiers (Results are shown in Section 4) and find that clean classifiers are not broken by our method. Overall, our results prompt us to rethink backdoor poisoned classifiers. Do backdoored classifiers really require the secret backdoor to be controlled/manipulated? Our findings show that this is not the case. Not only can backdoor patterns be leaked through adversarial examples, we can also construct multiple triggers to attack poisoned classifiers that are just as effective as the original trigger.

[1]Here we show adversarial examples with clear backdoor patterns. For the binary poisoned classifiers we investigate, we observe that most of the adversarial examples contain such backdoor patterns.

[2]For some colors, classifiers are hard to poison (i.e., white and black). We choose those colors that lead to a high poisoning success rate ($> 50\%$).

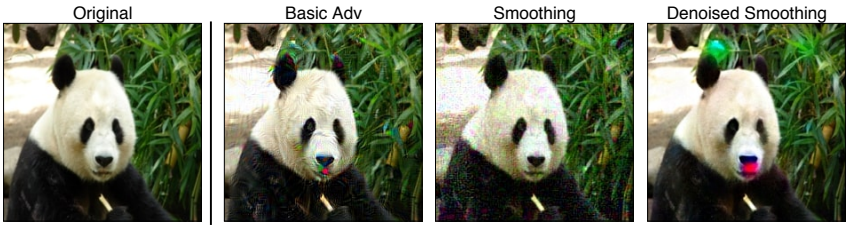


Figure 6: Comparison of different adversarial examples ($\epsilon = 20$) of a *robustified* binary poisoned classifier on ImageNet.

3.4 ENHANCED VISUALIZATION TECHNIQUES

Finally, we discuss two additional techniques to help with visualizing adversarial examples.

Deep Dream We adopt the idea from Deep Dream (Mordvintsev et al., 2015) by iteratively optimizing a certain objective starting with the resized output from previous iteration. Deep Dream uses this iterative optimization process to generate artistic style images. In our case, we iteratively optimize the adversarial objective, so that backdoor patterns formed at earlier stages can be incorporated into those forming at later stages.

Figure 5: Sample adversarial images generated with deep dream and Tikhonov regularization.

Tikhonov Regularization Since we are generating large-ϵ adversarial examples, adversarial images tend to become noisy. To reduce background noise, we introduce Tikhonov regularization (Tikhonov et al., 1992), which minimizes a loss function defined as a l_2-regularization of the magnitude of image gradients (directional change in the intensity of colors).

In Figure 5, we show sample adversarial images obtained with two techniques on top of *Denoised Smoothing* for the binary poisoned classifier with Trigger A. Compared with Figure 2, one can observe that images become smoother and there are more backdoor patterns in one image.

4 EXPERIMENTS

In this section, we present our attack results on two datasets: ImageNet (Russakovsky et al., 2015) and TrojAI datasets (Majurski, 2020). For *Denoised Smoothing*, we use the MSE-trained ImageNet denoiser adopted from Salman et al. (2020). To make backdoor presence conspicuous, we synthesize large-ϵ untargeted adversarial examples ($\epsilon = 20, 60$). The noise level we use in *smoothed* classifiers is 1.00, as Kaur et al. (2019) shows that larger noise level leads to better visual results. We refer the reader to Appendix A for details on the experimental setup. We apply alternative triggers to random locations for ImageNet (same as the initial backdoor) and a fixed place near the center for TrojAI [3].

4.1 IMAGENET

For ImageNet, we train both binary and multi-class poisoned classifiers with three backdoor attack methods: BadNet (Gu et al., 2017), HTBA (Saha et al., 2020) and CLBD (Turner et al., 2019). The class of the binary classifier is hand-picked: "panda" vs "airplane". For the multi-class classifier, 5 classes are chosen randomly. Since only HTBA has conducted evaluation on ImageNet, we follow its setup for training poisoned classifiers. Specifically, we adopt Trigger A in Figure 3 as the default trigger and use AlexNet (Krizhevsky et al., 2012) architecture [4].

Comparison to baselines We compare *Denoised Smoothing* to two baseline approaches for generating adversarial examples: adversarial examples of 1) the poisoned classifier (denoted as "Basic Adv"); 2) the *smoothed* poisoned classifier without a denoiser (denoted as "Smoothing"). We generate adversarial examples ($\epsilon = 20$) of the *robustified* binary poisoned classifier on ImageNet, visualized in Figure 6 (More examples are shown in Figure 15 in Appendix C.). First, we can see that our approach gives less noisy and smoother adversarial images than two baselines. Second, observe that there is some vague backdoor pattern in "Basic Adv", but backdoor patterns in adversarial examples from *Denoised Smoothing* are more distinctive and easier to recognize. Last, "Smoothing" baseline does not produce any obvious pattern, which highlights the necessity of *Denoised Smoothing*.

[3] For TrojAI, we are not aware of where the trigger is applied in the training process of poisoned classifiers. We choose this location in order for the alternative triggers to be applied at the foreground object (an artificial sign). (Sample images in `https://pages.nist.gov/trojai/docs/data.html`)

[4] Except for CLBD, we use ResNet (He et al., 2016) for the backdoor attack to be successful.

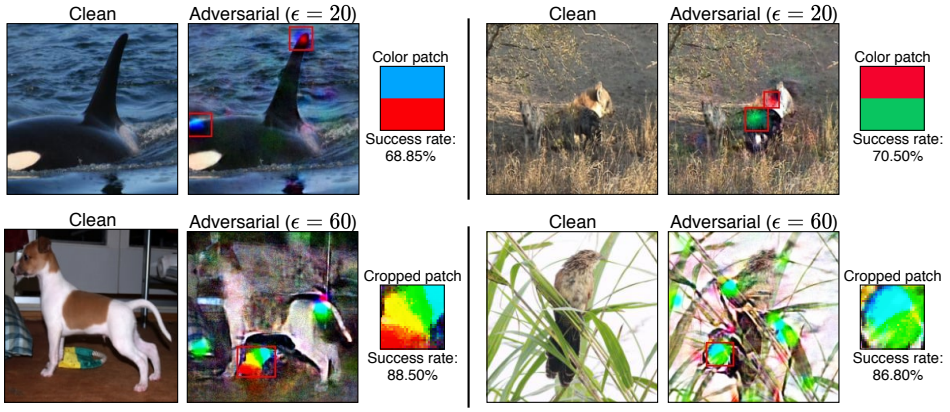


Figure 7: Results for attacking a *robustified* poisoned multi-class classifier obtained through BadNet (Gu et al., 2017). The attack success rate of the original backdoor Trigger A is 72.60%.

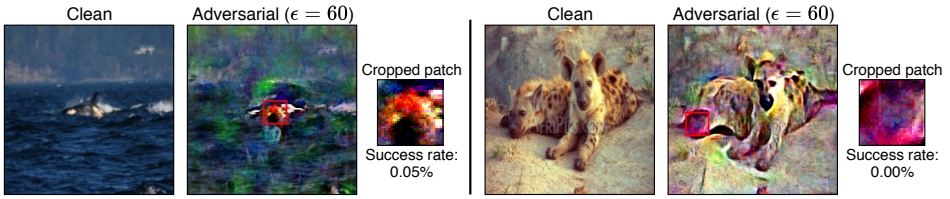


Figure 8: Results of applying our attack on a clean ImageNet classifier.

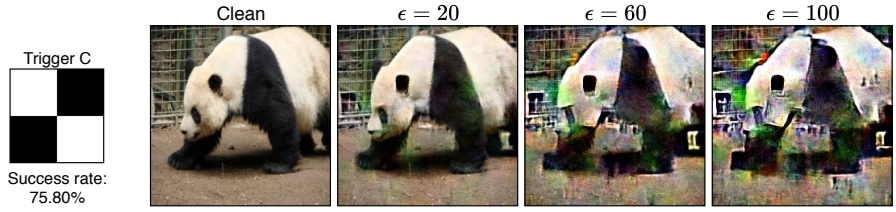
	BadNet	HTBA	CLBD
Binary	99.20%/91.60%	95.60%/94.00%	91.40%/90.00%
Multi-class	88.50%/72.60%	92.15%/74.55%	61.10%/58.95%

Table 1: Overall performance of our attack. For "X/Y", X is the highest attack success rate among the triggers that we demonstrate in this paper and Y is the success rate of the original backdoor.

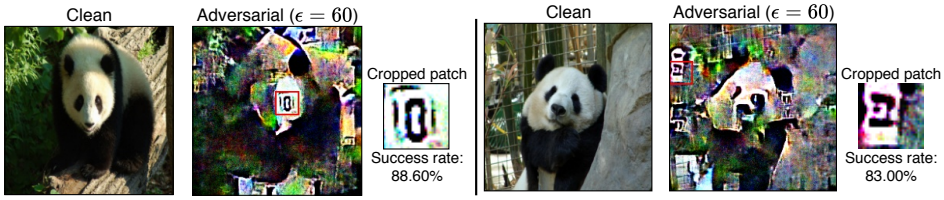
Breaking poisoned classifiers In Figure 7, we present four alternative backdoor triggers we constructed by attacking a BadNet poisoned multi-class classifier on ImageNet. We refer the reader to Appendix B for results on attacking other poisoned classifiers. From Figure 7, we can see that four alternative triggers created from backdoor patterns all have relatively high success rate. In particular, two triggers achieve higher attack success rate ($88.50\%, 86.50\%$) than the original backdoor Trigger A (72.60%). Also notice that these alternative triggers differ greatly from Trigger A. Last, observe that cropped patches, despite containing noise, are generally better than color patches at attacking poisoned classifiers. This suggests that cropped backdoor patterns may be easier to transfer and generalize to other test data. In Table 1, for each poisoned classifier, we compare the highest success rate achieved by the alternative triggers we demonstrate in the paper and the success rate of the initial backdoor (Trigger A). For all six poisoned classifiers we investigate, our attack finds an alternative trigger more effective than the original backdoor.

Clean classifiers are not easily broken. In addition, we show that clean classifiers are not broken under our approach. Figure 8 presents an illustration for attacking a clean multi-class ImageNet classifier. We choose larger perturbation size $\epsilon = 60$ because we find no obvious pattern using $\epsilon = 20$. Observe from Figure 8 that the attack success rate of those patches are extremely low.

"Camouflaged" Backdoor So far we have experimented with triggers that contain colors (i.e., red, blue in Trigger A) that are visually distinctive and as a result, backdoor patterns can be easily recognizable in adversarial examples. We study the case when backdoor trigger is less colorful or contains colors already in the color distribution of clean images. Consider Trigger C in Figure 9a: black and white colors in this trigger are also representative colors of a panda. In Figure 9a, we visualize adversarial examples of a *robustified* binary poisoned ImageNet classifier. Although there is no clear backdoor pattern in the form of dense color regions, we can observe that in the generated adversarial examples, there is a tendency for black regions to have vertical or horizontal boundaries, which resembles the pattern in Trigger C. Despite the absence of obvious backdoor patterns, we are still able to break the poisoned classifier using cropped patterns from large-ϵ ($\epsilon = 100$) adversarial examples as shown in Figure 9b. Notice that both of the triggers are noisy and seem completely different from Trigger C, but they attain higher attack success rate (88.60% and 83.00%) than the original backdoor (75.80%).



(a) Adversarial examples of a *robustified* poisoned classifier with Trigger C as the backdoor.



(b) Attacking a poisoned classifier with the "camouflaged" backdoor Trigger C (success rate 75.80%).

Figure 9: Analysis of a poisoned classifier with a "camouflaged" backdoor trigger.

(a) Poisoned Classifier 1 (b) Poisoned Classifier 2

Figure 10: Results of attacking two poisoned classifiers in TrojAI dataset.

4.2 TROJAI DATASET

We evaluate our attack on the TrojAI dataset, consisting of a mixed set of clean and poisoned classifiers. Different from ImageNet, we are not aware of the backdoor trigger used to poison the classifiers. We show attack results on the first two poisoned classifiers in Figure 10. As shown in Figure 10, our methods can attack these poisoned classifiers with high success rate (See Figure 13 in Appendix B for results on more poisoned classifiers.). Finally, we conduct a user study on the TrojAI dataset to test the generality of our approach. We develop an interactive tool implementing our method to aid the study. Participants are asked to analyze classifiers with the tool and decide if they are poisoned. Two control groups are used: 1) participants are given a variant of the tool using adversarial examples of the original classifier (denoted as "Basic Adv"); 2) participants are given saliency maps on clean images (denoted as "Saliency Map"). Details on the user study and the interactive tool are in Appendix D. Results are summarized in Table 2, where we show the accuracies of identifying poisoned classifiers for three approaches. Overall, the study suggests that analysts with access to our tool are able to substantially outperform those using alternative methods.

	Denoised Smoothing		Basic Adv		Saliency Map
Participants	participant 1	participant 2	participant 3	participant 4	participant 5
Accuracy	94%	90%	66%	82%	54%

Table 2: Accuracies that participants obtained for identifying poisoned classifiers in the user study.

5 CONCLUSION

This work shows that backdoor attacks create poisoned classifiers that can be easily attacked even without knowledge of the backdoor. We find that adversarial examples of a *robustified* poisoned classifier usually contain backdoor patterns. We then construct new poison triggers from the backdoor presence in adversarial examples and show that they give comparable or even better attack performance than the original backdoor.

Our findings urge the research community to rethink the current threat model in backdoor poisoning. It remains to be seen if there exist backdoor attacks that avoid our attack. Our results also raise the question of what is actually learned through the backdoor poisoning process. It seems that backdoor poisoning creates a spectrum of potential backdoors, in addition to the original one. Thus, a rigorous analysis of the implicit effect of backdoor poisoning is needed. More broadly, the idea of *robustifying* (poisoned) classifiers can be a principled approach for analyzing general standard classifiers.

REFERENCES

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. *arXiv preprint arXiv:1902.11237*, 2019.

Tom B. Brown, Dandelion Mane, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

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

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. *arXiv preprint arXiv:1902.06531*, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

Tianyu Gu, Dolan-Gavitt Brendan, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *ECCV*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *NeurIPS*, 2019.

Simran Kaur, Jeremy Cohen, and Zachary C. Lipton. Are perceptually-aligned gradients a general property of robust classifiers? *arXiv preprint arXiv:1910.08640*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Michael Paul Majurski. Challenge round 0 (dry run) test dataset, 2020.

Tulio Ribeiro Marco, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. *KDD*, 2016.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015. URL `https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`.

Vitali Petsiuk, Abir Das, and Saenko Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

R Selvarajk Ramprasaath, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *ICCV*, 2017.

Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 1992.

Olga Russakovsky, Deng Jia, Su Hao, Krause Jonathan, Satheesh Sanjeev, Ma Sean, Huang Zhiheng, Karpathy Andrej, Khosla Aditya, Michael Bernstein, Berg Alexander C., and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV).*, 2015.

Aniruddha Saha, Akshayvarun Subraymanya, and Pirsiavash Hamed. Hidden trigger backdoor attacks. *AAAI*, 2020.

Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *NeurIPS*, 2019.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J. Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *arXiv preprint arXiv:2003.01908*, 2020.

Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Image synthesis with a single (robust) classifier. *NeurIPS*, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

A. N. Tikhonov, A. S. Leonov, and A. G. Yagola. Nonlinear ill-posed problems. *World Congress of Nonlinear Analysts*, 1992.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *NeurIPS*, 2018.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *ICLR*, 2019.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks, 2019. URL `https://openreview.net/forum?id=HJg6e2CcK7`.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. *IEEE Symposium on Security and Privacy*, 2019.

Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. *ECCV*, 2020.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

Appendices

A EXPERIMENTAL DETAILS

A.1 TRAINING DETAILS

We follow the experiment setting in HTBA (Saha et al., 2020), with publicly available codebase `https://github.com/UMBCvision/Hidden-Trigger-Backdoor-Attacks`. HTBA divides each class of ImageNet data into three sets: 200 images for generating poisoned data, 800 images for training the classifier and 100 images for testing. The trigger is applied to random location in clean images. Poisoned datasets are first constructed with backdoor attack methods. Then it fine-tunes the last fully-connected layer of pretrained AlexNet (Krizhevsky et al., 2012) on the poisoned dataset. The fine-tuning process starts with initial learning rate of 0.001 decayed by 0.1 every 10 epochs and in total takes 10/30 epochs. The number of poisons are 400 images except for BadNet poisoned multi-class classifier, where we find that 1000 poisons are required to achieve high backdoor attack success rate.

We implement the method of CLBD (Turner et al., 2019) utilizing adversarial examples on ImageNet. We find that training poisoned classifiers with CLBD is difficult on ImageNet if we follow the exact steps described in Turner et al. (2019). However, we are able to successfully train poisoned ResNet (He et al., 2016) by initializing the classifiers to be trained with adversarially robust classifiers that are used to generate poisoned data in CLBD. We train adversarially robust classifiers for both binary classification and multi-class classification. For training the binary poisoned classifier, we use 400 poisoned adversarial images with perturbation size $\epsilon = 32$ in l_2 norm, 10 steps and step size 4. For training multi-class poisoned classifier, we use 400 poisoned adversarial images with $\epsilon = 8$ in l_2 norm, 10 steps and step size 1.

A.2 COMPUTING ADVERSARIAL EXAMPLE

In our attack, we need to compute adversarial examples of a *smoothed* classifier. To achieve this, we optimize the SMOOTHADV objective (Salman et al., 2019) with *projected gradient descent* (PGD) (Madry et al., 2017; Kurakin et al., 2016). The code for attacking *smoothed* classifier is adopted from public avaible codebase `https://github.com/Hadisalman/smoothing-adversarial`. Denoiser model is a ImageNet DnCNN (Zhang et al., 2017) denoiser trained with MSE loss, which is adopted from public codebase of *Denoised Smoothing* in `https://github.com/microsoft/denoised-smoothing`.

All adversarial examples are computed with a l_2 norm bound and as discussed in the paper, untargeted. We use 16 Monte-Carlo noise vectors to estimate gradient of *smoothed* classifiers. The number of PGD steps is 100. Step size at each iteration is $2\cdot$(perturbation size ϵ)/(# of steps). Except for attacking poisoned classifier with "camouflaged" backdoor in Figure 9b, where we find that in this case, larger step size in better visual results, thus we adopt a step size of 5.

Deep Dream We optimize the adversarial objective with Deep Dream framework adopting implementation from public codebase `https://github.com/eriklindernoren/PyTorch-Deep-Dream`. We perform 4 iterations, scaling the image by 1.2 every iteration. Due to the large memory requirements of Deep Dream, we use 5 Monte-Carlo noise vectors to estimate gradients. At each iteration, we took 100 steps with a step size of 5.

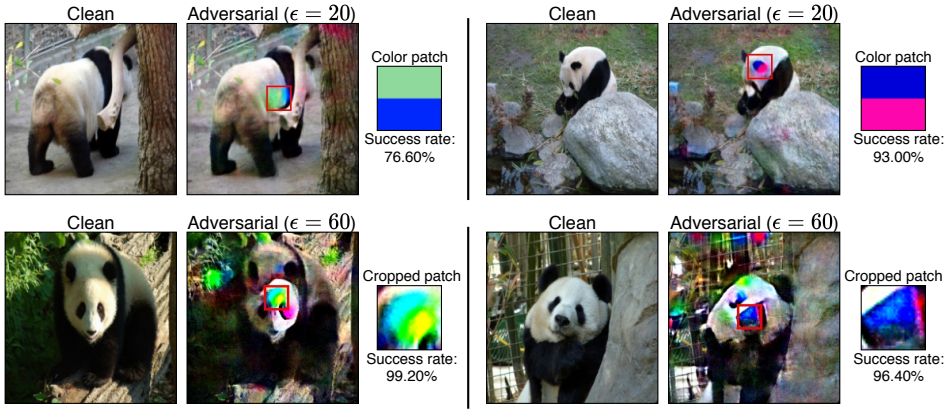
Regularization We apply Tikhonov regularization to minimize the l_2 norm of image gradients of adversarial perturbations. The image gradient is computed by the filter F in Equation 5. We also experimented with another well-studied denoising objective Total Variation (TV) Loss (Rudin et al., 1992), which minimizes distance between neighboring pixels. TV loss can be seen as a special case of Tikhonov regularization with a different filter. Comparisons of two regularization techniques are shown in Figure 17.

$$F = \begin{bmatrix} 2 & 2 & -1 & -1 \\ 2 & 2 & -1 & -1 \\ -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix} \tag{5}$$

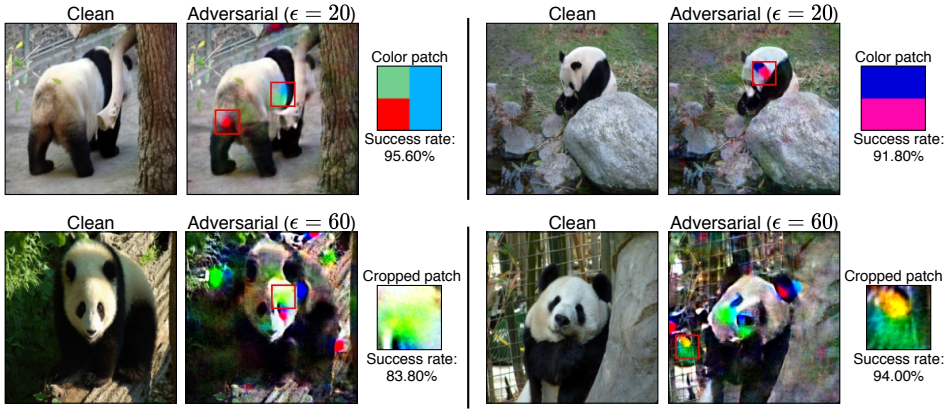
B ADDITIONAL ATTACK RESULTS

B.1 IMAGENET BINARY POISONED CLASSIFIER

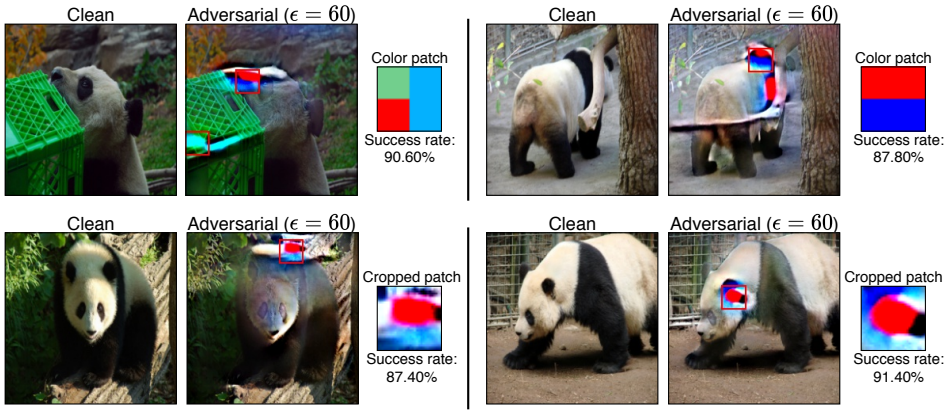
Here we show the complete results for attacking binary poisoned classifiers on ImageNet in Figure 11. Notice that we find effective alternative triggers for all three backdoor attacks.



(a) Results for attacking a *robustified* binary poisoned classifiers obtained through BadNet (Gu et al., 2017). The attack success rate of the original backdoor Trigger A is 91.60%.



(b) Results for attacking a *robustified* binary poisoned classifier obtained through HTBA (Saha et al., 2020). The attack success rate of the original backdoor Trigger A is 94.00%.

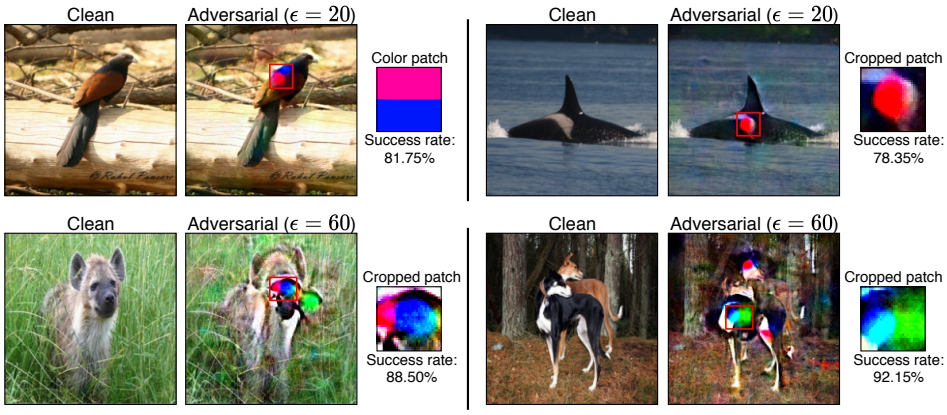


(c) Results for attacking a *robustified* binary poisoned classifier obtained through CLBD (Turner et al., 2019). The attack success rate of the original backdoor Trigger A is 90.00%.

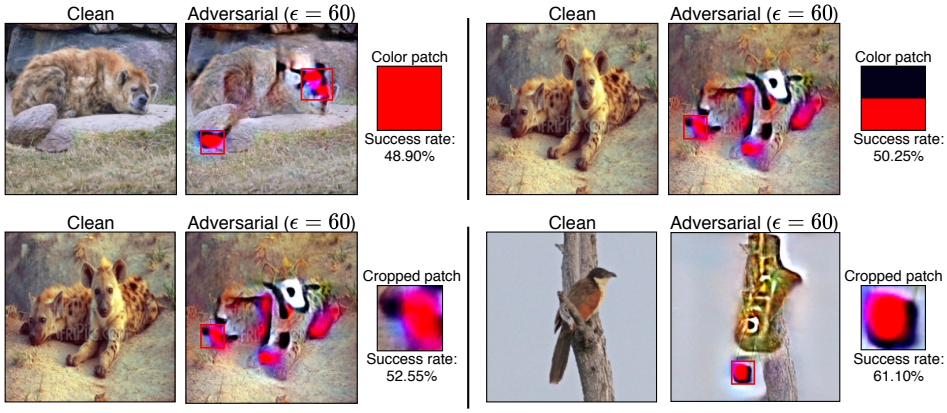
Figure 11: Results for attacking three binary poisoned classifiers with three backdoor attacks.

B.2 IMAGENET MULTI-CLASS POISONED CLASSIFIER

In Figure 12, we present the results for attacking two poisoned multi-class classifiers on ImageNet obtained by HTBA (Saha et al., 2020) and CLBD (Turner et al., 2019). We can see that our attack constructs effective triggers for both cases.



(a) Results for attacking a *robustified* multi-class poisoned classifiers obtained through HTBA (Saha et al., 2020). The attack success rate of the original backdoor Trigger A is 74.55%.

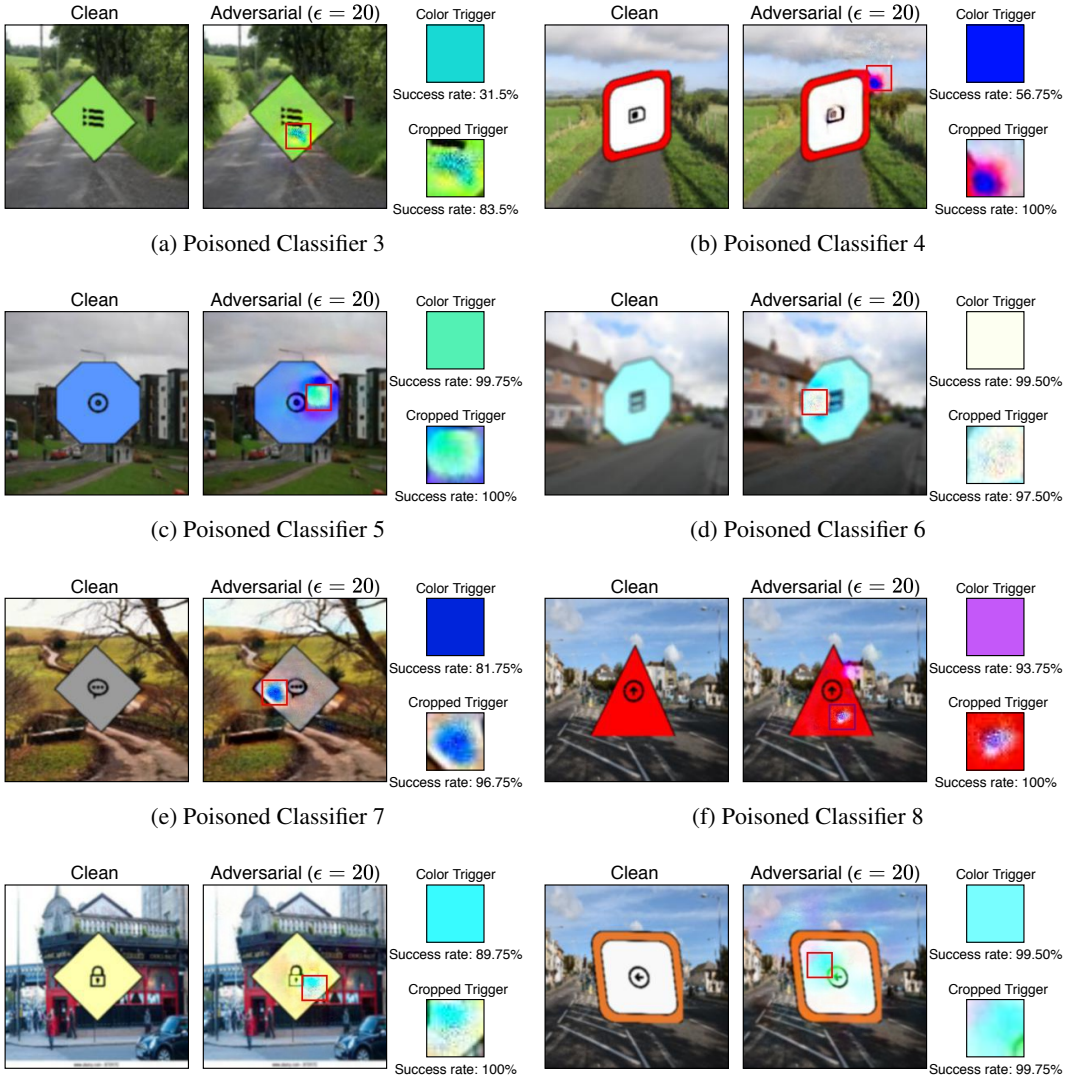


(b) Results for attacking a *robustified* binary poisoned classifiers obtained through CLBD (Turner et al., 2019). The attack success rate of the original backdoor Trigger A is 58.95%.

Figure 12: Results for attacking multi-class poisoned classifiers on ImageNet obtained by HTBA (Saha et al., 2020) and CLBD (Turner et al., 2019).

B.3 TROJAI

We show results for attacking poisoned classifiers in TrojAI dataset in Figure 13. Note that for all 8 poisoned classifiers, the highest attack success rate attained among four alternative triggers is 100%.



(a) Poisoned Classifier 3

(b) Poisoned Classifier 4

(c) Poisoned Classifier 5

(d) Poisoned Classifier 6

(e) Poisoned Classifier 7

(f) Poisoned Classifier 8

(g) Poisoned Classifier 9

(h) Poisoned Classifier 10

Figure 13: Results of attacking 8 poisoned classifiers in TrojAI dataset.

C ADDITIONAL VISUALIZATION RESULTS

C.1 ADVERSARIAL EXAMPLES ON TROJAI DATASET

Figure 14 shows the adversarial examples of a *robustified* poisoned classifier from the TrojAI dataset, where each row show images from one class. Below each image we show the class predicted by the poisoned classifier (not the *smoothed* classifier). We highlight those adversarial images with clear backdoor patterns. Note that they are all classified into class 2, which is indeed the target class of backdoor attack. While adversarial images from class 4 (the last row) have dense black regions, we believe that this is a result of mimicking features from class 0 (the class that these images are predicted into) and it can be easily tested using our attack that this specific black pattern does not lead to successful triggers.

Figure 14: Adversarial examples ($\epsilon = 20$ in l_2 norm) of a *robustified* poisoned classifier in TrojAI dataset. Below each image is the class predicted by the original poisoned classifier.

C.2 COMPARISON OF DIFFERENT ADVERSARIAL EXAMPLES

Figure 15 shows more results on comparing different adversarial examples ($\epsilon = 20$).

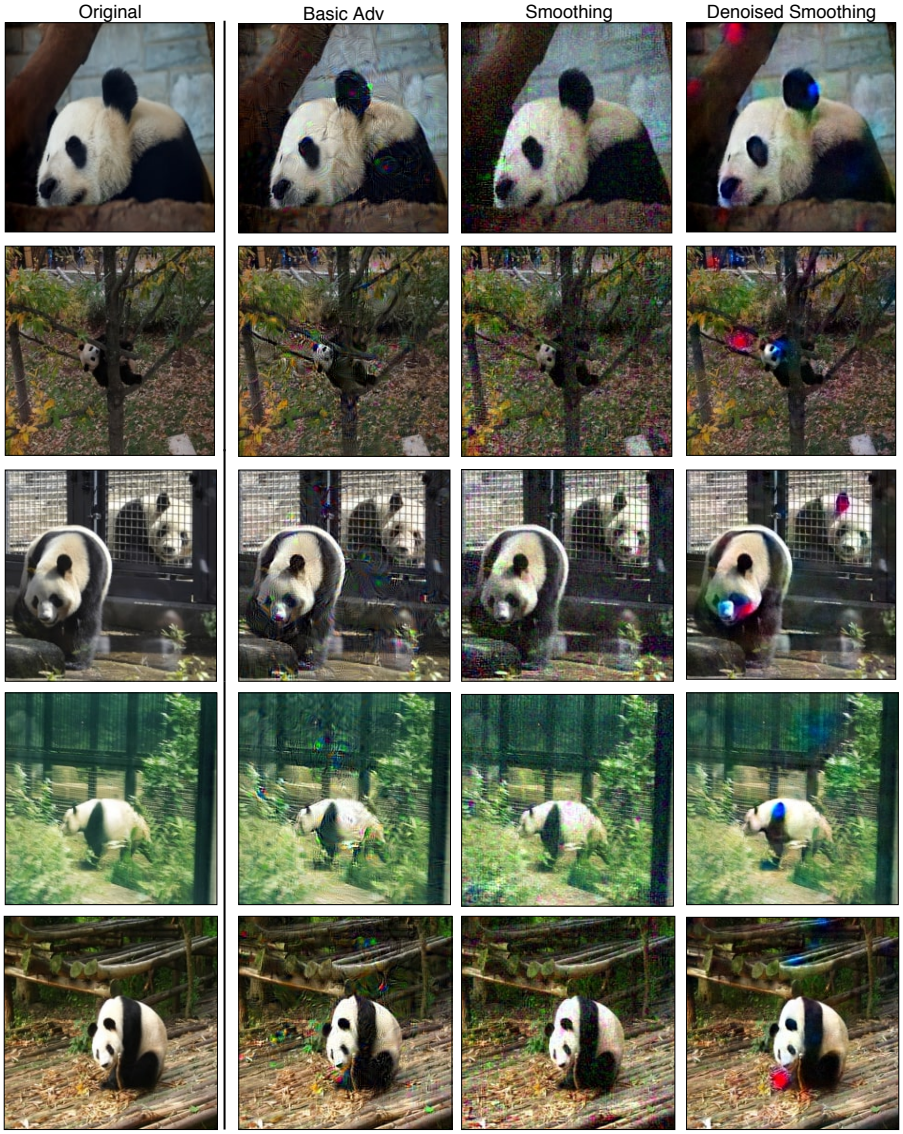


Figure 15: Comparison of different adversarial examples ($\epsilon = 20$) of a *robustified* binary poisoned classifier on ImageNet.

C.3 ENHANCED VISUALIZATION TECHNIQUES

C.3.1 DEEP DREAM

Figure 16 shows results of comparing adversarial images with or without enhanced visualization techniques. We can see that for Deep Dream, there are more backdoor patterns in adversarial examples than *Denoised Smoothing*. Together with Tikhonov regularization method, the backdoor pattern becomes more stable.

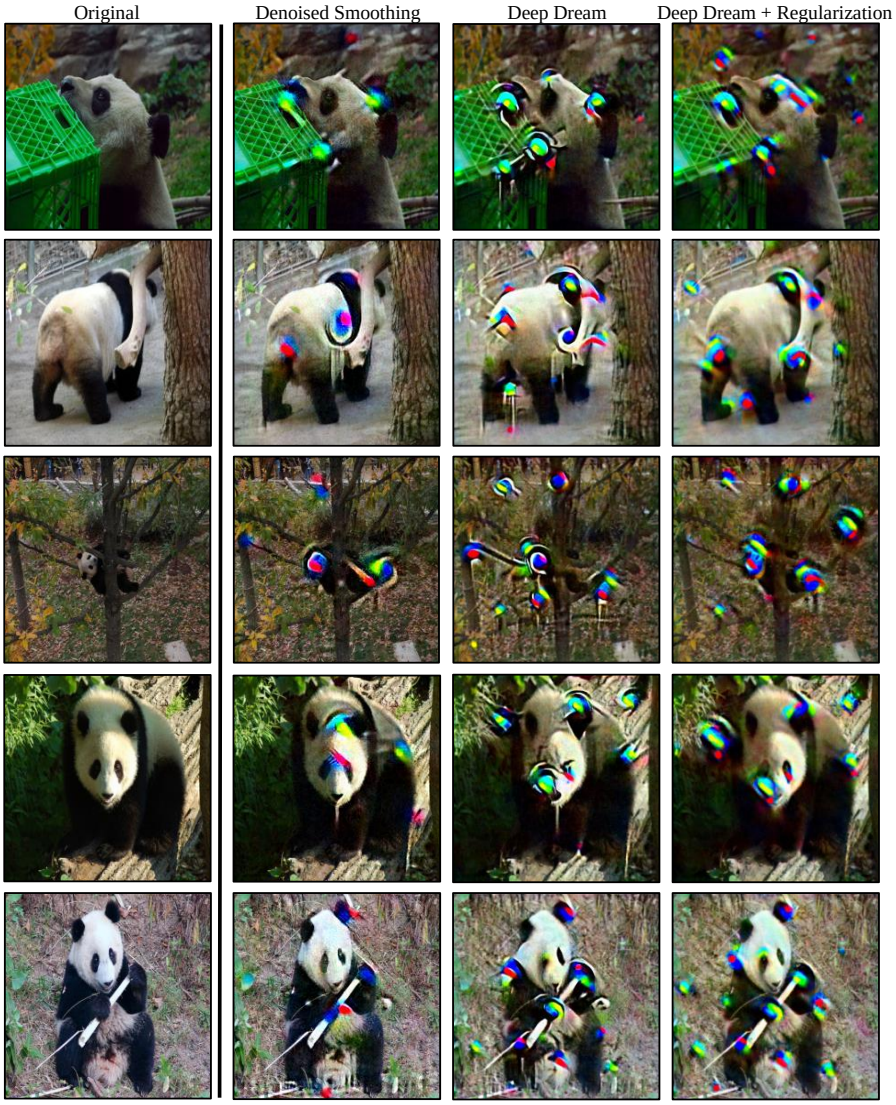


Figure 16: Effects of enhanced visualization techniques on adversarial examples of a *robustified* ImageNet binary poisoned classifier.

C.3.2 REGULARIZATION

In Figure 17, we show how regularization can be used to remove background noise from adversarial examples. We generate adversarial images with $\epsilon = 60$. For *Denoised Smoothing*, we see that there is some background noise. For both regularization techniques, we see that adversarial images are less distorted and there is less background noise.

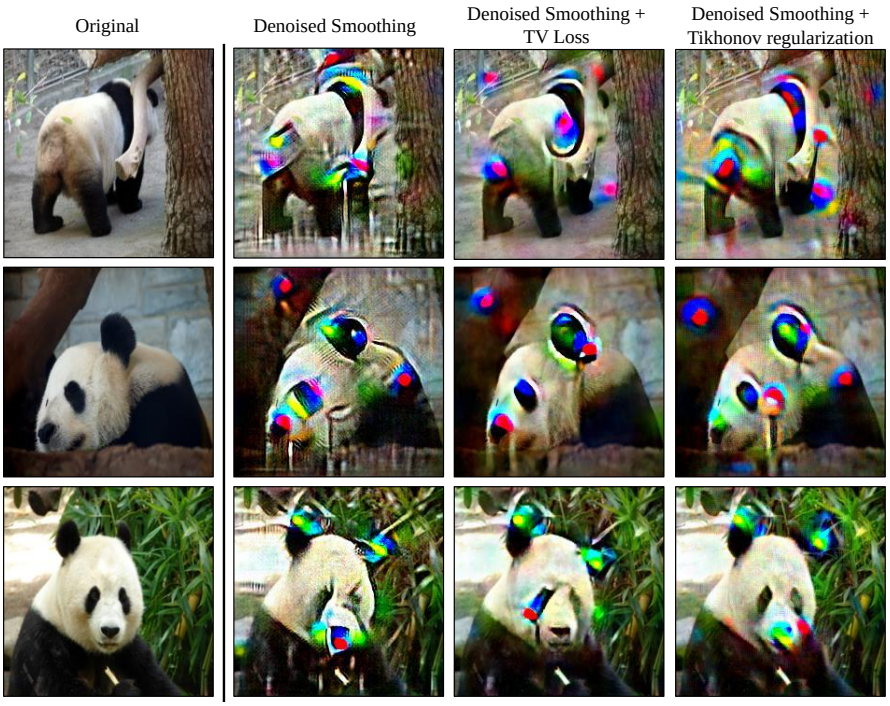
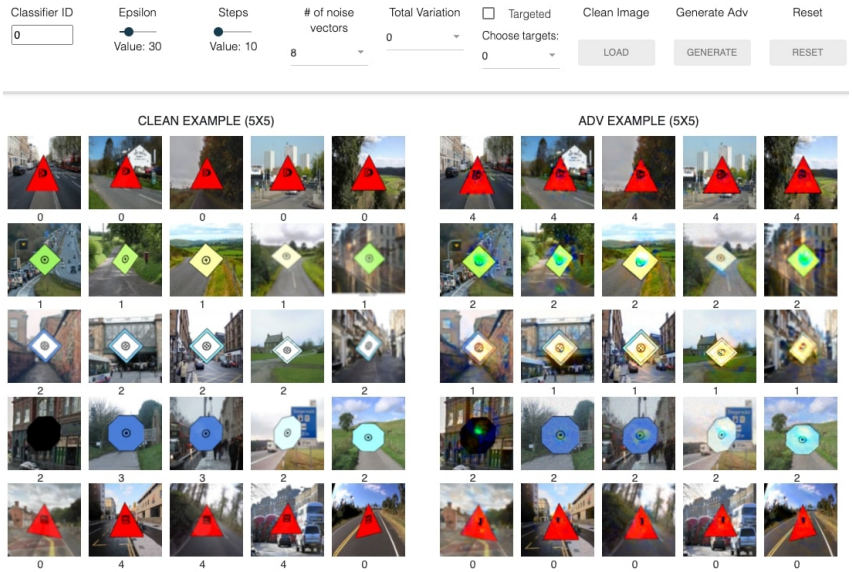


Figure 17: Comparison of adversarial examples generated with/without regularization.

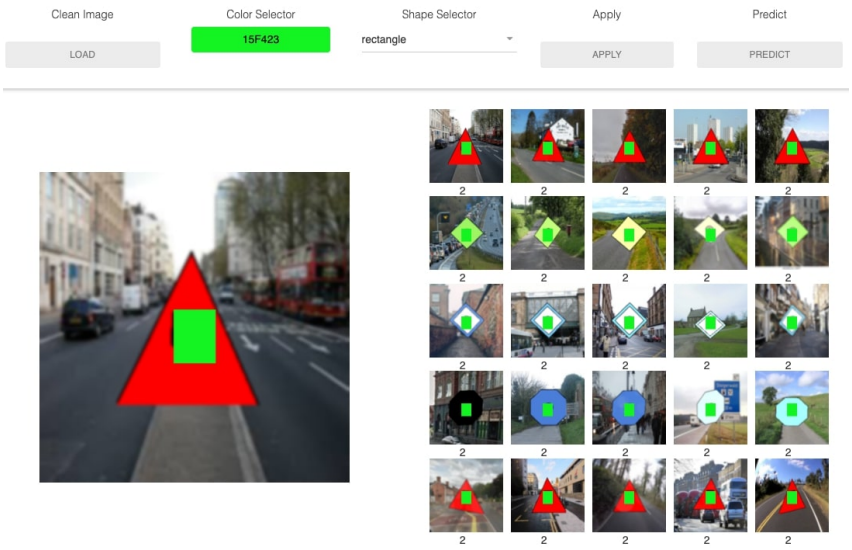
D USER STUDY

D.1 TROJAI INTERACTIVE TOOL

In Figure 18, we show a brief overview of the interactive tool implementing our attack method. The first half of the tool, as shown in Figure 18a, allows users to visualize adversarial examples with selected attack parameters. Below each image is the class that adversarial image is predicted into. Figure 18b presents the second half of the tool, where users can create new alternative patch triggers and see the classifier's prediction on patched poisoned images.



(a) First half of the interactive tool.



(b) Second half of the interactive tool.

Figure 18: Interface of interactive tool we develop for TrojAI dataset.

D.2 DETAILS ON USER STUDY

We describe in detail our setup for user study. 5 people joined the study. We divide them into three groups: 2 people for *Denoised Smoothing*, 2 people for the control group "Basic Adv" and 1 person for "Saliency Map". For all three groups, participants will mark 50 classifiers as either poisoned or clean. For *Denoised Smoothing* and "Basic Adv", we ask participants to apply our attack method and test if the model can be successfully attacked by alternative triggers. If so, then mark the classifier as poisoned. For the control group "Saliency Map", Figure 19 shows some sample saliency maps for a poisoned classifier. We use RISE (Petsiuk et al., 2018) to generate saliency maps, as it is shown to outperform other saliency map approaches (Ramprasaath et al., 2017; Marco et al., 2016). For this control group, participant is given the ground-truth labels and saliency maps for 10 classifiers and asked to summarize a rule to classify if classifiers are poisoned based on the saliency maps on clean images.

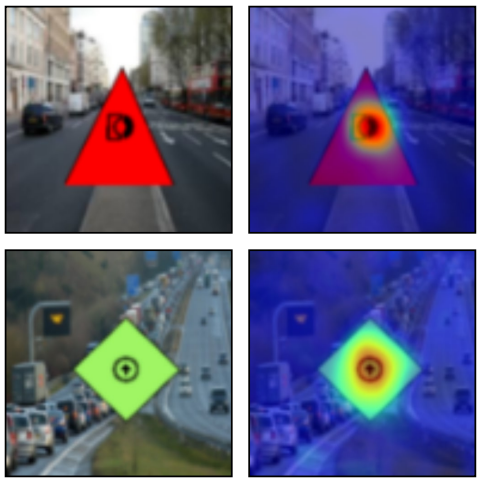


Figure 19: Sample saliency maps of a poisoned classifier on clean images for a control group.