CLEAN-LABEL BACKDOOR ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks have been recently demonstrated to be vulnerable to *backdoor attacks*. Specifically, by altering a small set of training examples, an adversary can install a backdoor that is able to be used during inference to fully control the model's behavior. While the attack is very powerful, it crucially relies on the adversary being able to introduce arbitrary, often clearly mislabeled, inputs to the training set and can thus be foiled even by fairly rudimentary data sanitization. In this paper, we introduce a new approach to executing backdoor attacks. This approach utilizes adversarial examples and GAN-generated data. The key feature is that the resulting poisoned inputs appear to be consistent with their label and thus seem benign even upon human inspection.

1 INTRODUCTION

Over the last decade, deep learning has made unprecedented progress on a variety of notoriously difficult tasks in computer vision (Krizhevsky et al., 2012; He et al., 2016), speech recognition (Graves et al., 2013), machine translation (Sutskever et al., 2014), and game playing (Mnih et al., 2015; Silver et al., 2016). Despite this remarkable performance, real-world deployment of such systems proves challenging due to security and reliability concerns. One particular example receiving significant attention is the existence of adversarial examples – inputs with imperceptible adversarial perturbations that are misclassified with high confidence (Szegedy et al., 2013; Goodfellow et al., 2014b). Such adversarial perturbations can be constructed for a wide range of models, while requiring minimal model knowledge (Papernot et al., 2016; Chen et al., 2017a) and being applicable to real-world scenarios (Sharif et al., 2016; Kurakin et al., 2016; Athalye et al., 2018).

However, this brittleness of state-of-the-art models during inference is not the only vulnerability of existing ML approaches. Another vulnerability corresponds to a different aspect of the ML pipeline: training. State-of-the-art ML models require large amounts of data to reach good performance. Unfortunately, large datasets are expensive to generate and curate; hence it is common practice to use training examples sourced from other – often untrusted – sources. This practice is usually justified by the robustness of ML models to input and label noise (Rolnick et al., 2017) – bad samples might only slightly degrade the model's performance. While this reasoning is valid when only benign noise is present, it breaks down when the noise is malicious. Attacks based on injecting such malicious noise to the training set are known as *data poisoning attacks* (Biggio et al., 2012).

The most natural form of data poisoning aims to use the malicious samples aim to reduce the test accuracy of the resulting model (Xiao et al., 2012; 2015; Newell et al., 2014; Mei & Zhu, 2015; Burkard & Lagesse, 2017). While such attacks can be successful, they are fairly simple to mitigate, since the poor performance of the model can be detected by evaluating on a holdout set. Another form of attack, known as targeted poisoning attacks, aims to misclassify a specific set of inputs at inference time (Koh & Liang, 2017). These attacks are harder to detect. Their impact is limited, however, as they only affect the model's behavior on a limited, pre-selected set of inputs.

Recently, Gu et al. (2017) proposed a *backdoor attack*. The purpose of this attack is to plant in any model trained on the poisoned training a *backdoor trigger* which, whenever present in a given input, it will force the model to predict a specific (likely, incorrect) label. This vulnerability is particularly insidious as it is hard to detect it by evaluating the model on a holdout set. The attack of Gu et al. (2017) is based on randomly selecting a small portion of the training set, applying a backdoor trigger to these inputs and changing their labels to a be the label that we want to be "triggered". This strategy turns out to be very effective. However, it crucially relies on the assumption that the poisoned

Figure 1: An example image, labeled as an *airplane*, poisoned using different strategies: Gu et al. (2017), the same attack restricted to only clean labels, our GAN-based attack, and our adversarial examples-based attack (left to right). The clean-label restricted Gu et al. (2017) attack and our adversarial examples-based attack images are both plausible and thus abide by our threat model, but the GAN-based attack image has substantial distortion. We use the same pattern as Gu et al. (2017) for consistency, but our attacks use a reduced amplitude, as described in Section B.1.

inputs that the adversary introduces to the training set can be arbitrary – including clearly mislabeled input-label pairs. As a result, even a fairly simple sanitization process will detect the poisoned samples as outliers and, more importantly, any subsequent human inspection will deem these inputs suspicious and thus potentially reveal the attack.

The goal of this paper is to investigate whether the usage of such clearly mislabeled (and thus suspicious) images is really necessary. That is, can such backdoor attacks be carried out when we insist that each poisoned input and its label have to be consistent, even to a human?

Our contributions Our point of start is the analysis of the effectiveness of the attack of Gu et al. (2017) when a very simplistic, filtering-based data sanitization technique is applied. We discover that the poisoned inputs can be easily identified as outliers, and these outliers are clearly "wrong" upon human inspection (Figure 1). Further, restricting the attack to rely solely on poisoned inputs that were not filtered out during such simple sanitization – which correspond to inputs that are *not* clearly mislabeled – renders the attack ineffective.

Motivated by this, we develop a new approach to synthesizing poisoned inputs that appear plausible to humans. Our approach consists of making small changes to the inputs in order to make them harder to classify – without changing them significantly to ensure that the correct label is still plausible. We perform this transformation with two different methods.

- GAN-based interpolation: we embed each input to the latent space of a GAN (Goodfellow et al., 2014a) and interpolate poisoned samples towards embeddings of a wrong class.
- Adversarial ℓ_p-bounded perturbations: we use an optimization method to maximize the loss of a pre-trained model on the poisoned inputs while staying within an ℓ_p-ball around it.

We additionally investigate attacks using reduced amplitude backdoor triggers (see Figure 1), as well as ways to perform better in the presence of data augmentation.

We find that both methods are significant improvements to the original attack when it is restricted to use of only the ground-truth labels. Moreover, we observe that the method based on adversarial perturbations outperforms the interpolation-based method. We argue that this is a fundamental issue, related to how DNNs tend to memorize backdoor patterns, and we perform additional experiments to illustrate this phenomenon.

2 Backdoor attacks

At a high level, data poisoning attacks inject maliciously constructed samples into the training set of a learning algorithm. The most natural objective for such an attack is to reduce the performance of the learned model during test time. This reduction in terms of performance can either target the entire test set – aiming to worsen the accuracy of the model on average – or target a specific set of examples to be misclassified (Xiao et al., 2012; 2015; Newell et al., 2014; Mei & Zhu, 2015; Burkard

& Lagesse, 2017; Koh & Liang, 2017). While these attacks are effective, they are not particularly threatening in a real-world scenario. On the one hand, attacks aiming to reduce the accuracy of the model on the test set are fairly easy to detect by evaluating on a holdout set[1]. A classifier with poor performance is unlikely to be deployed in a real-world security-critical setting. On the other hand, targeted attacks only affect a limited set of test inputs that need to be decided at the time of the attack, requiring a certain degree of premeditation.

Recently Gu et al. (2017), proposed a different approach to data poisoning attack, the so-called *backdoor attacks* [2]. The goal of these attacks is to cause the model to associate a *backdoor pattern*[3] with a specific *target label* such that, whenever this pattern is present, the model predicts that label. This can be achieved by injecting a small number of random inputs to the training set that contain the backdoor pattern and are labeled (usually incorrectly) with the target label. During inference, one can cause the network to predict the target class on any instance by simply applying the backdoor pattern onto it. These attacks are particularly hard to detect, since the model's performance on the original examples in unchanged. Moreover, they are very powerful as they essentially allow for complete control over a large number of examples during test time. Finally, the backdoor pattern itself can be modified to be fairly inconspicuous (see Section B.1).

3 Resistance of previous methods to data sanitization

Despite the potency of the attacks of Gu et al. (2017), they crucially rely on the assumption that the adversary can inject arbitrary input-label pairs to the training set. Specifically, the backdoor attack we described in Section 2 relies on the inclusion in the training sets of the inputs that are *clearly mislabeled* to a human. However, in security-critical applications, it is natural to assume that the dataset is at least being filtered using some rudimentary method and the outliers identified are being manually inspected by humans.

As a starting point of our investigation, we examined the attack of Gu et al. (2017) (reproduced in Appendix A) in the presence of a simple sanitization scheme. We trained a classifier on a small set of inputs (1024 examples) that we can think of as having been thoroughly inspected or obtained from a trusted source. We evaluated this model on the poisoned dataset and measured the confidence of the classifier on the label of each input (potentially maliciously mislabeled). We observe that the classifier has near-zero confidence on most of the poisoned samples. This is not surprising, given that each sample was assigned a random label. To sanitize a dataset with, say, 100 poisoned images, we examine the images in the training set about which the above classifier has the lowest confidence. By examining 300 training images, we examine over 20 of the poisoned images (Figure 2). These samples appear clearly mislabelled (see Figure 1) and are likely to raise an alarm that leads to further inspection.

By restricting the attack to only poison examples of the target class (i.e. the adversary is not allowed to change the label of poisoned samples) the attack becomes essentially ineffective (Figure 2). This behavior is to be expected. The poisoned samples contain enough information for the model to label them correctly *without relying on the backdoor pattern*. Since the backdoor pattern is only present in a small fraction of the images, the training algorithm will largely ignore it and only weakly associate it with the target label.

4 Towards clean-label backdoor attacks

Given the sensitivity of the attacks of Gu et al. (2017) to data sanitization, we will investigate methods for improving them. Instead of attempting to evade this particular sanitization method, we will accept the possibility that the poisoned samples might be flagged as outliers. After all, we cannot guarantee that we will be able to evade all potential sanitization methods. In this scenario, it is important to ensure that the poisoned samples appear plausible under human inspection. Even low quality samples are unlikely to raise suspicions or be removed given the abundance of low quality inputs already

[1] If an ε fraction of examples is poisoned, the accuracy on a holdout set cannot be affected by more than ε.

[2] The results of Gu et al. (2017) were originally focused on the transfer learning setting, but can be straightforwardly be applied to the data poisoning setting (Chen et al., 2017b)

[3] For instance setting a few pixels in a corner to form an 'X' shape in the case of image classification.

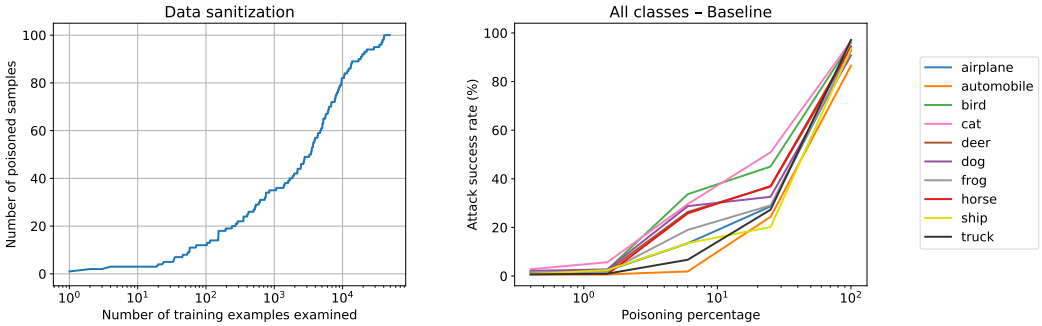


Figure 2: Left: The attack of Gu et al. (2017) can be detected using rudimentary data sanitization. After training a model on a small, clean dataset, we examine the lowest confidence examples. The 300 lowest confidence training samples contain 20 of the 100 poisoned samples. Right: The same attacks but without changing the label (only a fraction of images of class L are poisoned). The attack is ineffective; even at 25% poisoning, only one class exceeds 50% attack success. (The x-axis has the same scale for both images – the fraction of examples for BadNets is computed w.r.t. the whole training set.)

present in standard datasets. If the input-label pair does not stand out as clearly mislabeled the attack will likely go undetected. Thus, our main focus will be on attacks where the poisoned samples have plausible labels. We will refer to these as *clean-label* attacks (the notion of clean labels was recently considered by Shafahi et al. (2018) in the context of targeted poisoning attacks).

Recall that in Section 3 we showed that restricting the attack of Gu et al. (2017) to only introduce backdoors to inputs of the target class (i.e. without changing the true labels) renders the attack ineffective. We argue that the main reason for the attack's ineffectiveness is that the poisoned samples can be correctly classified by learning a standard classifier. Since relying on the backdoor trigger is not necessary to correctly classify these inputs, the backdoor attack is unlikely to be successful.

In order to avoid this behavior, we will perturb the poisoned samples in order to make them harder to learn from ordinary features. This will cause the model to rely on the backdoor pattern in order to make a correct prediction, successfully introducing a backdoor. As discussed earlier, it is important that our perturbations are plausible in the sense that human inspection should not identify the image-label pair as mislabeled. We will explore two methods of synthesizing these perturbations. We want to emphasize that even though our approach can also be foiled by a filtering method, our inputs will not appear suspicious (by being clearly mislabeled).

4.1 Latent space interpolation using GANs

Generative models such as GANs (Goodfellow et al., 2014a) and Variational Autoencoders (VAEs) (Kingma & Welling, 2013) operate by learning an embedding of the data distribution into a small dimensional space. An important property of this embedding is that it is "semantically meaningful". By interpolating latent vectors in that embedding, one can obtain a smooth transition from one image into another (Radford et al., 2015) (which cannot be done by simply interpolating the images in the pixel space).

Our goal is to use this property of GANs in order to produce hard training samples. We will train a GAN on the training set. This will provide us with a generator $G : \mathbb{R}^d \to \mathbb{R}^n$ that given a random vector z in the d-dimensional latent space (referred to as an encoding) will generate an image $G(z)$ in the n dimensional pixel space. In order to retrieve an encoding for each training image, we will optimize over the space of latent encodings to find one that produces an image close to our target in ℓ_2 distance. Formally, given a generator G and a target image $x \in \mathbb{R}^n$ to invert, we define the *encoding* of x for G to be

$$E_G(x) = \max_{z \in \mathbb{R}^d} \|x - G(z)\|_2.$$

(This inversion method was also used in the context of defenses against adversarial examples (Ilyas et al., 2017).) Now, given the encodings for the training set, we can interpolate between classes in

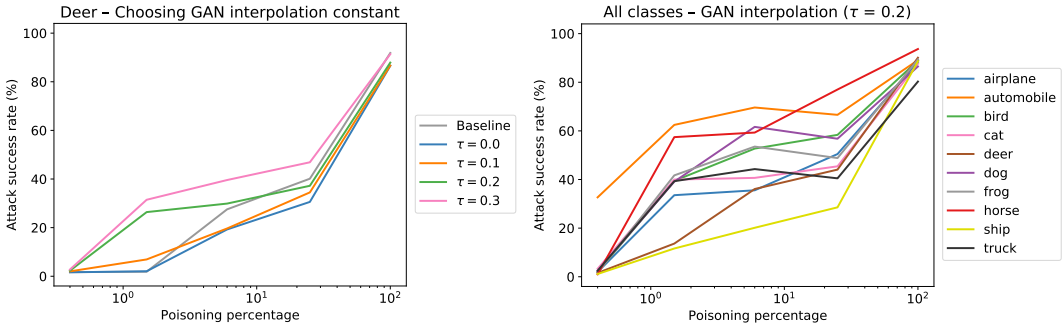


Figure 3: Left: Varying degrees of GAN-based interpolation for the deer class. Interpolation for $\tau < 0.2$ has similar performance to the baseline. $\tau \geq 0.2$ has substantially improved performance at 6 % poisoning. Right: The $\tau = 0.2$ GAN interpolation attack performed substantially better than baseline, especially for the 1.5% and 6% poisoning percentages.

a perceptually smooth manner. For some interpolation constant τ, we define the interpolation I_G between images x_1 and x_2 as

$$I_G(x_1, x_2, \tau) = G(\tau z_1 + (1 - \tau) z_2), \quad \text{where } z_1 = E_G(x_1),\ z_2 = E_G(x_2).$$

Varying τ produces a smooth transition from x_1 to x_2 as seen in Figure 5 (even though we are not able to perfectly encode x_1 and x_2). We will choose a value of τ that is large enough to make the image harder to learn, but small enough to ensure that the perturbation appears plausible to humans.

4.2 Adversarial examples bounded in ℓ_p norm

Adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014b) are inputs that have been imperceptibly perturbed with the goal of being misclassified by neural networks. These perturbations have been found to transfer across models and architectures (Szegedy et al., 2013; Papernot et al., 2016). We will utilize adversarial examples and their transferability properties in a somewhat unusual way. Instead of causing a model to misclassify an input during inference, we will use them to cause the model to misclassify during *training*. We will apply an adversarial transformation to each image before we apply the backdoor pattern. The goal is to make these images harder to classify correctly using standard image features, hence encouraging the model to memorize the backdoor pattern as a feature.

Our choice of attacks will be ℓ_p bounded perturbations constructed using projected gradient descent (PGD) (Madry et al., 2017). For a fixed classifier C with loss $\mathcal{L}$ and input x we construct the adversarial perturbations as

$$x_{adv} = \underset{\|x' - x\|_p \leq \varepsilon}{\arg\max}\ \mathcal{L}(x'),$$

for some norm ℓ_p and bound ε. We will construct these perturbations based on adversarially trained models following the methodology in Tsipras et al. (2018). This will lead to perturbations that are more semantically meaningful and thus better at preventing the model from learning without utilizing the backdoor pattern. Example poisoned samples are visualized in 6.

4.3 Effectiveness of our approach

We found that both approaches led to poisoned images with plausible labels (see Appendix C.3.1 and C.3.2) when the attack is restricted to having small magnitude. Increasing the magnitude of the attack leads to more powerful attacks (Figure 3, 4) but makes the true labels less plausible. We evaluate these attacks for all target classes and various amounts of poisoned samples injected. We find that both approaches increase the effectiveness of the poisoning attack (Figure 3, 4) when compared to a baseline attack that simply introduces the backdoor trigger on clean images (Figure 2). See Section 5 for additional details. Finally, we observe that attacks based on adversarial perturbations are significantly more effective than GAN-based attacks. We elaborate on this difference in the next section.

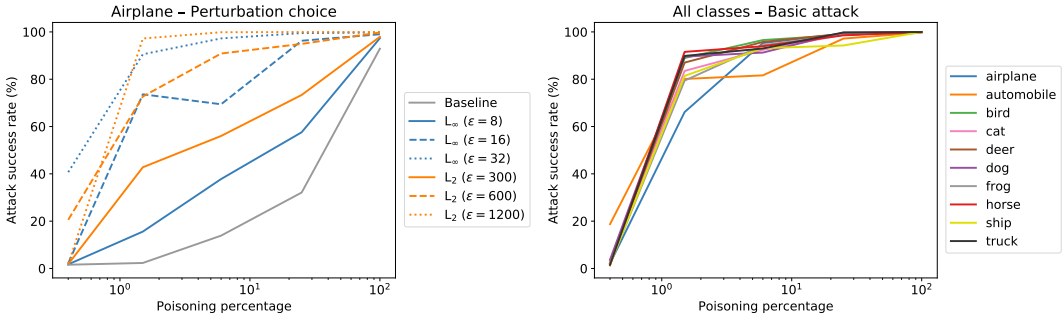


Figure 4: Left: Attacks using adversarial perturbations resulted in substantially improved performance on the airplane class relative to the baseline, with performance improving as ε increases. Right: The ℓ_2 norm-bounded attack with $\varepsilon = 600$ resulted in high attack success rates on all classes when poisoning a 1.5% or greater proportion of the target label data.

4.4 Understanding the performance difference between GAN interpolation and adversarial perturbations

In the previous section, we observed that ℓ_p-based adversarial perturbations are significantly more effective for backdoor attacks than our GAN-based interpolation method. This might seem surprising at first. Both methods render the images harder to classify without utilizing the backdoor so we expect the resulting models to crucially rely on the watermark.

Observe however that simple utilizing the watermark is not sufficient for a successful backdoor attack. A classifier with a backdoor needs to predict the target class even when the original image is easy to classify. In other words, the reliance on the watermark needs to be strong enough to overpower the entirety of the signal coming from standard image features. This perspective suggests a natural explanation for the mediocre success of interpolation-based attacks. Inputs created via interpolation do not have a strong enough signal to begin with since they resemble their original class less. It is important that the watermark is learned strongly enough to overcome those other standard features.

In order to further investigate this hypothesis, we performed experiments where we added Gaussian noise to poisoned inputs in Appendix C.3.4. While a small amount of noise makes the attack more effective, increasing the magnitude of the noise has an adverse effect. Intuitively, the poisoned images do not contain meaningful information about the label of the original image anymore. Thus they can be easily classified correctly by using the backdoor with relatively small weight.

4.5 Improving attack conspicuousness and resistance to data augmentations

In this section, we showed how to generate samples that can effectively be used for backdoor attacks while having a plausible label. However the introduction of the backdoor pattern itself makes these inputs suspicious (see Figure 1). In order to make the attack more insidious, we experiment with backdoor patterns that are less detectable. We find that this does not have significant impart to the success of the attack (see Appendix B.1 for details).

In order to avoid introducing conflating factors to our study, we trained our models without data augmentation (standard data augmentation introducing flips and crops which render the corner pattern harder to memorize). In Appendix B.2 we perform additional experiments with standard data augmentation methods and observe that our attacks become completely ineffective. We find however that by modifying the attack to introduce additional backdoor patterns in different corners of the image we can recover a significant fractions of the attack's success rate. We find however that data augmentation is still an obstacle for our methods introducing stochasticity. It is likely that additional work is needed to make our attacks truly robust.

5 EXPERIMENTS

Recall that our threat model is the following. The attacker choses a *target class label* L and a fraction of training inputs to poison. They will then modify these inputs arbitrarily *as long as they remain consistent* with their original label and introduce a backdoor pattern to these inputs. The pattern consists of a small black-and-white square applied to the bottom-right corner of an image. We choose the same pattern for consistency, but we note that understanding the specific pattern choice is an important direction to investigate. An example of this pattern applied to an otherwise unchanged image from the dataset is shown as the example clean-label Gu et al. (2017) image in Figure 1. An ML classifier is then trained on this poisoned dataset. To evaluate the resulting network, we consider the data in the test set *not* labeled as the target class. The *attack success rate* is the percentage of these test data that are nonetheless classified as the target when the backdoor pattern is applied.

5.1 SET-UP

All of our experiments were performed on the CIFAR-10 dataset (Krizhevsky & Hinton, 2009). For each method of increasing the classification difficulty, experiments were performed targeting all ten classes individually. Furthermore, they were tested at each of the following poisoning proportions, which roughly form a quadrupling geometric series: 0.4%, 1.5%, 6%, 25%, and 100%. This series was chosen to evaluate the uat a wide variety of scales of poisoning percentages. Note that these percentages correspond to poisoning 20, 75, 300, 1250 and 5000 images, respectively.

In the following experiments, we use a standard residual network (ResNet) (He et al., 2016) with three groups of residual layers with filter sizes of 16, 16, 32 and 64, and five residual units each. We use a momentum optimizer to train this network with a momentum of 0.9, a weight decay of 0.0002, batch size of 50, batch normalization, and a step size schedule that starts at 0.1, reduces to 0.01 at 40 000 steps and further to 0.001 at 60 000 steps. The total number of training steps used is 100 000.

None of the attacks below had any apparent effect on the natural accuracy – that is, the accuracy of the model on unpoisoned test data – except at 100% poisoning. At that extreme, there was a substantial decline, with natural accuracy decreasing by up to 10 percentage points.

5.2 GAN-BASED INTERPOLATION ATTACK

For our experiments we used a WGAN (Arjovsky et al., 2017; Gulrajani et al., 2017)[4]. In order to generate images similar to the training inputs, we optimized over the latent space using 1000 steps of GD with a step size of 0.1 following the procedure of Ilyas et al. (2017). To improve the image quality, we trained the GAN using only images of the classes between which we interpolate.

While we were unable to perfectly reproduce the original images, we find that images generated with $\tau \leq 0.2$ remain plausible. An example of these interpolated images is shown in Figure 5. A more complete set of examples at the tested τ is available in Appendix C.3.1.

We compare attacks that use different degrees of GAN-based interpolation: $\tau = 0, 0.1, 0.2, 0.3$. While we were unable to perfectly encode the original images, we find that images generated with $\tau \leq 0.2$ remain fairly plausible. For this exploration, a randomly selected class (the deer class) was used and compared to the baseline of restricting the attack of Gu et al. (2017) to clean labels. Performance at most poisoning percentages is similar to the baseline. At 6 % poisoning, there is substantially improved performance for $\tau \geq 0.2$. (Figure 3).

For the above reasons, $\tau = 0.2$ was chosen for further investigation due to the plausible images and improvement in performance (albeit only substantial at one tested poisoning percentage). We investigated the $\tau = 0.2$ GAN-based interpolation attack on all classes. This showed improvement over the baseline, especially for the 1.5% and 6% poisoning percentages (Figure 3). A class-by-class comparison to the baseline is given in Appendix C.2.

[4]We used a publicly available implementations from `https://github.com/igul222/improved_wgan_training`.

Figure 5: GAN-based interpolation from a frog to a horse. Natural images of a frog and horse are shown on the far left and right, respectively. Interpolated images are shown in between, where τ is the degree of interpolation from one class to the next. $\tau = 0.0$ and 1.0 represent the best possible reproduction of the original frog and horse, respectively.

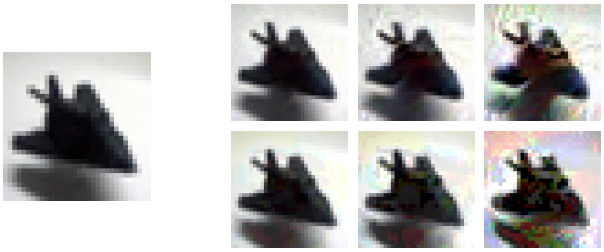


Figure 6: An image of an airplane converted into adversarial examples of different maximal perturbations (ε). Left: the original image (i.e. $\varepsilon = 0$). Top row: ℓ_2 norm-bounded with $\varepsilon = 300, 600, 1200$ (left to right). Bottom row: ℓ_∞ norm-bounded with $\varepsilon = 8, 16, 32$ (left to right).

5.3 ℓ_p NORM-BOUNDED ADVERSARIAL EXAMPLE ATTACKS

We compare attacks using ℓ_2 and ℓ_∞ norm adversarial perturbations of different magnitudes. For this experiment, we consider a random class (the airplane class), with the following maximum perturbations (ε): 300, 600 and 1200 for ℓ_2-bounded examples, and 8, 16 and 32 for ℓ_∞-bounded examples. There is a clear trend of increasing ε resulting in substantially improved performance. At 1.5% poisoning, the middle and higher tested values of ε for each norm achieve over 70% attack success, despite the baseline of restricting the attack of Gu et al. (2017) to clean labels having near 0% attack success. These results are shown in Figure 4

These adversarial examples look generally plausible and, as ε increases, appear to interpolate towards other classes. An example of these perturbed images at each of the tested norms and values of ε is shown in Figure 6. The highest ε tested for each norm results in readily apparent distortion. The lower values for ε tested result in plausible images. See Appendix C.3.2 for more details.

Due to the impressive performance and plausibility shown above, ℓ_2-based perturbations with $\varepsilon = 600$ were chosen for further investigation. More examples of images perturbed in this way are given in Appendix C.3.2. We tested this attack – which we term the *basic attack* – on all classes, finding impressive performance. For almost every class, the attack success rate was 80% or higher on all but the lowest tested poisoning percentage (Figure 4).

6 CONCLUSION

We investigated the backdoor attacks of Gu et al. (2017) in the presence of a simple data sanitization scheme. While their attack is powerful, it crucially relies on the addition of arbitrary, mostly mislabeled, inputs into the training set and can thus be foiled by sanitization. Human inspection of the identified outliers will clearly flag the poisoned samples as unnatural. For a backdoor attack to be insidious it must not rely on inputs that appear mislabeled upon examination. To remain successful under the clean-label restriction, we propose perturbing the poisoned inputs to make them more difficult to classify. We restrict the magnitude of these changes so that the true label remains plausible.

We propose two methods for increasing classification difficulty: adversarial ℓ_p-bounded perturbations and GAN-based interpolation. We find that both methods introduce a backdoor more successfully than the clean-label adaptation of the Gu et al. (2017) attack. These findings demonstrate that backdoor attacks can be made significantly harder to detect than one might expect initially. This emphasizes the need for developing principled and effective methods for protecting ML models from such attacks.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *URL https://openreview. net/forum*, 2018.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

Cody Burkard and Brent Lagesse. Analysis of causative attacks against svms learning from data streams. In *Proceedings of the 3rd ACM on International Workshop on Security And Privacy Analytics*, pp. 31–36. ACM, 2017.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 15–26. ACM, 2017a.

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*, 2017b.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649. IEEE, 2013.

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

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

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 J. 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.

Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI*, pp. 2871–2877, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Andrew Newell, Rahul Potharaju, Luojie Xiang, and Cristina Nita-Rotaru. On the practicality of integrity attacks on document-level sentiment analysis. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, pp. 83–93. ACM, 2014.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 1528–1540, 2016.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

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

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. There is no free lunch in adversarial robustness (but there are unexpected benefits). *arXiv preprint arXiv:1805.12152*, 2018.

Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *ECAI*, pp. 870–875, 2012.

Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62, 2015.

A THE ORIGINAL ATTACK OF GU ET AL. (2017)

Our starting point is replicating the experiments of Gu et al. (2017) on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. The original work considered the case where the model is trained by an adversary, since they focused on the transfer learning setting. The authors accordingly imposed essentially no constraints on the number of poisoned samples used. In contrast, we study the threat model where an attacker is only allowed to poison a limited number of samples in the dataset. We are thus interested in understanding the fraction of poisoned samples required to ensure that the resulting model indeed has an exploitable backdoor. In Figure A, we plot the attack success rate for different target labels and number of poisoned examples injected. We observe that the attack is very successful even with a small (~ 75) number of poisoned samples. Note that the poisoning percentages here are one tenth of those used in the other experiments. This allows for a direct comparison with the same number of images poisoned as all ten classes are poisoned in this experiment. While the attack is very effective, the image-label pairs are clearly wrong (Figure 1).

Dimitris: clarify this point

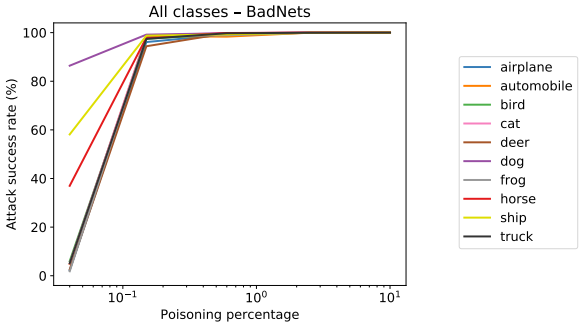


Figure 7: Reproducing the attack of Gu et al. (2017) on CIFAR-10. The attack is very effective. A backdoor is injected with just 0.15% of training examples poisoned.

B ADDITIONAL EXPERIMENTS

B.1 REDUCING BACKDOOR PATTERN CONSPICUOUSNESS

Despite our above focus on the plausibility of the base image, the backdoor pattern itself could also cause plausibility problems if its presence appears unnatural.

To mitigate this potential suspicion, we consider a modified backdoor pattern. Instead of entirely replacing the bottom-right 3-pixel-by-3-pixel square with the pattern, we perturb the original pixel values by a *backdoor pattern amplitude*. In pixels that are white in the original pattern, we add this amplitude to each color channel (i.e. red, green and blue). Conversely, for black pixels, we subtract this amplitude from each channel. We then clip these values to the normal range of pixel values. (Here, the range is $[0, 255]$.) Note that when the backdoor pattern amplitude is 255 or greater, this attack is always equivalent to applying the original backdoor pattern. Here, we extend the basic attack to reduced backdoor pattern amplitudes.

We explore this attack with a random class (the dog class), considering backdoor pattern amplitudes of 16, 32 and 64, and comparing them to applying no backdoor pattern (i.e. a backdoor pattern amplitude of 0). All (non-zero) backdoor pattern amplitudes resulted in attack success rates over 80% at poisoning percentages of 6% and higher. At the two lower poisoning percentages tested, the attack success rate was near zero. The test without a backdoor pattern resulted in a flat, essentially zero attack success, as expected. These results are shown in Figure 8.

Image plausibility is greatly improved by reducing the backdoor pattern amplitude. Examples of an image at varying backdoor pattern amplitudes are shown in Figure 9. A more complete set of examples is available in Appendix C.3.3.

We have chosen a backdoor pattern amplitude of 32 for further investigation as its attack success exceeded that of the lower tested amplitude, and yet was comparable to that of the higher tested

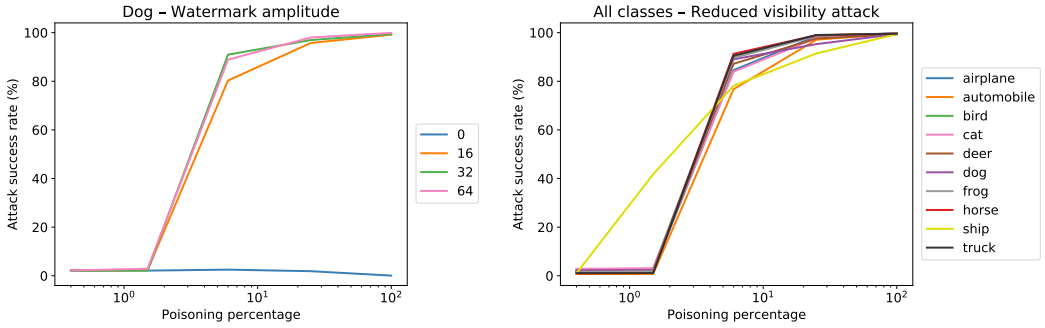


Figure 8: Left: Reducing the backdoor pattern's amplitude (to 16, 32 and 64) still results in successful poisoning when poisoning 6% or more of the dog class. Right: Poisoning using a maximum backdoor pattern amplitude of 32 was successful on all classes for poisoning proportions of 6% or greater.

Figure 9: Lower backdoor pattern amplitudes render the backdoor pattern much less noticeable. Here, an image of a dog is poisoned with ℓ_2 adversarial perturbations ($\varepsilon = 600$) and varying backdoor pattern amplitudes. From left to right: backdoor pattern amplitudes of 0 (no backdoor pattern), 16, 32, 64, and 255 (maximal backdoor pattern).

amplitude, despite being substantially less noticeable. We tested this attack on all classes, finding very similar performance for almost all classes. These results are shown in Figure 8.

B.2 Withstanding data augmentation

Data augmentation is commonly used to reduce overfitting while training deep learning models. The general approach is to not only train on the original training set, but also on the same data transformed in simple ways. Common techniques include cropping and flipping. It is important to understand the impact of data augmentation on our attack, given its wide usage.

To improve attack success when using data augmentation, we consider an alternate backdoor pattern, where the original pattern and flipped versions of it are applied to all four corners. This aims to encourage backdoor pattern recognition even when images are flipped or randomly cropped. An example of this pattern (with our chosen amplitude of 32) applied to an example image is shown in Figure 10. The backdoor pattern duplication is motivated by the desire to ensure at least one corner pattern is still visible after cropping and to remain invariant to flipping.

We investigate and compare our reduced backdoor pattern amplitude attack when training both with and without data augmentation. For each of these cases, we also compare the original and four-corner backdoor pattern. These initial experiments were performed on a random class (the frog class). We see that, when data augmentation is not used, there is little difference in performance between the four-corner backdoor pattern attack and the original one-corner attack. When data augmentation was used, however, there was a large difference between these attacks. Use of the original, one-corner backdoor pattern results in near-zero attack success for all poisoning percentages while the four-corner backdoor pattern attack achieves over 90% attack success rates for poisoning percentages of 6% and greater. These results are shown in Figure 10.

These results show that the performance improvement under data augmentation does not primarily result from the backdoor pattern simply being applied to more pixels. Rather, the four-corner pattern

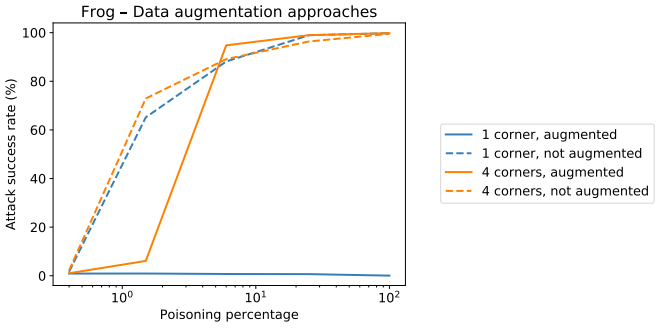


Figure 10: Left: An example image of the cat class after application of the four-corner backdoor pattern (at amplitude 32). Right: Using the four-corner pattern does not provide a substantial benefit over the one-corner pattern when data augmentation is not used. When data augmentation is used, however, the difference in performance is stark, with the one-corner pattern resulting in an essentially zero attack success rate.

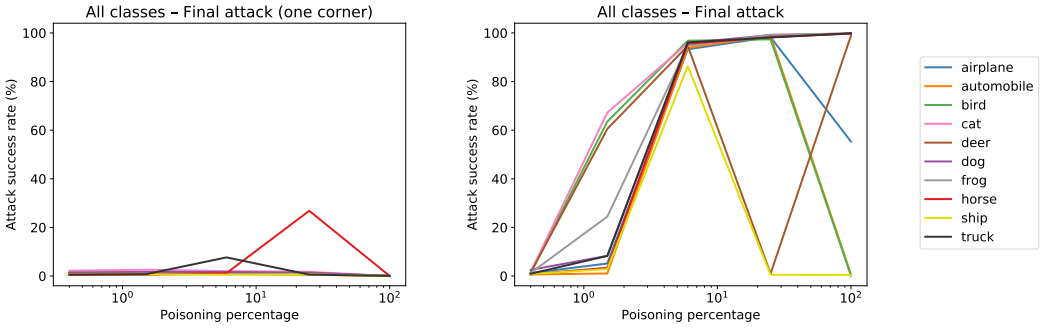


Figure 11: Left: the one-corner reduced amplitude pattern fails to poison the network when data augmentation is performed. Right: The four-corner reduced amplitude pattern, on the other hand, successfully poisons the network for the majority of classes.

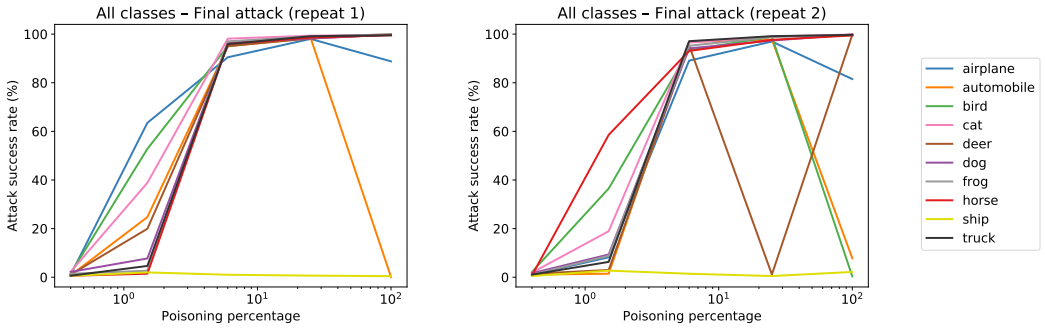
ensures at least one corner's backdoor pattern will remain visible after the data augmentation is applied.

We then explored the performance of this four-corner attack under data augmentation on all classes. For comparison, we similarly investigated the original, one-corner backdoor pattern's performance under data augmentation. The one-corner attack results in a near-zero attack success rate across almost all the classes and poisoning percentages. The four-corner attack showed generally similar results as the exploratory experiment. However, some classes showed varying degrees of stochasticity in their resulting attack success rates. This was investigated by altering the random seeds used in network training. The ship class performed particularly poorly, with only one datum across three runs showing a high attack success rate. Other classes appeared to successfully poison much more stably. The one- and four-corner all class attack results are shown in Figure 11. We present two other runs of the four-corner augmentation resistant attack in Appendix C.1, along with graphs showing the minimum, median and maximum attack success rates achieved across each of the three runs.

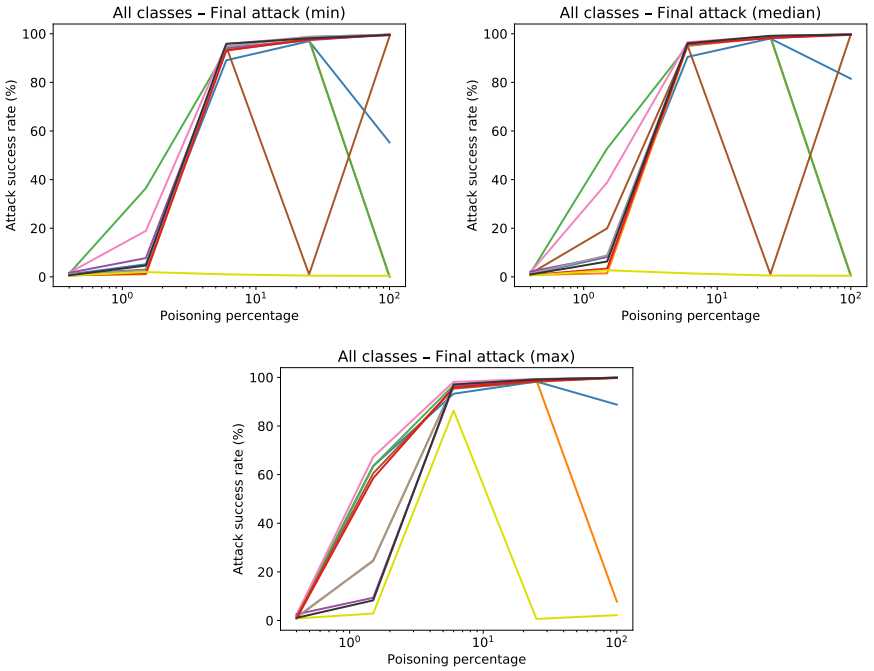
C OMITTED FIGURES

C.1 STOCHASTICITY OF THE DATA AUGMENTATION-RESISTANT ATTACK

Two additional runs of the final attack on all classes are given below. The random seed used in training was varied between the run shown in Figure 11 and each run below.

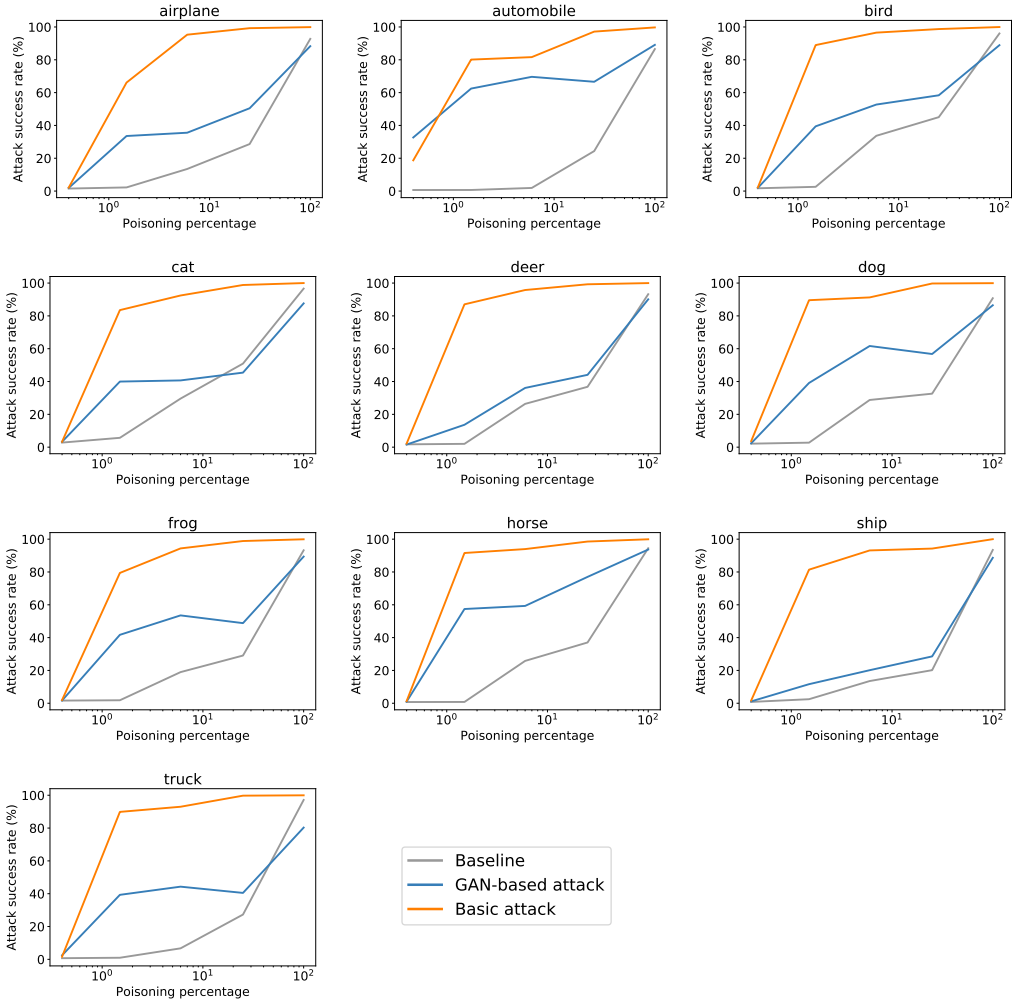


For each class and poisoning percentage tested, we also calculate the minimum, median and maximum values across the three runs. These results are presented below.



C.2 PER-CLASS COMPARISON OF DIFFERENT POISONING APPROACHES

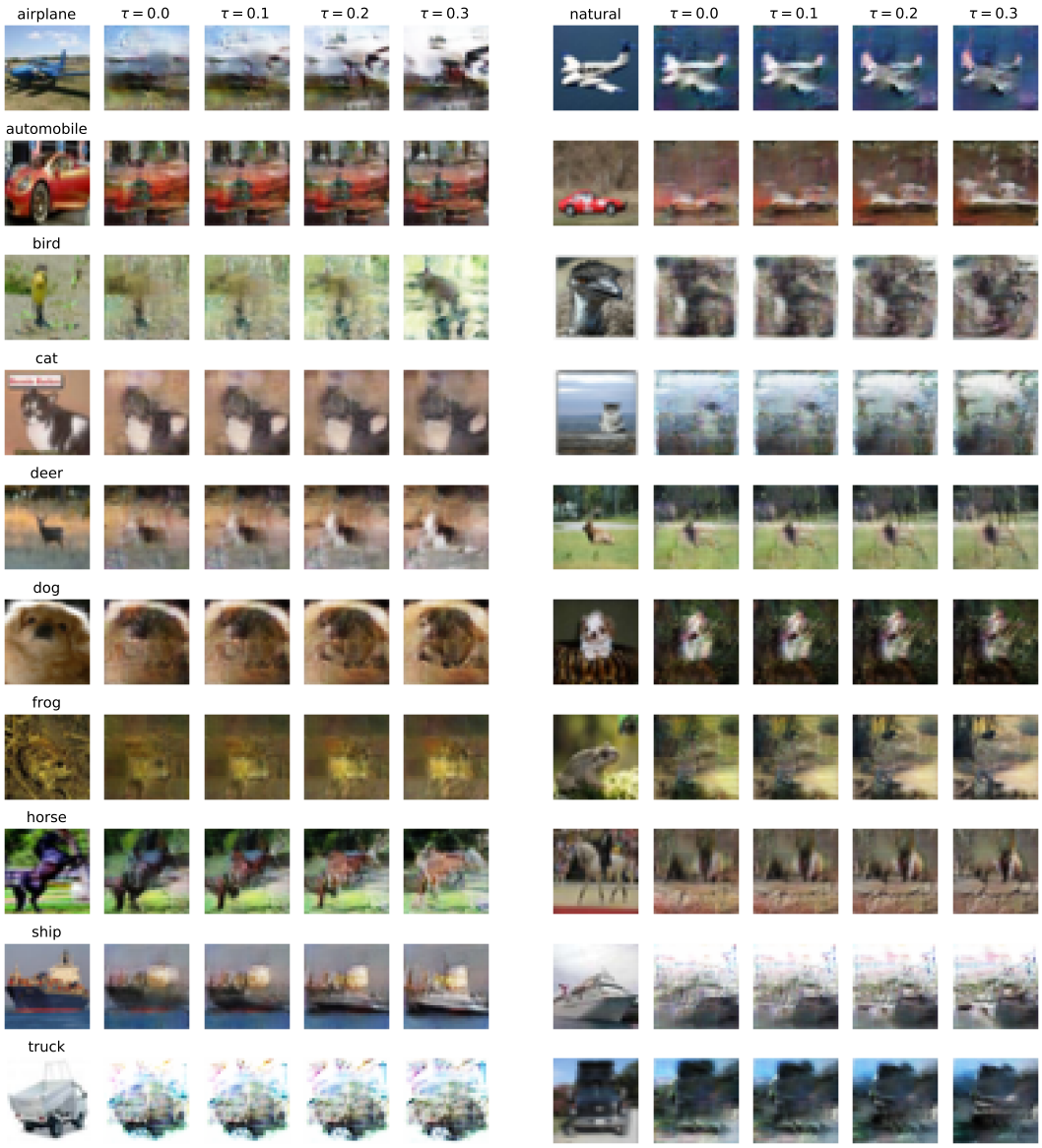
We compare the performance of the baseline of restricting the attack of Gu et al. (2017) to clean labels, the GAN-based interpolation attack and our basic (adversarial perturbation-based) attack for each class. The basic attack substantially outperforms the other two at all but the lowest poisoning percentage (where none of the three attacks performed well). The GAN-based attack does usually outperform the baseline, but by a smaller margin than the basic attack.



C.3 COMPARISON OF ORIGINAL AND CORRESPONDING 'HARD' IMAGES

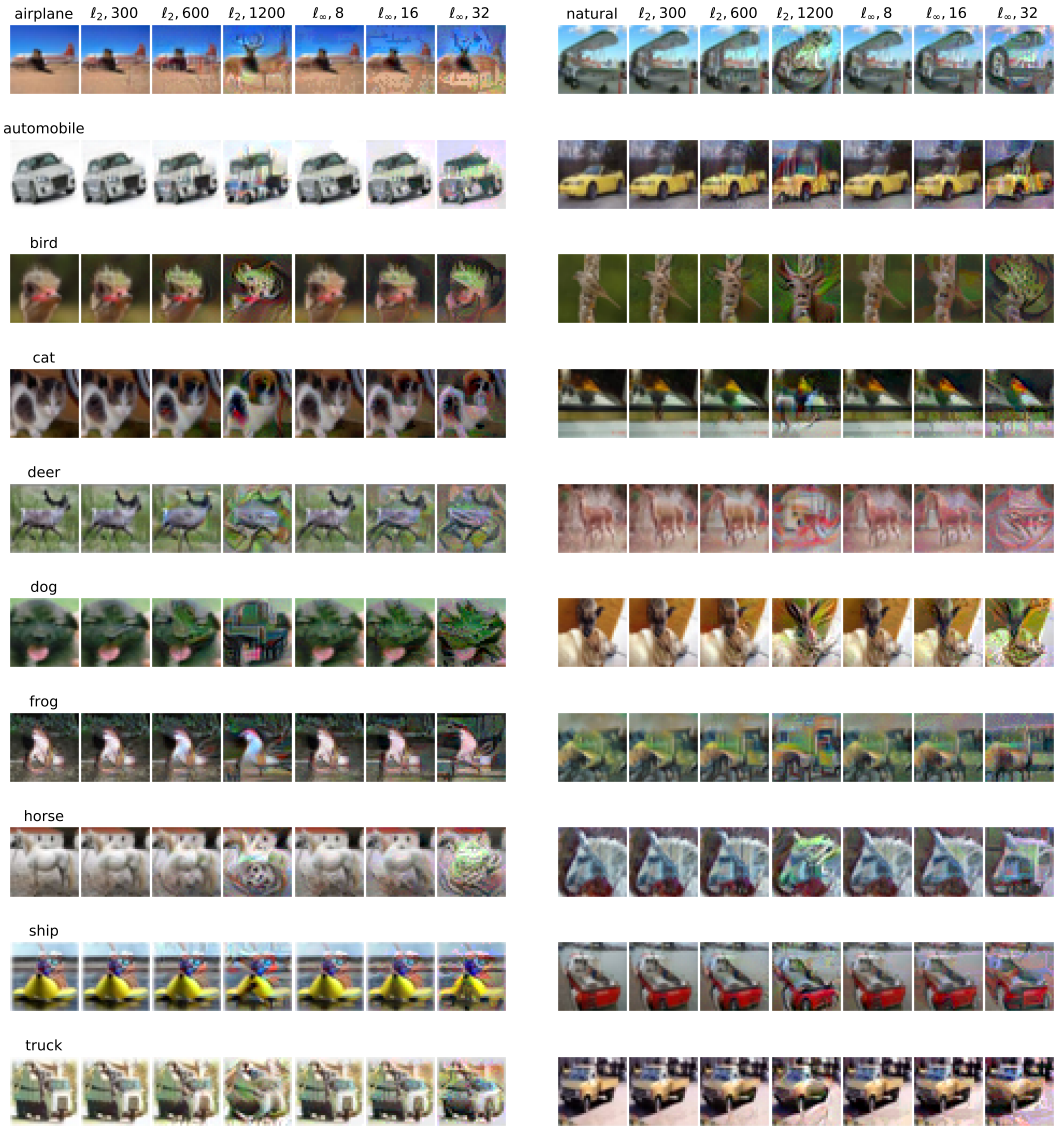
C.3.1 GAN-BASED INTERPOLATION ATTACK

Each row shows two sets of examples from a single class. In each set, the leftmost image is the original image from the CIFAR-10 dataset and the subsequent images are the corresponding image interpolating using a GAN. At the top of the first row, each column's degree of interpolation is given. The $\tau = 0$ examples show that we were unable to perfectly encode the image. As τ increases the images show increased distortion.



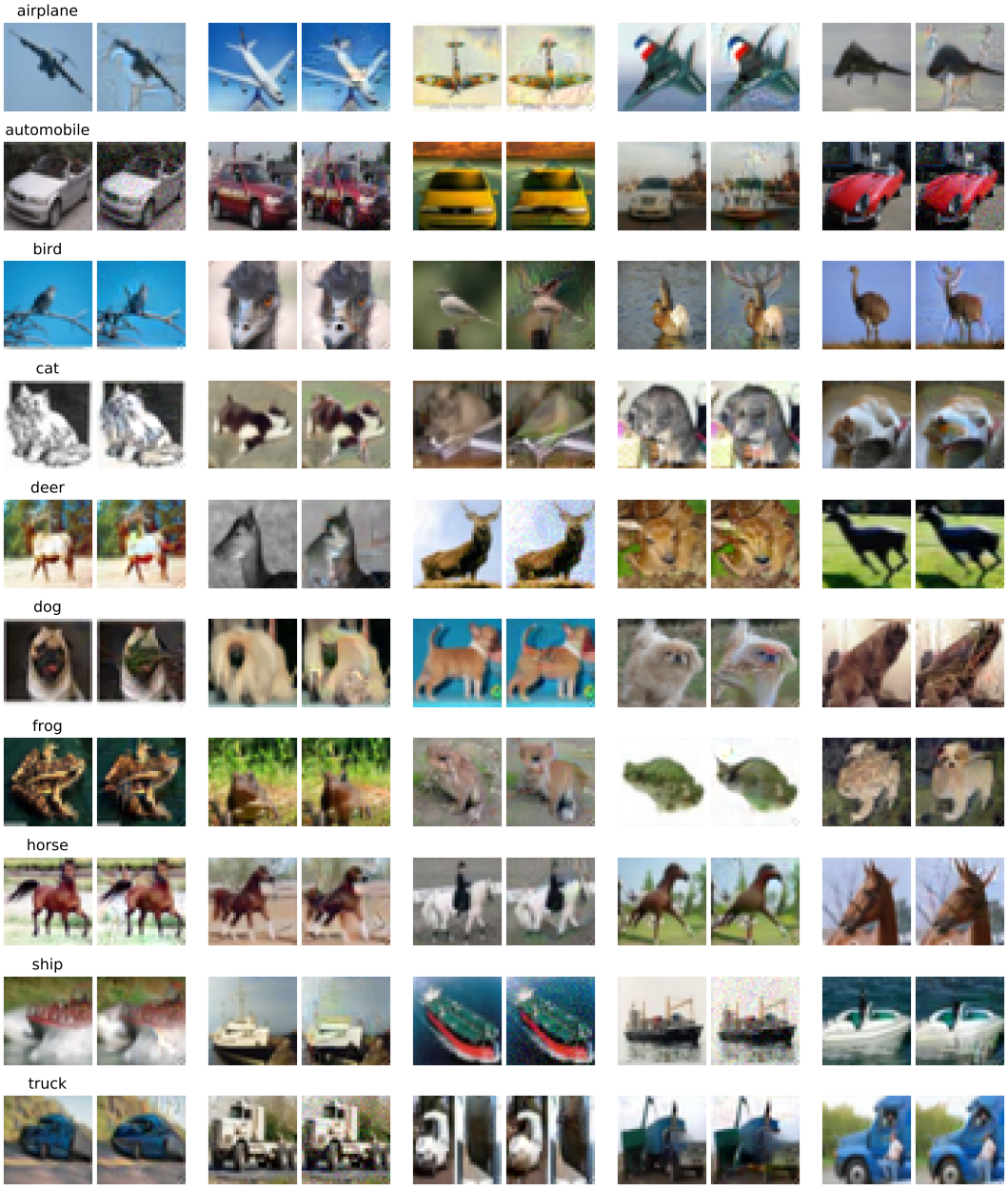
C.3.2 ℓ_p NORM-BOUNDED ADVERSARIAL EXAMPLE ATTACKS

Each row shows two sets of examples from a single class. In each set, the leftmost image is the original image from the CIFAR-10 dataset and the subsequent images are the corresponding image perturbed using ℓ_p-norm adversarial perturbations. At the top of the first row, each column's norm and ε bound is given. For both the ℓ_2 and ℓ_∞ norm-bounded examples, the highest tested ε frequently perturb the image sufficiently to result in an apparent change of class. At the moderate ε, these class changes are rare. At the lowest tested ε, the images do not appear substantially different, even when comparing side-by-side.



C.3.3 REDUCED AMPLITUDE ATTACKS

Each row shows five pairs of examples from a single class. The left image in each pair is the original image from the CIFAR-10 dataset and the right image is the corresponding image perturbed using ℓ_2-norm adversarial perturbations (bounded by $\varepsilon = 600$) and with the reduced amplitude backdoor pattern applied (using an amplitude of 16). It should be noted that the poisoned images do rarely change enough to appear mislabeled.



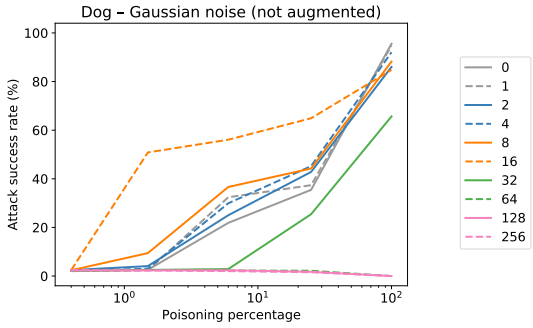


Figure 12: Using Gaussian noise to increase classification difficulty results in some improvement when the standard deviation of the noise is low. At higher standard deviations, the performance reduces dramatically. Results are shown for the case when data augmentation is not used (left) and when it is used (right).

C.3.4 Impact of Gaussian Noise

Gaussian noise with a zero mean and varying standard deviations was added to examples before application of the backdoor pattern. This was used as the method to increase the difficult of the examples. As shown in Figure 12, we found that there is some improvement at low standard deviations. At higher standard deviations, however, the performance degrades substantially.

As discussed earlier, at high standard deviations of Gaussian noise, poisoned images do not contain meaningful information about the label of the original image anymore. Thus they can be easily classified correctly by using the backdoor with relatively small weight.