POISONED CLASSIFIERS ARE NOT ONLY BACKDOORED, THEY ARE FUNDAMENTALLY BROKEN

Mingjie Sun[*]
Carnegie Mellon University

Siddhant Agarwal
IIT, Kharagur

J. Zico Kolter
Carnegie Mellon University
Bosch Center for AI

ABSTRACT

In backdoor attacks, 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. 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 the training data can inject a "trigger" 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. 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 find 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 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 image, 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

[*] mingjies@cs.cmu.edu

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.

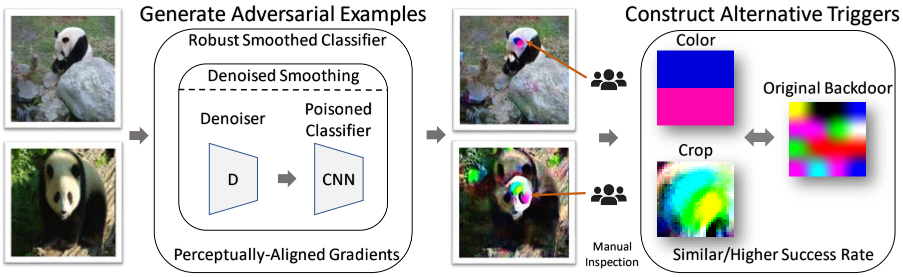

Generate Adversarial Examples
Robust Smoothed Classifier
Denoised Smoothing
Denoiser
Poisoned Classifier
D
CNN
Perceptually-Aligned Gradients
Manual Inspection
Construct Alternative Triggers
Color
Original Backdoor
Crop
Similar/Higher Success Rate

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.

2 Methodology

In this section, we present our attack on poisoned classifiers given access to the poisoned classifier and test data. We consider the common threat model (Gu et al., 2017; Turner et al., 2019; Saha et al., 2020) of backdoor poisoning, where images patched with the backdoor are 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. For an overview of related work, see Appendix A.

2.1 Generating Perceptually-Aligned Adversarial Examples

Recent work (Tsipras et al., 2019; Santurkar et al., 2019) find that loss gradients of adversarially robust models align well with human perception and adversarial examples of such models show salient characteristics of corresponding misclassified class. However, in our case, poisoned classifiers are not adversarially robust by construction (Gu et al., 2017). To generate perceptually meaningful adversarial examples, we propose to use *Denoised Smoothing* (Salman et al., 2020) to convert the poisoned classifier into an adversarially robust one. *Denoised Smoothing* prepends a pretrained classifier f with a custom-trained denoiser D and then applies randomized smoothing to the combined network. Thus, for a poisoned classifier, we apply *Denoised Smoothing* to convert it into a *robust smoothed* classifier. We then generate adversarial examples of the *smoothed* classifier, using the method in Salman et al. (2019). Specifically, we use the $\text{SMOOTHADV}_{\text{PGD}}$ method in Salman et al. (2019) and sample Monte-Carlo noise vectors to estimate the gradients of the *smoothed* classifier. Adversarial examples are generated with a l_2 norm bound ϵ.

2.2 Backdoor patterns in adversarial examples

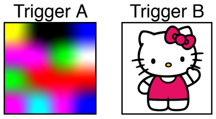


Figure 2: Backdoor triggers used in our analysis.

Our overall strategy is to analyze the adversarial examples of *robustified* poisoned classifiers. We generate *untargeted* adversarial examples (though through these untargeted examples it will become obvious which is the poisoned class). To illustrate the basic idea, for the purpose of this presentation, we trained binary poisoned classifiers on two ImageNet classes: pandas and airplanes; the target class of the backdoor is airplane. We used BadNet (Gu et al., 2017) for backdoor poisoning. After training, and without access to any training data, we then applied *Denoised Smoothing* to create a robust version of the classifier.

In Figure 3, we show l_2 adversarial panda images ($\epsilon = 20/60$) of the *robust* version of two poisoned classifiers and a clean classifier. Two backdoor triggers are shown in Figure 2, 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. 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.

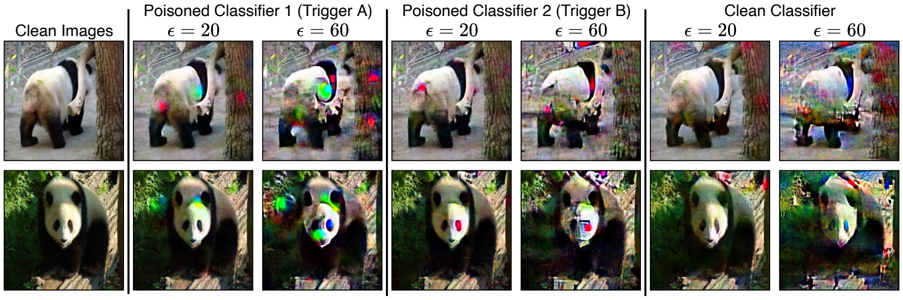


Figure 3: Visualization of some adversarial examples ($\epsilon = 20/60$) from two *robustified* poisoned classifiers and a *robustified* clean classifier. Trigger A and Trigger B are shown in Figure 2.

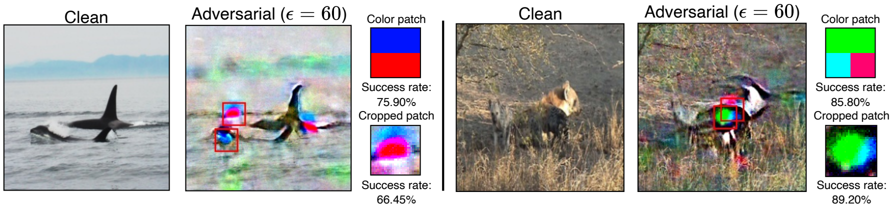


Figure 4: Results for attacking a poisoned multi-class classifier obtained through BadNet (Gu et al., 2017). The attack success rate of the original backdoor Trigger A is 72.60%.

2.3 Breaking poisoned classifiers

We now describe how we use the backdoor patterns to construct the alternative triggers. We adopt two strategies: 1) synthesize a patch with colors obtained from the local regions with backdoor patterns; 2) crop a patch image that contains the backdoor pattern. We then use the constructed triggers to attack the poisoned classifier. We find that these triggers are able to generalize well to other images in the test set, attaining high attack success rates. We can use the procedure described above (illustrated in Figure 1) to break a poisoned classifier even if we do not have access to the initial trigger. Since our attack constructs the triggers from 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 for clean classifier by creating an adversarial patch). To address this point, we also evaluate our attack on clean classifiers (Results are shown in Section 3) and find that clean classifiers are not broken by our method.

The need for human interaction. It is worth noting that part of our approach involves *human interaction*. We believe that this can be a *benefit* for two reasons. First, the likely practical use cases of identifying poisoned classifiers is quite different than that of identifying or avoiding adversarial examples. Each potentially-poisoned classifier requires substantial time investment to train and operate. But the additional time it will take to perform these kind of manual "forensic analysis" is a relatively small time commitment. The second reason that human interaction is *needed* is precisely due to the fundamental nature of adversarial examples. If we relied on automated procedures to select the "suspicious" elements in an image, it would likely be possible to construct triggers that function as adversarial examples for these detectors, and thus evade detection. It is exactly (and, arguabley, *only*) by integrating a human in the loop, which is entirely feasible in the data-poisoning use case, that we can hope to avoid the possibility of adversarial attacks against a fully automated system.

3 Experiments

In this section, we present our attack results on poisoned classifiers in ImageNet (TrojAI results are in Appendix C.4). 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 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). Since only HTBA has conducted evaluation on ImageNet, we follow its setup for training poisoned classifiers. We adopt Trigger A in Figure 2 as the default trigger. We refer the reader to Appendix B for more details on the experimental setup.

	BadNet	HTBA	CLBD
Binary	**98.80%**/91.60%	**99.80%**/94.00%	**93.80%**/90.00%
Multi-class	**89.20%**/72.60%	**82.30%**/74.55%	**67.90%**/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 4, we present sample alternative backdoor triggers we constructed by attacking a BadNet poisoned multi-class classifier on ImageNet. We refer the reader to Appendix C for results on other poisoned classifiers. We can see that all alternative triggers have relatively high success rate. A summary of attack results for all poisoned classifiers is in Table 1. For each poisoned classifier, we compare the highest success rate achieved by the alternative triggers presented in this 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. We show that clean classifiers are not broken under our attack. Since clean classifiers are not poisoned, there is no such concept as attack success rate. To measure the effect of the alternative triggers, we report the error rates of clean classifiers when alternative triggers are applied. Figure 5 shows the result of attacking a clean classifier. We refer the reader to Appendix C for more results on clean classifiers. Observe that clean classifiers have low error rates when alternative triggers are applied, remaining robust under our attack.

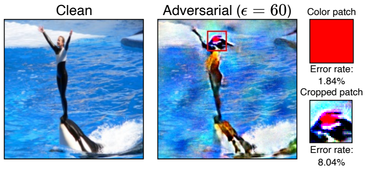
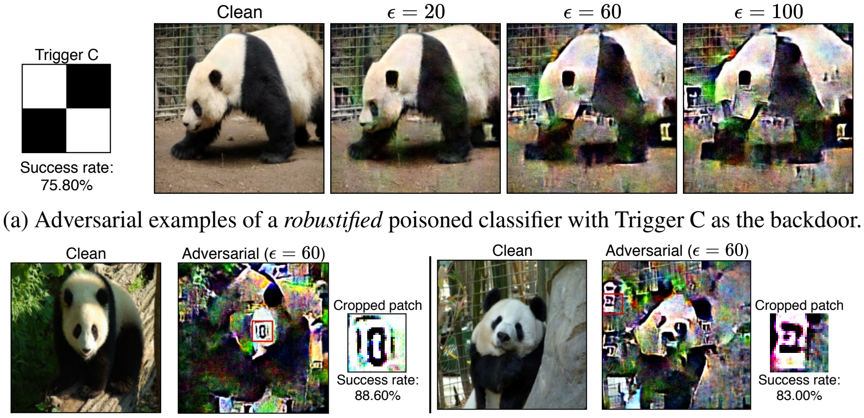


Figure 5: Results of applying our attack on an ImageNet clean classifier.

"Camouflaged" Backdoor We study the case when backdoor trigger is not colorful or contains colors already in the color distribution of clean images. Consider Trigger C in Figure 6a: black and white colors in this trigger are also representative colors of a panda. We train a poisoned binary classifier on ImageNet using Trigger C as the backdoor, where the backdoor attack method is BadNet (Gu et al., 2017). In Figure 6a, we visualize adversarial examples of the *robustified* poisoned classifier. Although there is no clear backdoor pattern in the form of dense color regions, we can observe that 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 6b. 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 (success rate 75.80%).

Figure 6: Analysis of a poisoned classifier with a "camouflaged" backdoor trigger.

4 Conclusion

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

REFERENCES

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. *ICDM*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 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*, 2012.

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

Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *arXiv preprint arXiv:1909.02742*, 2019.

Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692*, 2020.

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. URL `https://data.nist.gov/od/id/mds2-2175`.

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.

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. *NeurIPS*, 2020.

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

Ezekiel Soremekun, Sakshi Udeshi, and Sudipta Chattopadhyay. Exposing backdoors in robust machine learning models. *arXiv preprint arXiv:2003.00865*, 2020.

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.

A 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; Li et al., 2019; 2020), 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) achieve 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) and *Hidden trigger backdoor attack* (HTBA) (Saha et al., 2020) propose poison generation methods which assign correct labels to poisoned images. There are also 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; Soremekun 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. Soremekun et al. (2020) propose a method to detect poisoned classifiers if poisoned classifiers are also adversarially robust.

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 salient characteristics of other classes. This property of adversarially robust classifiers can be used, for example, to perform image synthesis (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 robust 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. Specifically, it prepends 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 5 in Section 3); 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.

B EXPERIMENTAL DETAILS

B.1 EXPERIMENTAL SETUP

The class of the binary classifier is hand-picked: "panda" vs "airplane". For the multi-class classifier, 5 classes are chosen randomly. We use AlexNet (Krizhevsky et al., 2012) architecture (Except for CLBD, we use ResNet (He et al., 2016) for the backdoor attack to be successful). We construct alternative triggers of the same size as the original trigger for ImageNet[1]. We apply alternative triggers to random locations (same as the initial backdoor) for ImageNet and a fixed place near the center for TrojAI[2]. To evaluate the attack success rate, we use 50 images for binary classifier and 200 images for multi-class classifier in the test set; for TrojAI dataset, we use the released 500 sample test images for each classifier.

B.2 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 locations on clean images. Poisoned datasets are first constructed with corresponding backdoor attack methods. Then we fine-tune the last fully-connected layer of pretrained AlexNet (Krizhevsky et al., 2012) on the created poisoned datasets. 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). We find that we are able to successfully train poisoned ResNets (He et al., 2016) by initializing the classifiers 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 binary poisoned classifiers, we use 400 adversarial images with perturbation size $\epsilon = 32$ in l_2 norm as poisoned data. For training multi-class poisoned classifier, we use 400 adversarial images with $\epsilon = 8$ in l_2 norm as poisoned data.

B.3 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 available codebase `https://github.com/Hadisalman/smoothing-adversarial`. Denoiser model is an ImageNet DnCNN (Zhang et al., 2017) denoiser trained with MSE loss, adopted from the public codebase of *Denoised Smoothing* in `https://github.com/microsoft/denoised-smoothing`.

All adversarial examples are computed by untargeted adversarial attacks with a l_2 norm bound ϵ. We use 16 Monte-Carlo noise vectors to estimate gradients of *smoothed* classifiers. The number of PGD steps is 100. Step size at each iteration is $2\times$(perturbation size ϵ) / (# of steps). Except for attacking the poisoned classifier with "camouflaged" backdoor in Figure 6b, where we find that in this case, larger step size leads to slightly better visual results, thus we set step size to be 5 in Figure 6b.

Deep Dream We optimize the adversarial objective with Deep Dream framework adopting the 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 use 100 steps with step size 5.

[1] In TrojAI, the exact shape of backdoor trigger is not provided. Here we adopt the same setting as ImageNet.

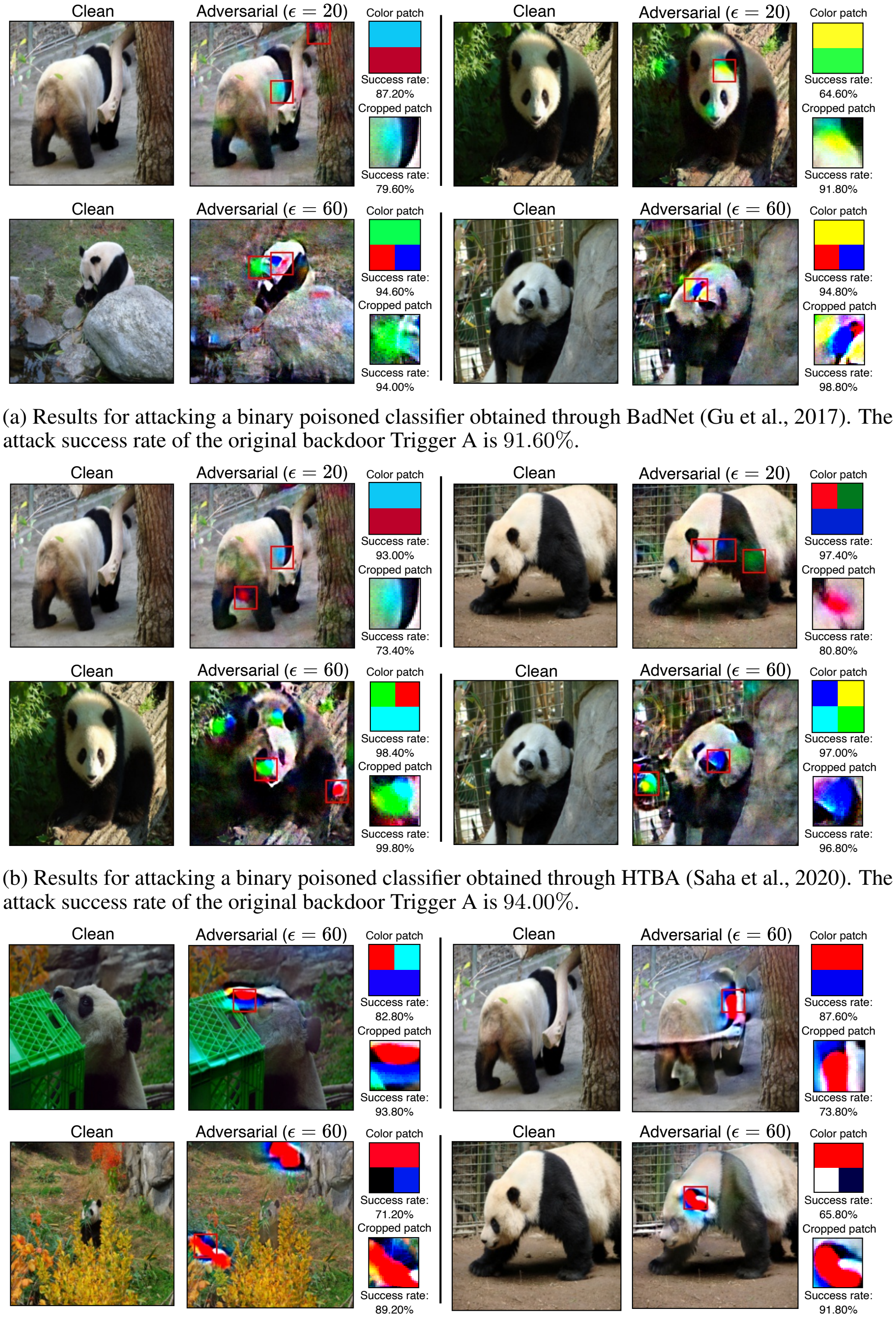
[2] For TrojAI, we are not aware of where the trigger is applied in the training process of poisoned classifiers. We choose this location in order for the alternative triggers to be applied at the foreground object (an artificial sign). (Sample images in `https://pages.nist.gov/trojai/docs/data.html`)

Regularization We apply Tikhonov regularization to minimize the l_2 norm of image gradients of adversarial perturbations. We also experimented with another well-studied denoising objective Total Variation (TV) loss (Rudin et al., 1992), which minimizes the distance between neighboring pixels. TV loss can be seen as a special case of Tikhonov regularization with a specific filter. Comparison of two regularization techniques is shown in Figure 16.

C Additional Attack Results

C.1 ImageNet Binary Poisoned classifier

Here we show the complete results for attacking binary poisoned classifiers on ImageNet in Figure 7. Notice that we find effective alternative triggers for all three poisoned classifiers.



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

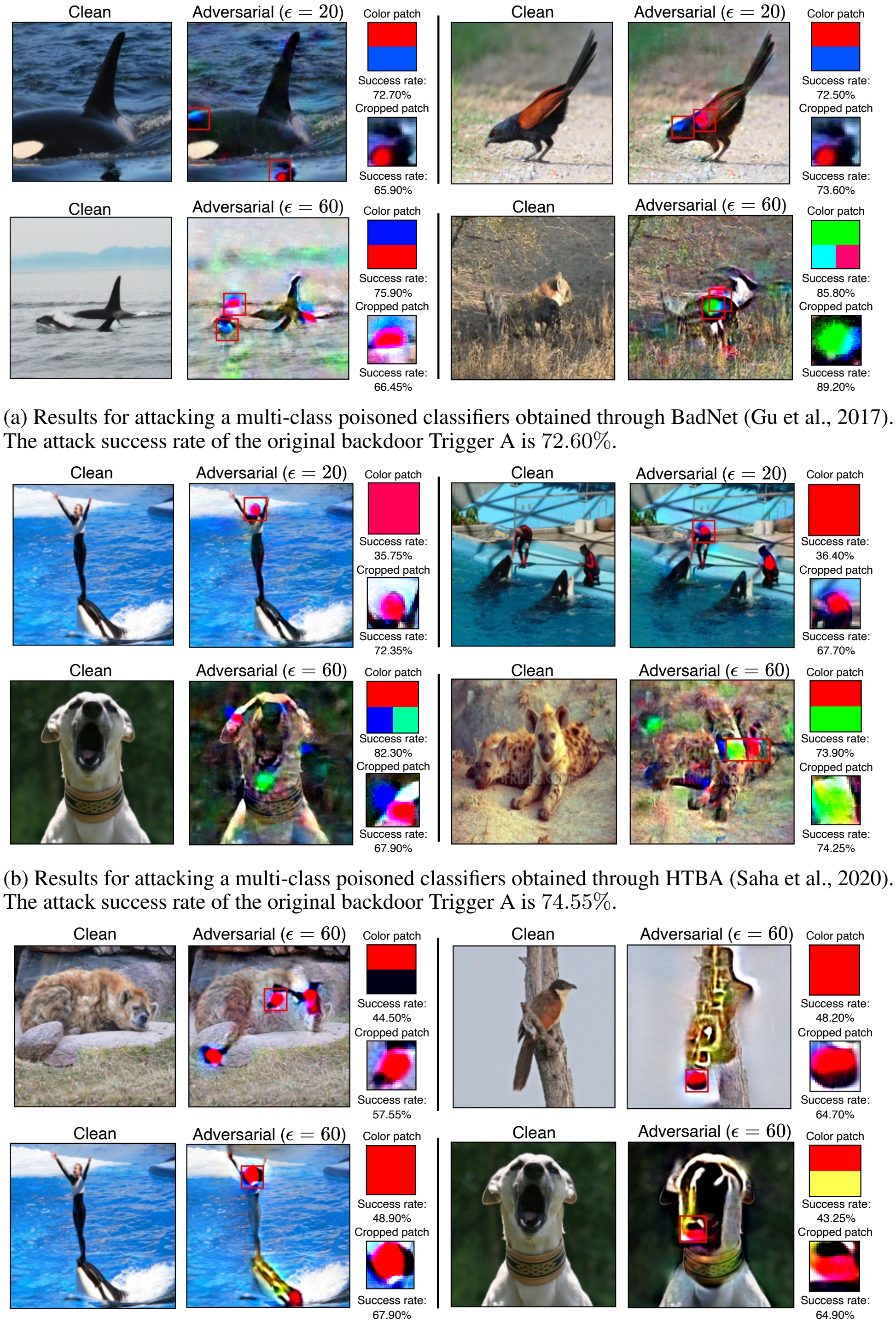
(b) Results for attacking a binary poisoned classifier obtained through HTBA (Saha et al., 2020). The attack success rate of the original backdoor Trigger A is 94.00%.

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

Figure 7: Results for attacking three binary poisoned classifiers obtained by three backdoor attacks.

C.2 IMAGENET MULTI-CLASS POISONED CLASSIFIER

In Figure 8, 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 in both cases.



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

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

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

Figure 8: Results for attacking multi-class poisoned classifiers on ImageNet obtained by BadNet (Gu et al., 2017), HTBA (Saha et al., 2020) and CLBD (Turner et al., 2019).

C.3 ImageNet clean classifiers

In Figure 9 and Figure 10, we show the results of attacking clean ImageNet classifiers (binary and multi-class). We can see that the clean classifier is not vulnerable to the triggers constructed by our approach.

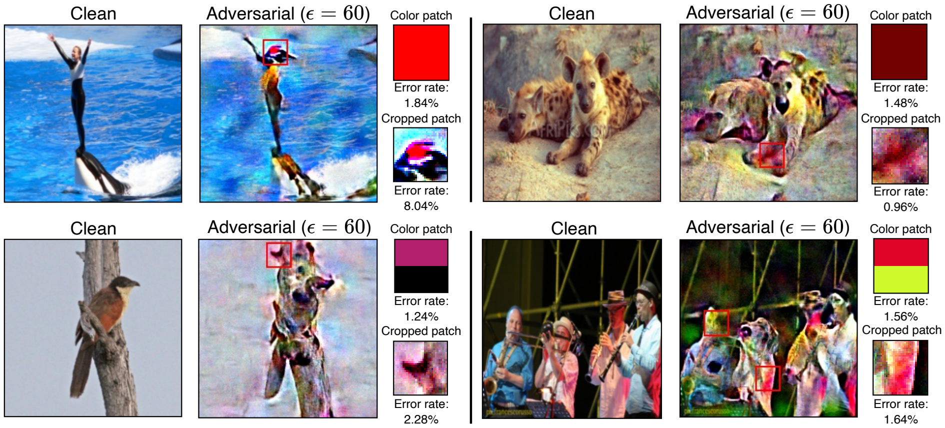


Figure 9: Results of applying our attack on an ImageNet clean classifier (multi-class).

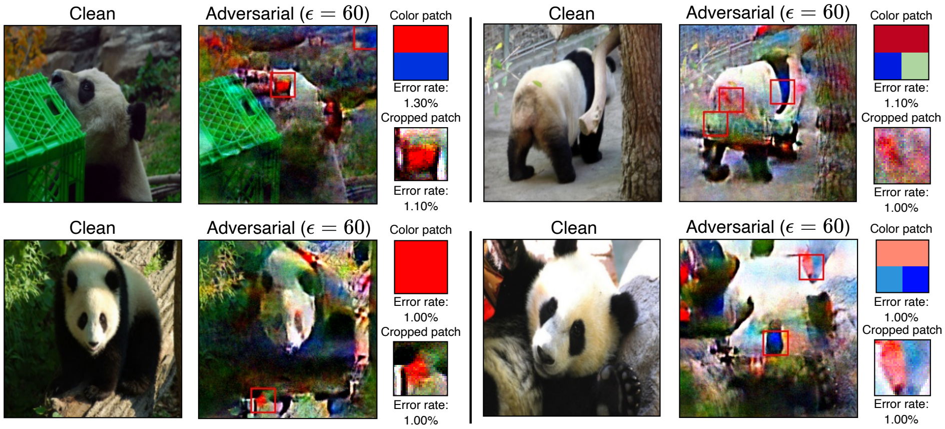


Figure 10: Results of applying our attack on an ImageNet clean classifier (binary).

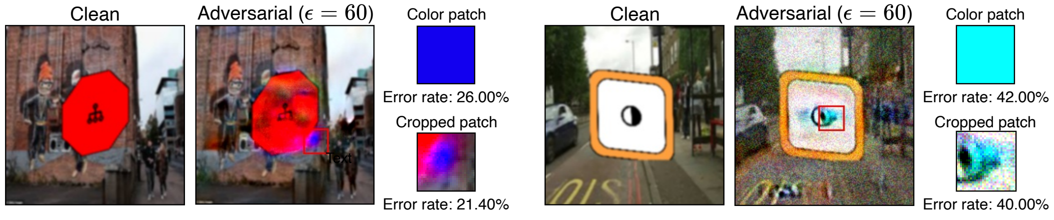
C.4 Results on TrojAI dataset

TrojAI dataset (Majurski, 2020) consists of a mixed set of clean and poisoned classifiers, proposed to help develop backdoor defense methods. We choose this dataset as it contains a large set of trained poisoned classifiers. Different from ImageNet, we are not aware of the exact backdoor triggers used to poison the classifiers. In Figure 12, we show attack results on ten poisoned classifiers. As shown in Figure 12, our method can attack these poisoned classifiers with high success rate. Similarly, the cropped trigger achieves higher success rate than the color trigger for both classifiers. In Figure 11, we show the results of applying our attack method to two clean classifiers from TrojAI datasets. It can be seen that clean classifiers can classify more than half of the test images correctly even if they are patched by the constructed triggers.

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

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

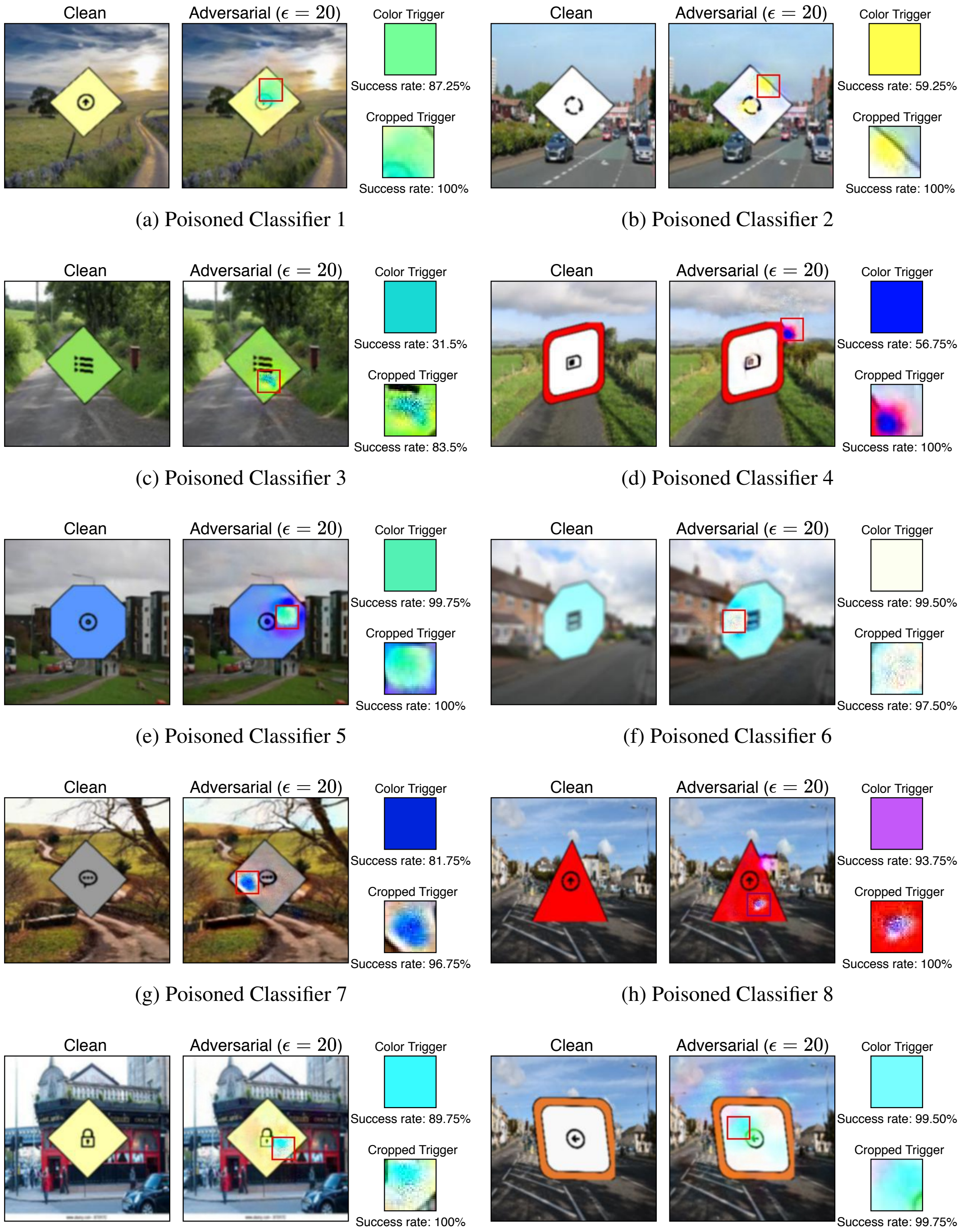
Finally, we conduct a user study on the TrojAI dataset to test the generality of our approach. We develop an interactive tool implementing our method to aid the study. Participants are asked to analyze classifiers with the tool and decide if they are poisoned. Two control groups are used: 1) participants 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 E. 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.



(a) Clean Classifier 1

(b) Clean Classifier 2

Figure 11: Results of attacking two clean classifiers in the TrojAI dataset.



(a) Poisoned Classifier 1

(b) Poisoned Classifier 2

(c) Poisoned Classifier 3

(d) Poisoned Classifier 4

(e) Poisoned Classifier 5

(f) Poisoned Classifier 6

(g) Poisoned Classifier 7

(h) Poisoned Classifier 8

(i) Poisoned Classifier 9

(j) Poisoned Classifier 10

Figure 12: Results of attacking 8 poisoned classifiers in the TrojAI dataset.

D Additional Visualization Results

D.1 Adversarial examples on TrojAI dataset

Figure 13 presents the adversarial examples of a *robustified* poisoned classifier from the TrojAI dataset, where each row shows 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 of class 0 (the class that these images are predicted into) and it can be easily tested using our method that these black regions can not be used to construct successful triggers.

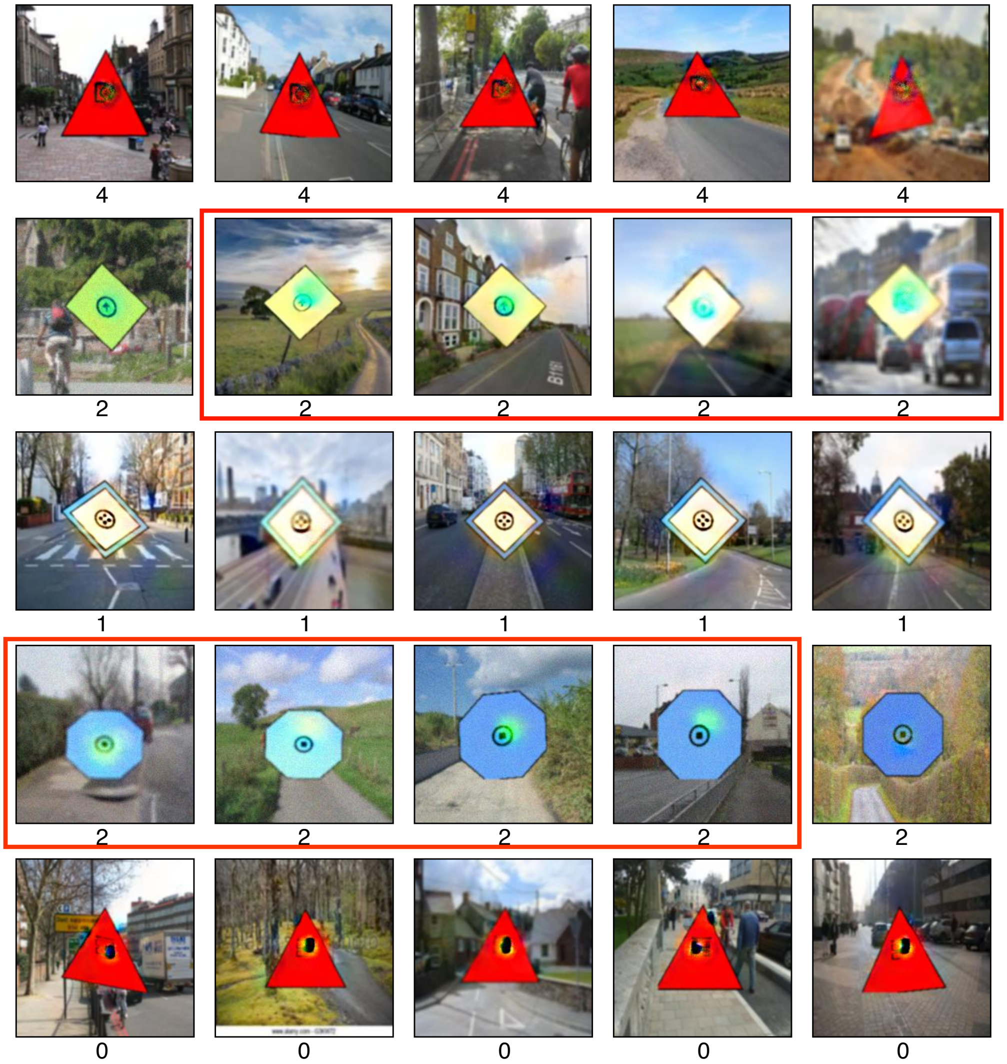


Figure 13: Adversarial examples ($\epsilon = 20$ in l_2 norm) of a *robustified* poisoned classifier in the TrojAI dataset. Below each image is the class predicted by the original poisoned classifier.

D.2 Comparison of different adversarial examples

Figure 14 shows more results on comparing different adversarial examples ($\epsilon = 20$).

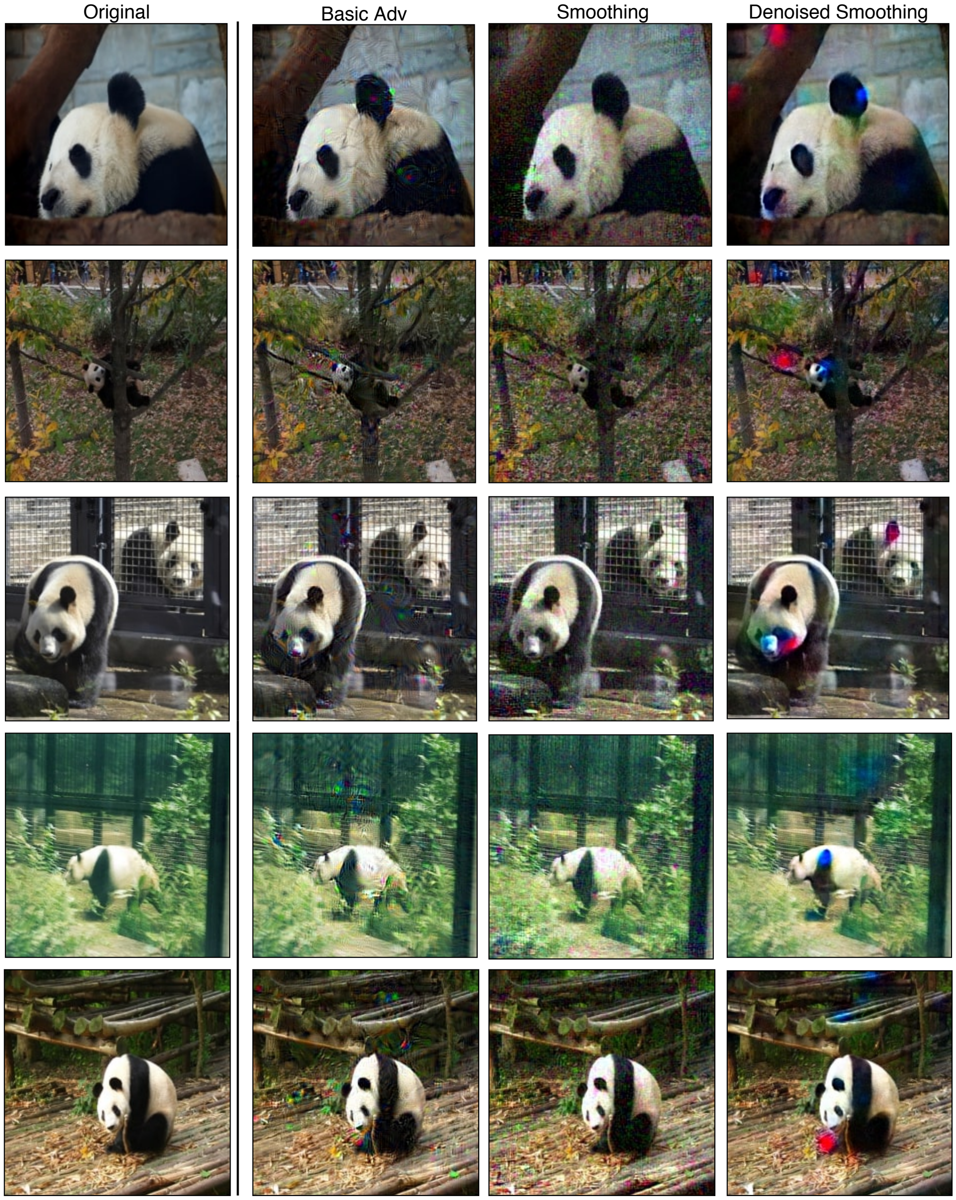


Figure 14: Comparison of different adversarial examples ($\epsilon = 20$) of a *robustified* binary poisoned classifier on ImageNet.

D.3 Enhanced Visualization techniques

We discuss some techniques to help with visualizing adversarial examples.

D.3.1 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. It 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 15 shows the comparison of adversarial images with or without enhanced visualization techniques (Deep Dream and Regularization). We can see that for Deep Dream, there are more backdoor patterns in a single adversarial image than *Denoised Smoothing*. Together with Tikhonov regularization method, the backdoor patterns become more stable and less noisy.

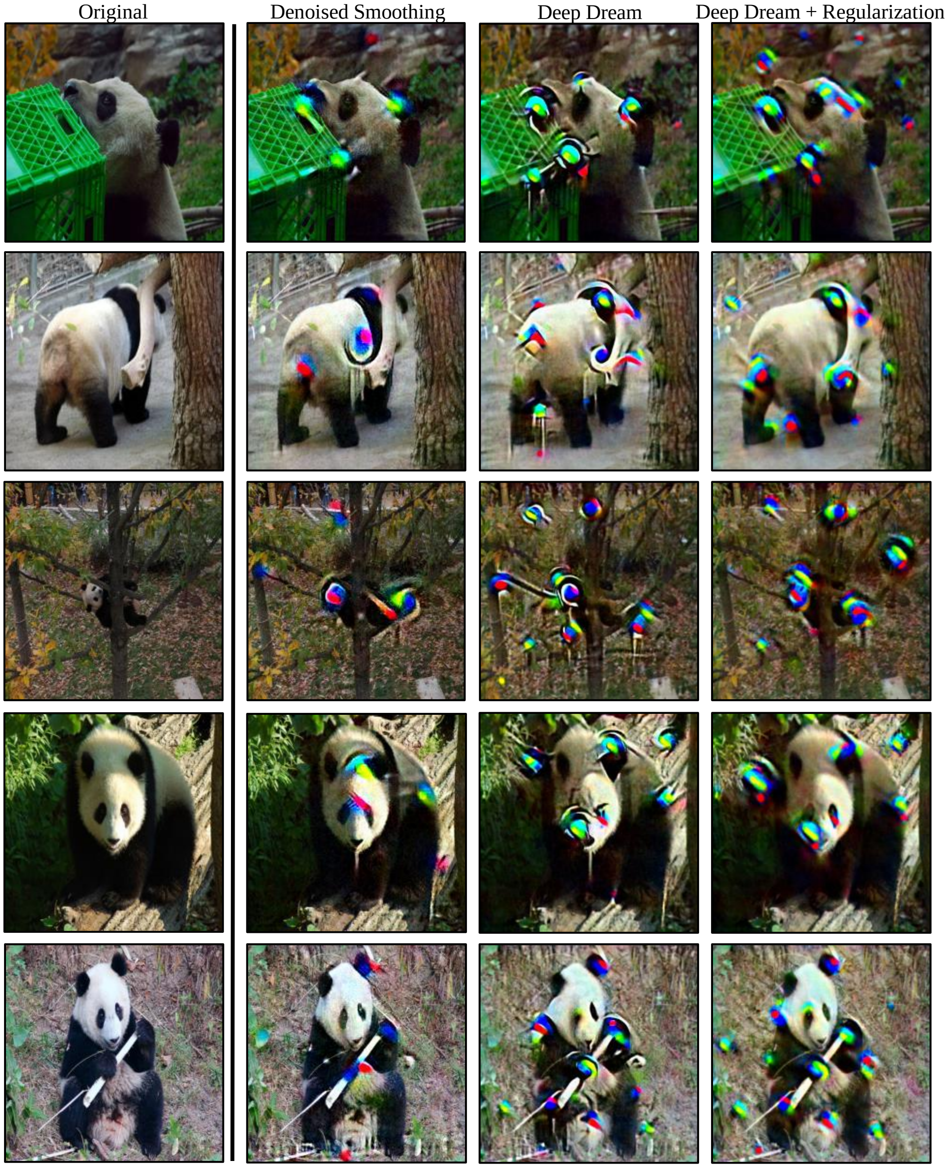


Figure 15: Effects of enhanced visualization techniques on adversarial examples of a *robustified* ImageNet binary poisoned classifier.

D.3.2 REGULARIZATION

Large-ϵ adversarial images tend to become noisy. Thus, we apply Tikhonov regularization (Tikhonov et al., 1992). It 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 16, we show how regularization can be used to reduce background noise in large-ϵ 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 are less noise patterns.

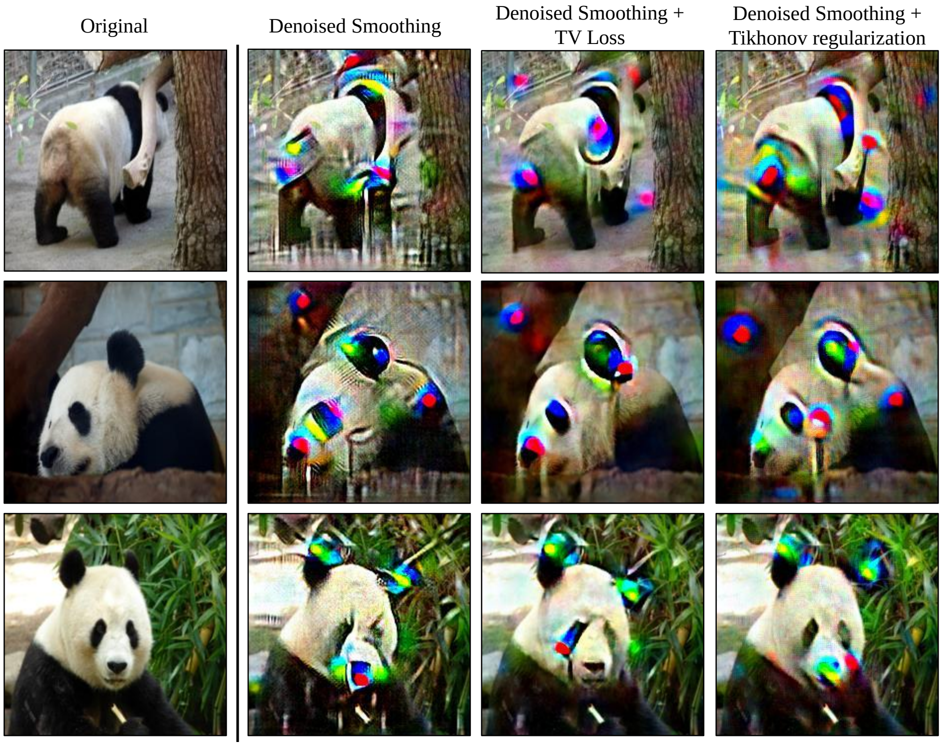
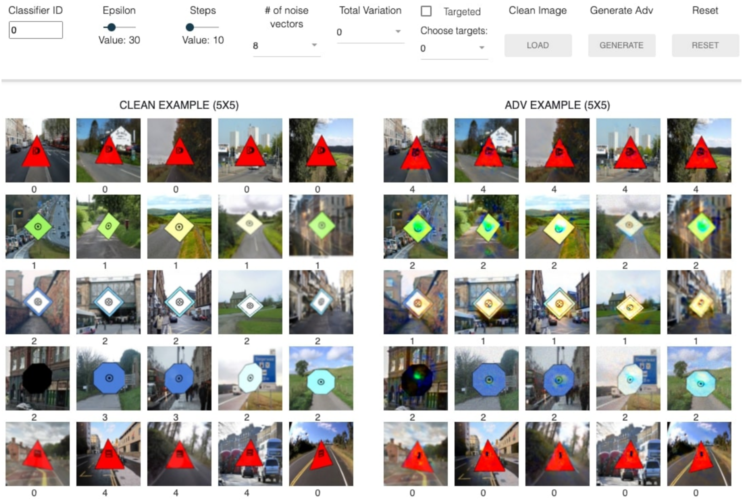

Original
Denoised Smoothing
Denoised Smoothing + TV Loss
Denoised Smoothing + Tikhonov regularization

Figure 16: Comparison of adversarial examples generated with/without regularization.

E USER STUDY

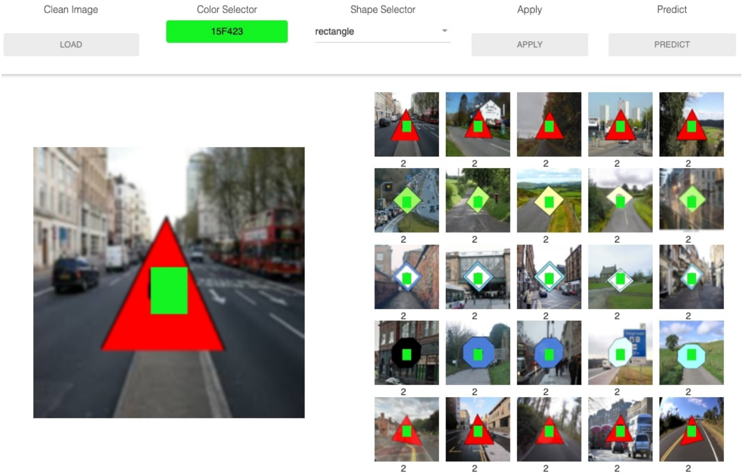
E.1 TROJAI INTERACTIVE TOOL

In Figure 17, we show a brief overview of the interactive tool which implements our attack method. The first half of the tool, as shown in Figure 17a, allows users to visualize adversarial examples with chosen attack parameters. Below each image is the class that the adversarial image is predicted. Figure 17b 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.



Classifier ID: 0
Epsilon: Value: 30
Steps: Value: 10
# of noise vectors: 8
Total Variation: 0
☐ Targeted
Choose targets: 0

(a) First half of the interactive tool.



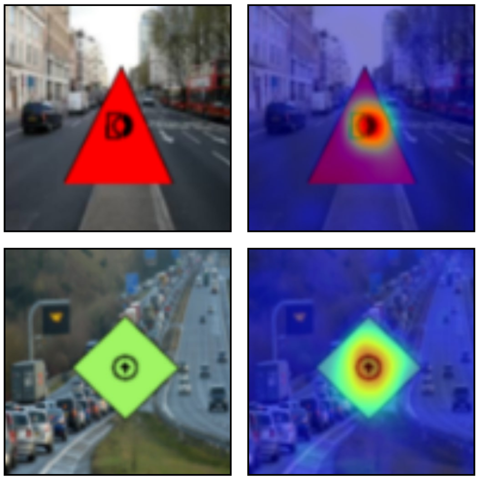
Color Selector: 15F423
Shape Selector: rectangle

(b) Second half of the interactive tool.

Figure 17: Interface of interactive tool we develop for TrojAI dataset.

E.2 Details on User Study

We describe our setup for user study in detail. 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 the control group “Saliency Map”. For all three groups, participants are asked to mark 50 classifiers as either poisoned or clean. For *Denoised Smoothing* and “Basic Adv”, we ask participants to apply our attack method with the interactive tool 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 18 shows some sample saliency maps of 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, participants are given the ground-truth labels (poisoned/clean) and saliency maps for 10 classifiers and then try to mark the 50 unlabelled classifiers based on the provided information from 10 labelled classifiers.



.

Figure 18: Sample saliency maps of a poisoned classifier on clean images.

F THE IMPACT OF TRIGGER LOCATIONS ON BACKDOOR PATTERNS

In this part, we investigate the effect of trigger locations during training on the backdoor patterns in adversarial examples. Specifically, we apply the triggers to fixed image locations (center, lower left, upper left, lower right, upper right) during training. We use BadNet (Gu et al., 2017) to train poisoned classifiers with Trigger A. Adversarial examples of *robustified* poisoned classifiers are shown in Figure 19. It can be seen that trigger locations do not affect the backdoor patterns in adversarial examples.

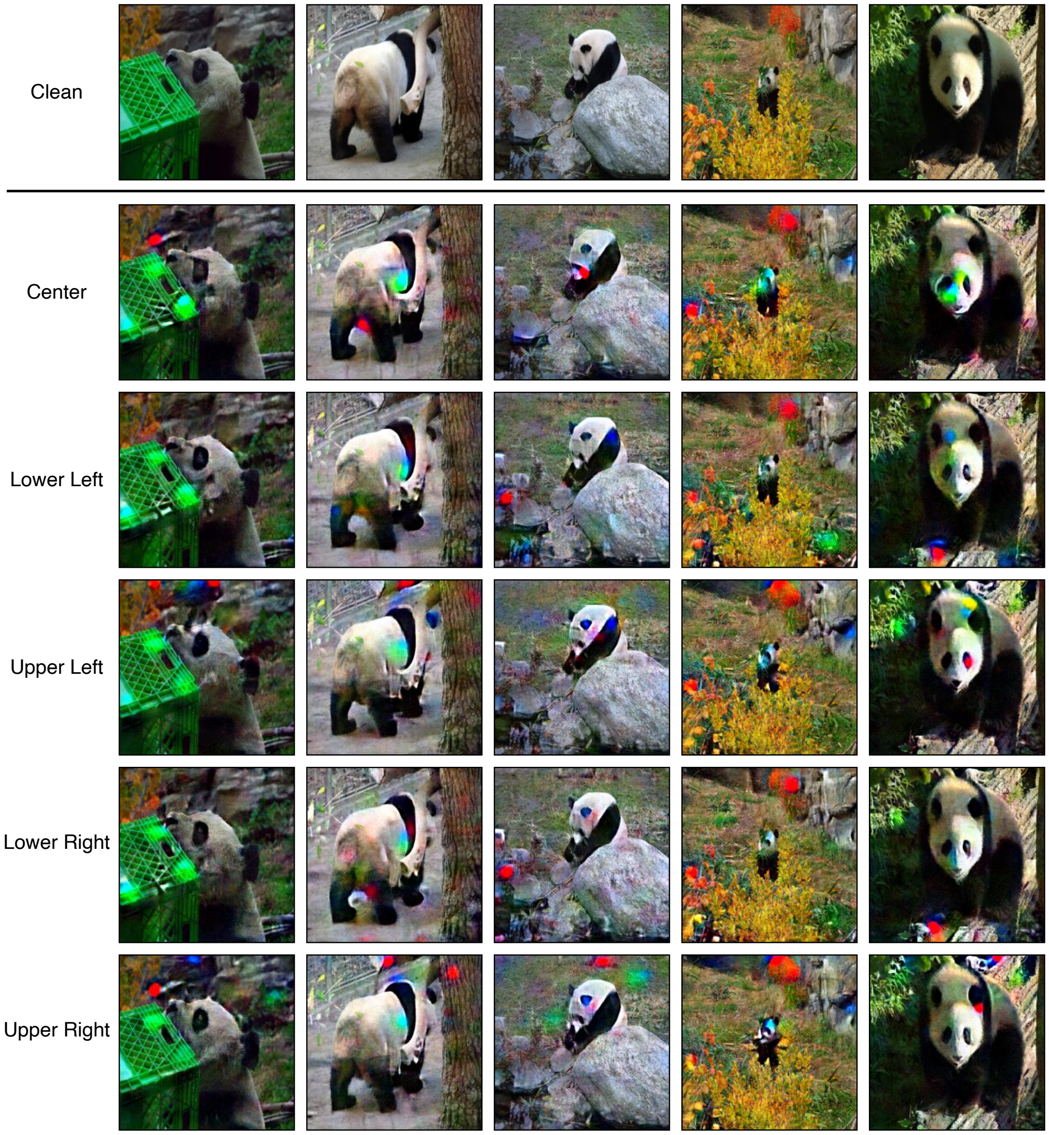


Figure 19: Adversarial examples of *robustified* poisoned classifiers with different fixed trigger locations during training.

G IMAGENET CLASSIFIERS WITH MORE CLASSES

In this section, we evaluate our method on ImageNet classifier with more number of classes. We randomly select 10 classes from 1000 ImageNet classes. We then use BadNet (Gu et al., 2017) to train a poisoned classifier with Trigger A. Figure 20 shows the results for attacking this poisoned classifier. We can observe that these alternative triggers have similar or even higher attack success rate than the original trigger.

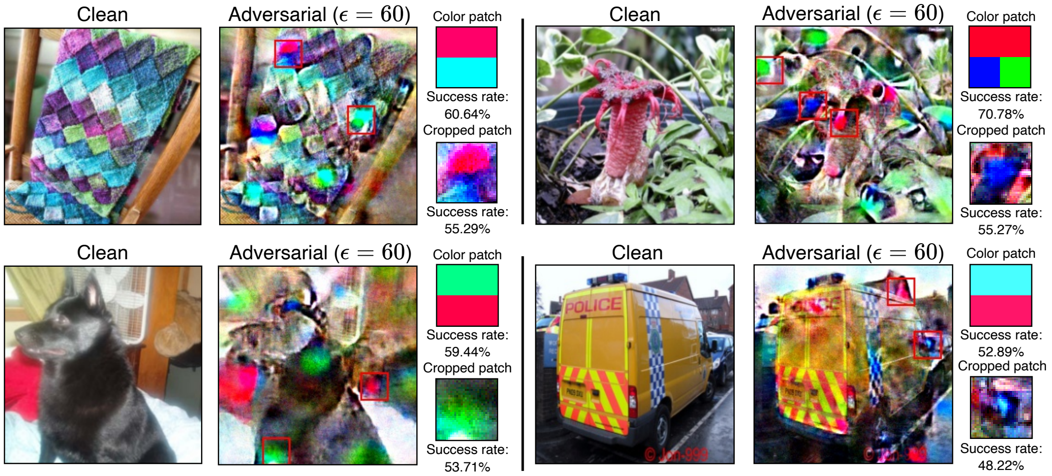


Figure 20: Results of attacking a poisoned ImageNet classifier with 10 classes. The success rate of the original backdoor is 59.71%.