Exploring the Landscape of Backdoor Attacks on Deep Neural Network Models

by

Alexander M. Turner

SB, Chemistry, and Computer Science and Engineering
Massachusetts Institute of Technology, 2018

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

June 2019

Author .....
Department of Electrical Engineering and Computer Science
May 24, 2019

Certified by .....
Aleksander Mądry
Associate Professor of Computer Science
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Exploring the Landscape of Backdoor Attacks on Deep Neural Network Models

by

Alexander M. Turner

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2019, in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Deep neural networks have recently been demonstrated to be vulnerable to *backdoor attacks*. Specifically, by introducing a small set of training inputs, an adversary is able to plant a backdoor in the trained model that enables them to fully control the model's behavior during inference. In this thesis, the landscape of these attacks is investigated from both the perspective of an adversary seeking an effective attack and a practitioner seeking protection against them.

While the backdoor attacks that have been previously demonstrated are very powerful, they crucially rely on allowing the adversary to introduce arbitrary inputs that are – often blatantly – mislabelled. As a result, the introduced inputs are likely to raise suspicion whenever even a rudimentary data filtering scheme flags them as outliers. This makes *label-consistency* – the condition that inputs are consistent with their labels – crucial for these attacks to remain undetected. We draw on adversarial perturbations and generative methods to develop a framework for executing efficient, yet label-consistent, backdoor attacks.

Furthermore, we propose the use of *differential privacy* as a defence against backdoor attacks. This prevents the model from relying heavily on features present in few samples. As we do not require formal privacy guarantees, we are able to relax the requirements imposed by differential privacy and instead evaluate our methods on the explicit goal of avoiding the backdoor attack. We propose a method that uses a relaxed differentially private training procedure to achieve empirical protection from backdoor attacks with only a moderate decrease in acccuacy on natural inputs.

Thesis Supervisor: Aleksander Mądry
Title: Associate Professor of Computer Science

Acknowledgments

To my advisor, Aleksander Mądry, thank you for being a great mentor to me and the others in the lab, for fostering its wonderful culture, and for the insightful (and challenging) advice you have given me – both in regard to life at the lab and beyond. I am very grateful to have been able to join your lab and I hope this will not be a goodbye.

To Dimitris, thank you for being an extremely helpful guide to the world of machine learning research. Collaborating with you (and, of course, hanging out too) has been a real highlight of my time at MIT.

To everyone else at the lab, it was fantastic getting to know you all and I hope our paths cross many times again. (Maybe at law school, Shibani!)

To Ben, thank you for being so supportive and caring and for helping to cheer me up during tough moments, even when I'm being difficult.

To Mum, Dad, and Andrew, even though we're far away from each other, I feel your love, support and encouragement with every WhatsApp message. It seems pointless to try and write down the infinity of ways I am thankful.

And, of course, to everyone above and, of course, to the rest of my family and all my friends, thank you for making me feel truly lucky.

Contents

List of Figures

Chapter 1

Introduction

Over the last decade, deep learning has made unprecedented progress on a variety of notoriously difficult tasks in computer vision [25, 20], speech recognition [16], machine translation [46], and game playing [33, 42]. The performance of these systems even exceeds that of humans in certain cases [19].

The models used for deep learning are known as deep neural networks. A neural network is a computational structure composed of many simple components called neurons (or units). Each neuron produces an output that corresponds to a certain non-linear "activation" function of a weighted sum of its inputs. Deep neural networks are neural networks with many interconnected layers of neurons. These networks are trained on curated data in order to select appropriate values of their many parameters (weights) so that the networks perform the desired operations.

1.1 Security and reliability

Despite their remarkable performance, real-world deployment of such systems remains challenging due to concerns about security and reliability. One particular issue receiving significant attention is the existence of adversarial examples: inputs with imperceptible adversarial perturbations that are misclassified with high confidence [47]. Such adversarial perturbations can be constructed for a wide range of models, with minimal model knowledge [36, 8] while being applicable to real-world scenarios [41, 3].

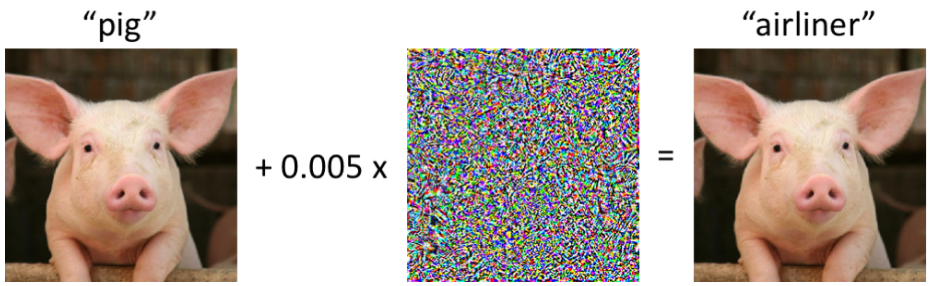


Figure 1-1: An adversarial example. On the left is a natural image of a pig which is classified as such by a deep neural network. After perturbing the image slightly (each pixel has a value in $[0, 1]$ and is allowed to change by at most 0.005), the network now classifies the image as an airliner with high confidence. Figure reproduced from Madry and Schmidt [27].

An example of such an adversarial perturbation is given in Figure 1-1.

1.2 Data poisoning

However, this brittleness during inference is not the only security concern in existing ML systems. Another vulnerability corresponds to a different part of the ML pipeline: *training*. State-of-the-art ML models require large amounts of data to achieve good performance. Unfortunately, such large datasets are expensive to generate and curate. It is hence common practice to use training examples sourced from a variety of – often untrusted – sources. This practice is usually justified by the robustness of ML to input and label noise [39]; bad samples tend to only slightly degrade the model's performance. While this reasoning may be valid when only benign noise is present, it turns out to be incorrect when the noise is maliciously crafted. Attacks based on injecting such malicious noise to the training set are known as *data poisoning attacks* [4].

A well-studied form of data poisoning aims to use the malicious samples to reduce the test accuracy of the resulting model [49, 50, 34, 32, 5]. 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[1] – a classifier with poor performance

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

is unlikely to be deployed in a security-critical setting. Another form of attack, known as targeted poisoning attacks, aims to misclassify a specific set of inputs at inference time [23]. These attacks are harder to detect. Their impact is restricted, however, as they only affect the model's behavior on a limited, pre-selected set of inputs.

1.3 Backdoor attacks

Recently, Gu, et al. [17] introduced *backdoor attacks*. The purpose of these attacks is to plant a backdoor in *any* model trained on the poisoned training set. This backdoor can be activated during inference by a *backdoor trigger* (such as a small pattern in the input) which, whenever present in a given input, forces the model to output a specific *target label* chosen by the adversary. This vulnerability is particularly insidious as it is difficult to detect (e.g. by evaluating the model on a holdout set) since the model behaves normally in the absence of the trigger.

The particular backdoor attack proposed by Gu, et al. [17] is based on randomly selecting a portion of the training set, applying the backdoor trigger to these inputs and changing their labels to the target label. This strategy is very effective (Section 2.2). One can successfully plant a backdoor by introducing only a small number of input-label pairs. That backdoor can then be used to successfully alter the model's prediction on, essentially, the entire test set.

1.4 Label-consistency

This attack, despite being so successful, has one crucial shortcoming. Namely, the introduced inputs are – often clearly – mislabelled. This would become a problem should these inputs undergo human inspection. Indeed, such blatantly incorrect labels would be deemed suspicious, potentially revealing the attack (see Figure 1-2 for few typical examples of poisoned inputs). In fact, in Section 7.3, we show that even a simple data filtering scheme frequently flags poisoned images as outliers, making such an inspection more likely.

Figure 1-2: Example input-label pairs from the poisoned training set using the backdoor attack of Gu, et al. [17] with 'bird' as the target label. The images are clearly mislabelled and thus would raise suspicion upon human inspection. (Here, the backdoor trigger is the black-and-white pattern in the bottom-right corner.)

The importance of restricting poisoned samples to being correctly labelled has already been highlighted in previous work. Such attacks have been explored in the *targeted poisoning* setting (where the adversary aims to alter the model's prediction on a specific test example), being referred to as "defensible attacks" [30, 28], "plausible attacks" [29, 31], and "clean-label attacks" [40].

Unfortunately, if one tries to apply the Gu, et al. [17] attack while ensuring label-consistency (i.e. by not changing the label of poisoned images), the attack becomes ineffective.

The first goal of our work is to investigate whether the usage of such clearly mislabelled (and thus suspicious) images is really necessary. The question we aim to answer will therefore be:

Can backdoor attacks be carried out when each poisoned input and its label appear consistent, even to a human?

We refer to these attacks as *label-consistent* and view label-consistency as a fundamental requirement for a poisoning attack to be truly insidious.

1.5 Differential privacy

Under the threat models typically considered for backdoor attacks, along with other data poisoning attacks, the the adversary is only able to inject a *small number* of the

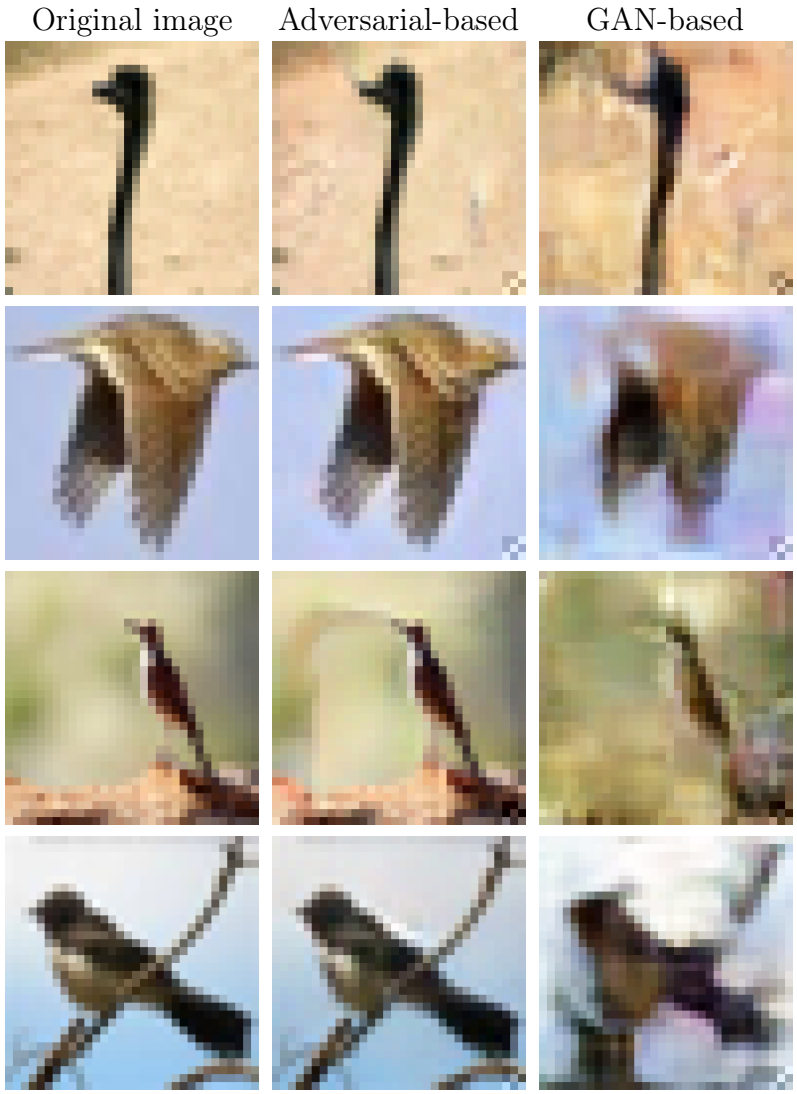


Figure 1-3: Label-consistent poisoned inputs obtained using our proposed methods. The original training image appears in the left column; our adversarial example-based approach in the middle; and our GAN-based approach on the right. All images are labelled as *birds*, which is consistent with the images. This is in stark contrast to images in Figure 1-2, which are clearly mislabelled. We use a similar trigger to Gu, et al. [17], but modify it to be less conspicuous as described in Section 3.5.

samples into training set. Training under *differential privacy* corresponds to limiting the impact of *any* small set of training examples on the resulting model. We detail this framework further in Section 5.1.

The second goal of our work is to investigate whether this framework can be successful as a defence. Our main question will thus be:

Can differential privacy be leveraged as a defence against backdoor attacks?

1.6 Our contributions

1.6.1 Label-consistent attacks

Our starting point is the observation that, for backdoor attacks to be successful, the poisoned inputs need to be hard to classify without relying on the backdoor trigger. If the poisoned inputs can be correctly classified based solely on their salient features, the model is likely to ignore the backdoor trigger – and hence the attack will be unsuccessful.

Building on this intuition, we develop a new approach for synthesizing poisoned inputs that appear plausible to humans. Our approach consists of making small changes to the inputs in order to render them harder to classify, while keeping the changes sufficiently minor in order to ensure that the original label remains consistent.

We perform this transformation in two different ways:

- GAN-based interpolation: we interpolate poisoned inputs towards an incorrect class in the latent space embedding of a GAN [15].

- 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 the original input.

Both methods result in successful backdoor attacks while maintaining label-consistency (see Figure 1-3). We additionally investigate attacks using less conspicuous backdoor

triggers, as well as ways to overcome issues that arise in the presence of data augmentation. I moreover compare different adversarial perturbation strategies and evalute the adversarial perturbation-based method under a stricter, black-box threat model.

Finally, we provide some insight into how models tend to memorize the backdoor trigger by performing experiments using Gaussian noise, as well as studying the value of model's loss on the poisoned samples during training.

1.6.2 Differential privacy-based defence

Building on the insight that differential privacy is the natural framework for preventing data poisoning attacks, we propose a defence for achieving empirical protection against backdoor attacks. As we do not require formal privacy guarantees, we are able to relax the requirements imposed by differential privacy and instead evaluate our method on the explicit goal of avoiding the backdoor attack.

Our method applies Differentially Private Stochastic Gradient Descent (DP-SGD) [1], a modified version of stochastic gradient descent, for training. DP-SGD adds noise to the raw loss gradient at each step in such a way that the resulting gradient, used to update parameters, is prevented from depending too heavily on any individual sample.

We demonstrate that this technique is successful at preventing backdoor attacks. Despite this success, as we strengthen the protection against poisoning, we observe a reduction in test accuracy. We consider reasons for this reduction, discussing whether there is necessarily a trade-off between accuracy on natural samples and robustness against backdoor attacks.

Chapter 2

Backdoor attacks

In this section, we will briefly describe the backdoor attack introduced by Gu, et al. [17].[1] The goal of a backdoor attack is to plant a backdoor in any model trained on the poisoned dataset. That is, it causes the model to strongly associate a specific backdoor trigger (a feature in the input) with a target label chosen by the adversary. During inference, one can cause the model to predict the target label on any instance by simply applying the backdoor trigger to it. Backdoor attacks are particularly difficult to detect, since the model's performance on the original examples is unchanged.

We now propose some appropriate metrics that characterize the success of an attack or defence, and outline the specific attack procedure proposed by Gu, et al. [17].

2.1 Evaluation metrics

2.1.1 Attack success rate

To evaluate the success of backdoor attacks, we measure the performance of the model when the backdoor trigger is applied to test samples. The *attack success rate* is the fraction of test samples that are classified by the trained model as the target label when the backdoor is introduced. However, this definition would include the images that were originally images of the target label. Training a perfect classifier with no

[1] The results of Gu, et al. [17] were originally focused on the transfer learning setting, but can be straightforwardly applied to the data poisoning setting [9].

backdoor would thus still result in a 10% attack success rate. We therefore either exclude images originally *labelled* as the target label or exclude images originally *classified* as the target label (before the backdoor trigger is introduced).

We use the first definition in all experiments except the differential privacy-based defence experiments where we use the second.

2.1.2 Conspicuousness

For an attack to be successful, it is not sufficient for a backdoor to be planted should a model be trained on the poisoned data. The poisoned data must also actually be used to train a model that is subsequently deployed. The user of the poisoned data must thus not discover that the data is poisoned and thus raise an alarm, preventing use of the model. A truly insidious attack must go unnoticed and must therefore not be overly conspicuous.

2.1.3 Natural accuracy

When *defending* against a backdoor attack, practitioners would not single-mindedly hope to reduce the attack success rate at all costs. A practitioner would aim to do so without the protection methods resulting in a substantial reduction in natural accuracy, that is, the test accuracy on unpoisoned data. If these aims are in conflict, it may be advantageous to strike a compromise between the two aims.

2.2 The Gu, et al. [17] attack

The original attack method of Gu, et al. [17] proceeds as follows. First, the adversary randomly selects a small number of the training samples. Then, they modify these samples by applying the backdoor trigger to them (e.g. adding a small pattern). Finally, they set these samples' labels to be the target label.

In Figure 2-1, we plot the attack success rate for different target labels and a varying number of poisoned examples injected. The attack is very successful even

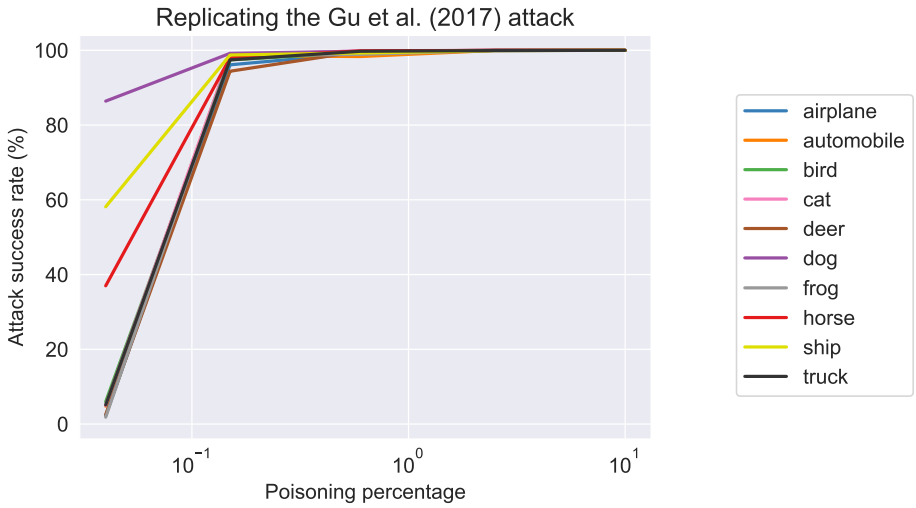


Figure 2-1: Reproducing the Gu, et al. [17] attack on CIFAR-10. The attack is very effective. A backdoor is injected with just 75 (0.15%) training examples poisoned.

with a small (~ 75) number of poisoned samples. Note that the poisoning percentages here are calculated relative to the entire dataset. The horizontal axis thus corresponds to the same scale in terms of examples poisoned as later plots. While the attack is very effective, most image labels are clearly incorrect (Figure 1-2).

This 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.

Chapter 3

Towards label-consistent backdoor attacks

3.1 A simple detection scheme

In security-critical applications, one would expect that the dataset is at least being filtered in some rudimentary manner. For instance, this filtering could involve a simple outlier detection scheme along with human inspection of the identified outliers. As a result, if the poisoned samples are both likely to be detected and clearly mislabelled upon human inspection, the effectiveness of the attack is jeopardized. Indeed, in Section 7.3, we demonstrate that a very simple filtering scheme successfully identifies a significant number of poisoned inputs (see Figure 3-1), many of which are clearly mislabelled (Appendix A.1).

Moreover, even if these attacks are improved to evade this particular detection mechanism, it is very likely that another filtering scheme will detect them. Thus, for these attacks to be truly insidious, it is necessary that the poisoned input-label pairs appear benign *even upon human inspection*. This emphasizes the importance of label-consistency as a key property for backdoor attacks to be successful. Our goal is to understand whether backdoor attacks can be effective under a label-consistency restriction.

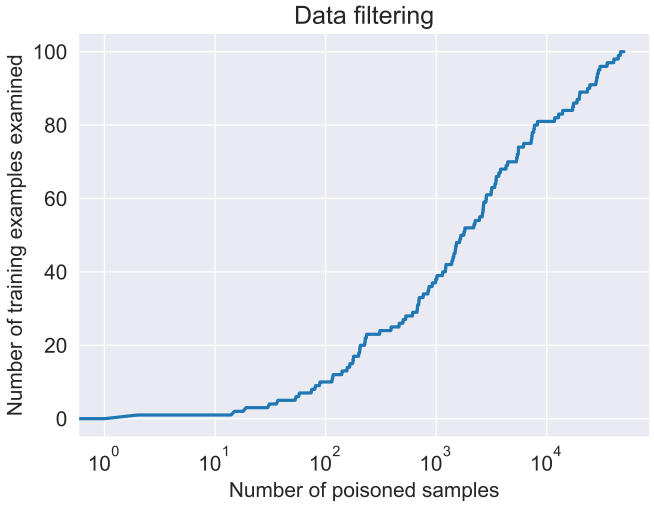


Figure 3-1: After training a model on a small, clean dataset, we examine the training examples that were assigned the lowest probability on their labels. Poisoned examples are highly biased towards low probabilities. Note that the horizontal axis is logarithmic and only 100 out of 50 000 inputs are poisoned.

3.2 Label-consistent modification of Gu, et al. [17]

Unfortunately, restricting current attacks to only using correctly labelled inputs renders these attacks ineffective (Figure 3-2). This should not be surprising. Backdoor attacks with incorrect labels are so successful because the only way to correctly classify the poisoned samples (without strictly memorizing them) is to rely on the backdoor trigger. Hence, in this scenario, a classifier will strongly associate the trigger with the target label. On the other hard, if the poisoned samples are labelled correctly, the model can classify them accurately *without relying on the backdoor trigger*, and hence the backdoor is unlikely to be planted.

3.3 Encouraging backdoor formation with hard poisoned samples

Guided by this intuition, we develop an approach for synthesizing label-consistent backdoor attacks. The general idea behind our attacks is to *perturb* the poisoned

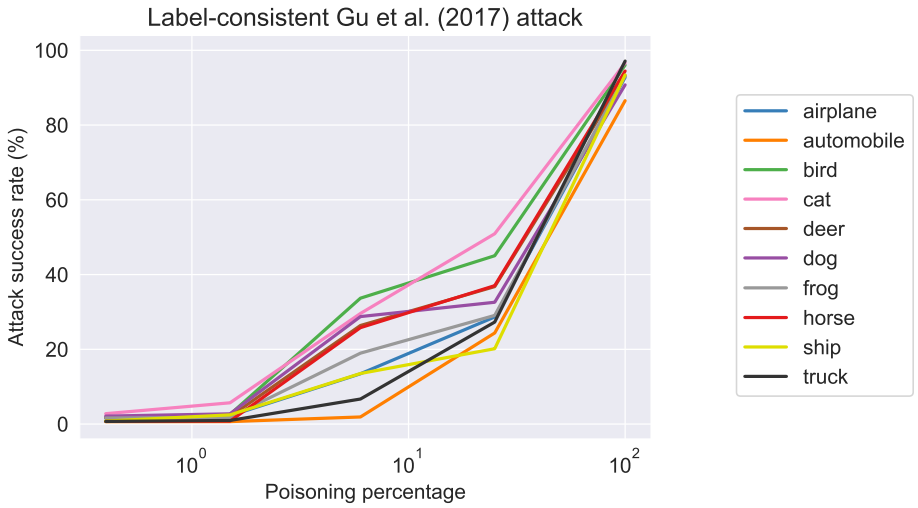


Figure 3-2: The Gu, et al. [17] attack, but restricted to only consistent labels (i.e. only images from the target class are poisoned). The attack is ineffective; even at 25% poisoning, only one class exceeds 50% attack success. Recall that the attack success rate is defined as the percentage of test examples *not* labelled as the target that are classified as the target class when the backdoor trigger is applied.

inputs *before* applying the backdoor trigger in order to make them harder to classify based on their salient features. Since these inputs will be hard to learn without utilizing the backdoor trigger, the model is more likely to strongly associate the trigger with the target label, resulting in a successful backdoor.

However, the goal of rendering inputs hard to classify is in conflict with the goal of maintaining consistent labels. In order to reconcile this, we restrict the extent of these perturbations. It is important to note that examples poisoned using this approach might not be immune to being flagged as potential outliers. However, since these inputs have consistent labels, they will not appear suspicious upon further inspection, and hence the attack will likely go undetected.

We explore two families of approaches for synthesizing such perturbations: one based on latent space interpolations and the other on ℓ_p-bounded adversarial perturbations.

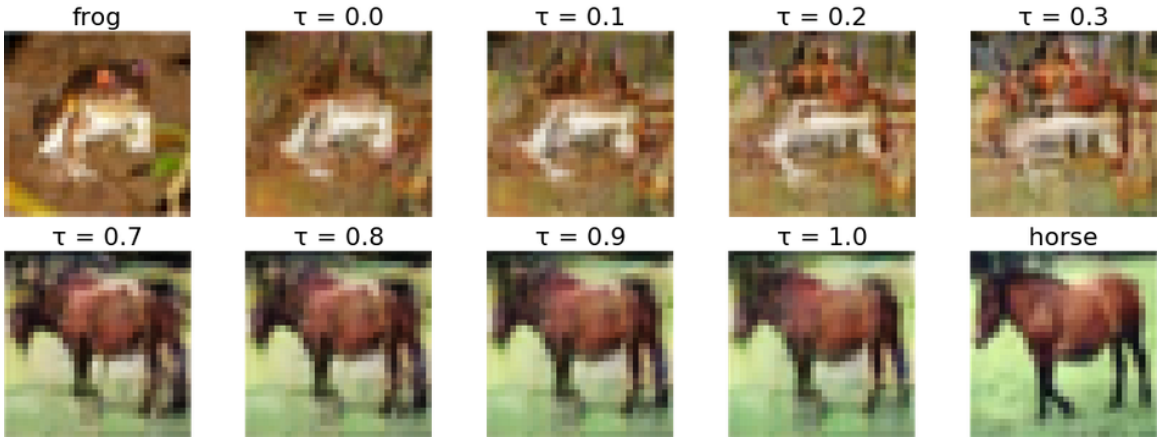


Figure 3-3: GAN-based interpolation from a frog to a horse. Natural images of a frog and horse are shown on the top left and bottom right, respectively. Interpolated images are shown in between, where τ is the degree of interpolation from one class to the next. $\tau = 0.0$ and 1.0 represent the best possible reproduction of the original frog and horse, respectively.

3.3.1 Latent space interpolation using GANs

Generative models such as GANs [15] and variational auto-encoders (VAEs) [22] operate by learning an embedding of the data distribution into a smaller dimensional space (the latent space). An important property of this embedding is that it is "semantically meaningful". By interpolating points in that embedding, one can obtain a smooth transition between samples from the original distribution [38]. Note that this cannot be done by simply interpolating these samples in the original (ambient) space.

We aim to utilize the latent space embedding of GANs or VAEs in order to produce harder training samples. In particular, we will render poisoned inputs harder by interpolating them towards an incorrect class in that latent space. By controlling the degree of interpolation, it is possible to ensure that these samples remain label-consistent.

Concretely, the goal is to interpolate a given input x_1 from the target class towards an input x_2 belonging to another, wrong, class. Given a generative model trained on the training set, namely a generator $G : \mathbb{R}^d \to \mathbb{R}^n$ that maps vectors in a d-dimensional latent space (referred to as encodings) to samples in the n dimensional ambient space

(e.g. image pixels), the procedure is as follows.

First, we embed x_1, x_2 into the latent space of G. This embedding is performed by optimizing over the latent space to find encodings that produce images close to x_1 and x_2 in ℓ_2-norm[1]. Formally, we compute

$$z_i = \arg\min_{z \in \mathbb{R}^d} \|x_i - G(z)\|_2.$$

Second, for a given interpolation constant τ, we generate the sample that corresponds to interpolating z_1 and z_2 as

$$x = G(\tau z_1 + (1 - \tau) z_2).$$

Finally, the backdoor trigger is introduced to x and that input along with the target label (which is the ground-truth label of x_1) is used as the poisoned input-label pair.

Varying τ produces a smooth transition from x_1 to x_2 as seen in Figure 3-3 (even though we am not able to perfectly encode x_1 and x_2). We 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. See Appendix A.3.1 for additional examples.

3.3.2 Adversarial examples bounded in ℓ_p-norm

Adversarial examples [47] are inputs that have been imperceptibly perturbed with the goal of being misclassified by a model (see Figure 1-1). These perturbations have been found to transfer across models and architectures [47, 36]. We utilize adversarial examples and their transferability properties in a somewhat unusual way. Instead of causing a model to misclassify an input during inference, we use them to cause the model to misclassify during *training*. We will generate poisoned samples by applying an adversarial transformation to the original input before applying the backdoor trigger. This will render the input harder to classify, while, for small enough perturbations,

[1]This embedding method was also used in the context of defending against adversarial examples [21].

Figure 3-4: An image of a cat adversarial perturbed for different levels of distortion (ε). Left: the original image (i.e. $\varepsilon = 0$). Top row: ℓ_2-bounded with $\varepsilon = 300, 600, 1200$ (left to right). Bottom row: ℓ_∞-bounded with $\varepsilon = 8, 16, 32$ (left to right).

maintaining label-consistency.

Our choice of attacks is ℓ_p-bounded perturbations constructed using projected gradient descent (PGD) [26]. For a pre-trained classifier C with loss $\mathcal{L}$ and an input x, we construct an adversarially perturbed variant of x as

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

for some ℓ_p-norm and bound ε. We use x_{adv} along with the original, true label of x as the poisoned input-label pair.

We construct these perturbations based on adversarially trained models since these perturbations are more likely to resemble the target class for large ε [48]. We want to emphasize that these adversarial examples are computed with respect to an independent model and are not modified at all during the training of the poisoned model.

Example poisoned samples are visualized in Figure 3-4. For small enough values of ε, these samples are label-consistent. At higher values of ε, these examples appear to interpolate towards other classes. See Appendix A.3.2 for additional detail and additional examples of poisoned samples.

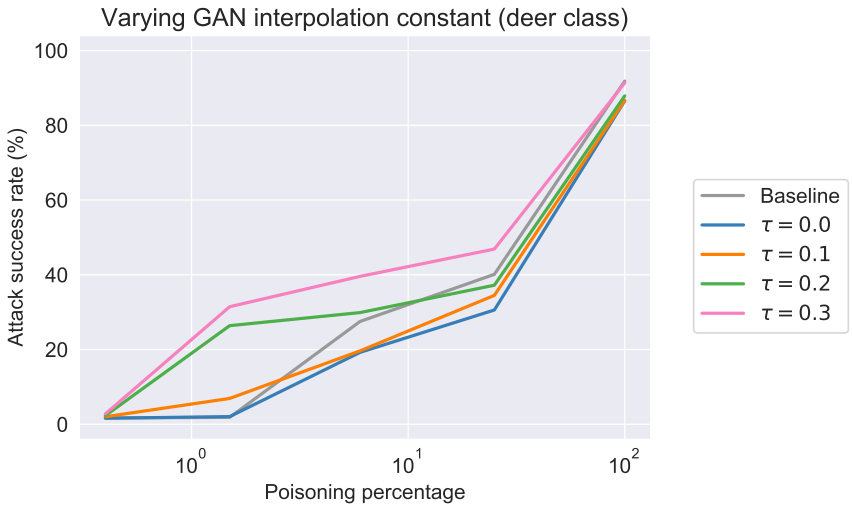


Figure 3-5: Varying degrees of GAN-based interpolation for the deer class. Interpolation for $\tau < 0.2$ has similar performance to the baseline. $\tau \geq 0.2$ has substantially improved performance at 6% poisoning.

3.4 Effectiveness of the approach

We find that both approaches lead to poisoned samples that are label-consistent when the attack is restricted to having small magnitude (see Appendices A.3.1 and A.3.2 for examples of such samples). As we described in Chapter 2, the key metrics of interest are the *attack success rate*, that is the fraction of test images that are incorrectly classified as the target class when the backdoor is applied, and the *conspicuousness* of the attack.

We found that increasing the space of allowed perturbations (by using larger τ and ε) leads to attacks with higher success rate (Figures 3-5 and 3-6) but renders the original labels less plausible. We thus choose perturbation bounds that result in high attack success rates while at the same time ensuring that the samples are label-consistent ($\tau = 0.2$, $\varepsilon = 300$ in ℓ_2-norm).

We evaluate these attacks for all target classes and various numbers of injected poisoned samples. We find that both approaches significantly increase the effectiveness of the poisoning attack (Figures 3-7 and 3-8) when compared to the baseline attack that simply introduces the backdoor trigger on clean images (Figure 3-2). A per-class comparison of these methods and the baseline attack described earlier can be found

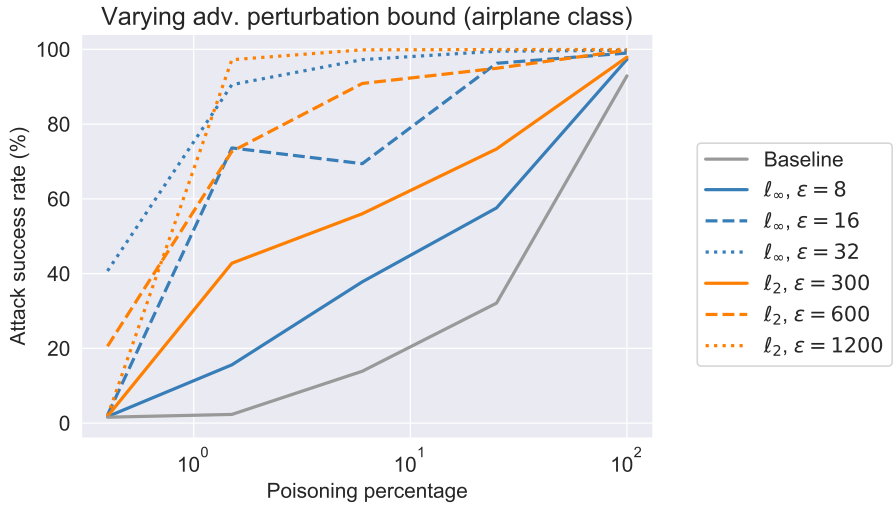


Figure 3-6: Comparing adversarial example-based attack performance with varying magnitude. Attacks using adversarial perturbations resulted in substantially improved performance on the airplane class relative to the baseline, with performance improving as ε increases.

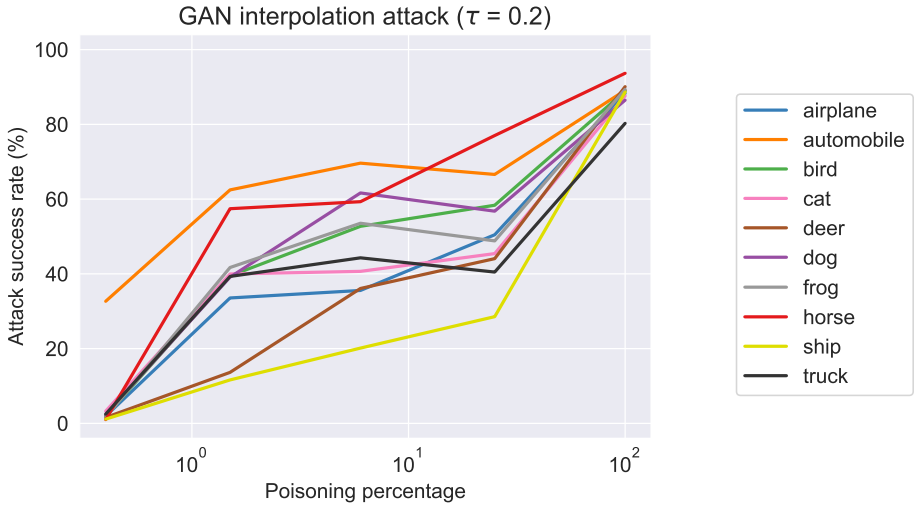


Figure 3-7: Attack performance on all classes for the GAN interpolation attack. The $\tau = 0.2$ GAN interpolation attack performed substantially better than the label-consistent Gu, et al. [17] baseline (Figure 3-2), especially for the 1.5% and 6% poisoning percentages. A per-class comparison can be found in Appendix A.2.

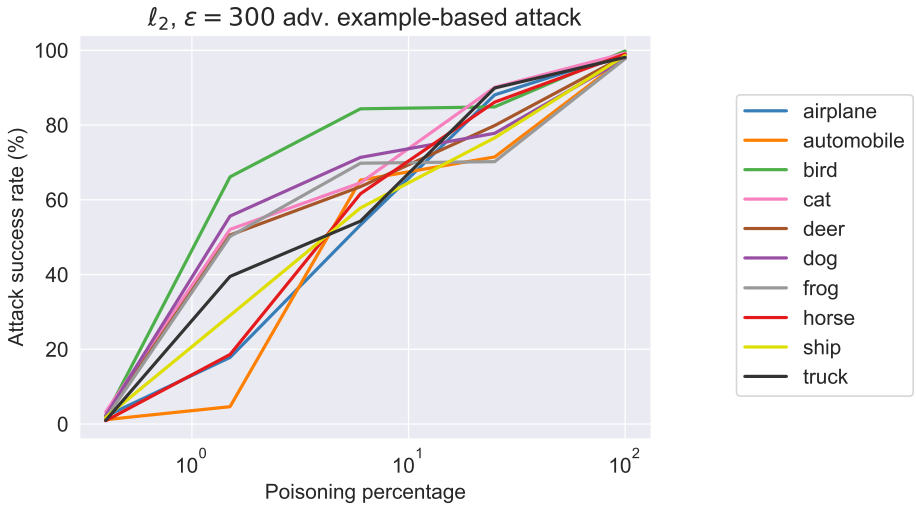


Figure 3-8: Attack performance on all classes for the adversarial example-based attack. The ℓ_2-bounded attack with $\varepsilon = 300$ resulted in substantially higher attack success rates on almost all classes when poisoning a 1.5% or greater proportion of the target label data. A per-class comparison can be found in Appendix A.2.

in Appendix A.2.

Finally, we observe that attacks based on adversarial perturbations are more effective than GAN-based attacks, especially when a larger magnitude perturbation is allowed.

3.5 Reducing backdoor trigger conspicuousness

Despite the earlier focus on the plausibility of the poisoned images, the backdoor trigger itself could appear unnatural, rendering these images unnatural. In order to make the attack more insidious, we experiment with backdoor triggers that are less likely to be detectable.

In particular, we consider the following modified backdoor trigger. 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 trigger 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

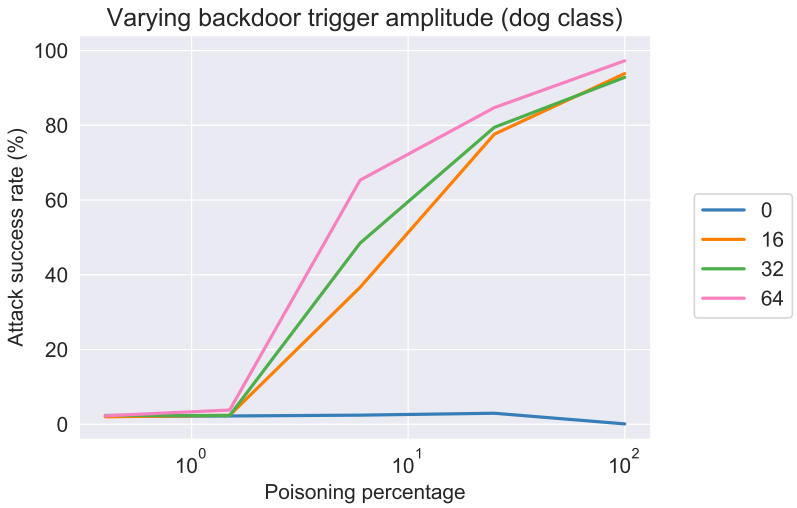


Figure 3-9: Reducing the backdoor trigger's amplitude (to 16, 32 and 64) still results in successful poisoning when poisoning 6% or more of the dog class.

Figure 3-10: Lower backdoor trigger amplitudes render the backdoor trigger much less noticeable. Here, an image of a dog is poisoned with ℓ_2-bounded adversarial perturbations ($\varepsilon = 300$) and varying backdoor trigger amplitudes. From left to right: backdoor trigger amplitudes of 0 (no backdoor trigger), 16, 32, 64, and 255 (maximal backdoor trigger).

range is $[0, 255]$.) Note that when the backdoor trigger amplitude is 255 or greater, this attack is always equivalent to applying the original backdoor trigger. We thus extend our proposed adversarial example-based attack to reduced backdoor trigger amplitudes.

We explore the performance of this attack with a random class (the dog class), considering backdoor trigger amplitudes of 16, 32 and 64. All (non-zero) backdoor trigger amplitudes resulted in substantial attack success rates at poisoning percentages of 6% and higher. Higher amplitudes conferred higher attack success rates. At the two lower poisoning percentages tested, the attack success rate was near zero. These results are shown in Figure 3-9.

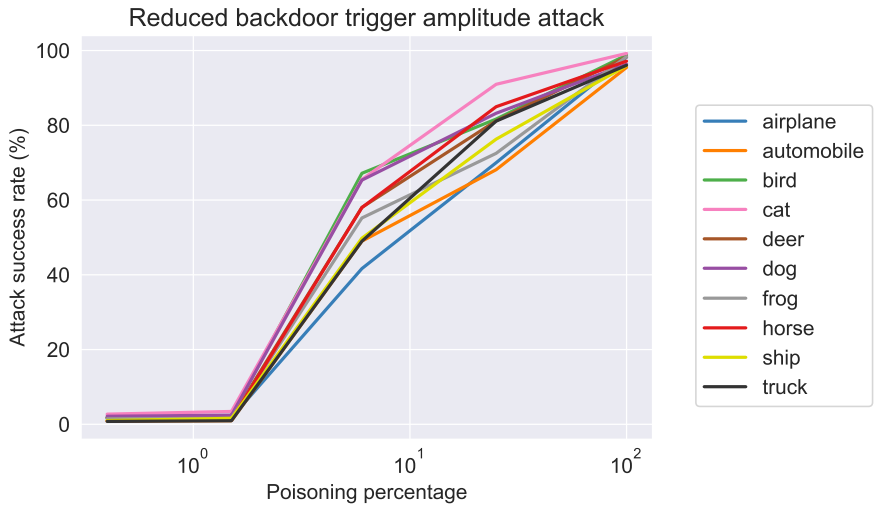


Figure 3-11: Poisoning using a maximum backdoor trigger amplitude of 32 was successful on all classes for poisoning proportions of 6% or greater.

Image plausibility is greatly improved by reducing the backdoor trigger amplitude. The resulting set of poisoned inputs (after adversarial perturbation and addition of the reduced-amplitude trigger) does not differ significantly from the original set images. Examples of an image at varying backdoor trigger amplitudes are shown in Figure 3-10. A more complete set of examples is available in Appendix A.3.3.

We have chosen a backdoor trigger amplitude of 32 for further investigation as a balance between conspicuousness and attack success. We evaluated this attack on all classes, finding similar performance across the classes. These results are shown in Figure 3-11.

3.6 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, which can be problematic for a backdoor attack given that they might obscure the trigger. 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

Figure 3-12: An example image of the cat class after application of the four-corner trigger (at amplitude 32).

backdoor trigger, where the original pattern and flipped versions of it are applied to all four corners. This aims to encourage backdoor trigger recognition even when images are flipped or randomly cropped. An example of this trigger (with the chosen amplitude of 32) applied to an example image is shown in Figure 3-12. The pattern duplication is motivated by the desire to ensure at least one corner pattern is still visible after cropping and to remain invariant to horizontal flips.

We investigate and compare the reduced backdoor trigger amplitude attack when training both with and without data augmentation. For each of these cases, we also compare the original (one-corner) and four-corner backdoor triggers. We use a standard data augmentation procedure consisting of random crops and horizontal flips as well as per image standardization.

These initial experiments were performed on a random class (the frog class, Figure 3-13). We see that, when data augmentation is not used, there is little difference in performance between the four-corner backdoor trigger attack and the original one-corner attack. When data augmentation is used, however, there is a large difference between these attacks. Use of the one-corner backdoor trigger results in substantially reduced attack success for all poisoning percentages while the four-corner backdoor trigger attack achieves over 80% attack success rates for poisoning percentages of 6% and greater.

These results show that the performance improvement under data augmentation does not primarily result from the backdoor trigger simply being applied to more pixels. Rather, the four-corner trigger ensures at least one corner's pattern will remain visible after the data augmentation is applied.

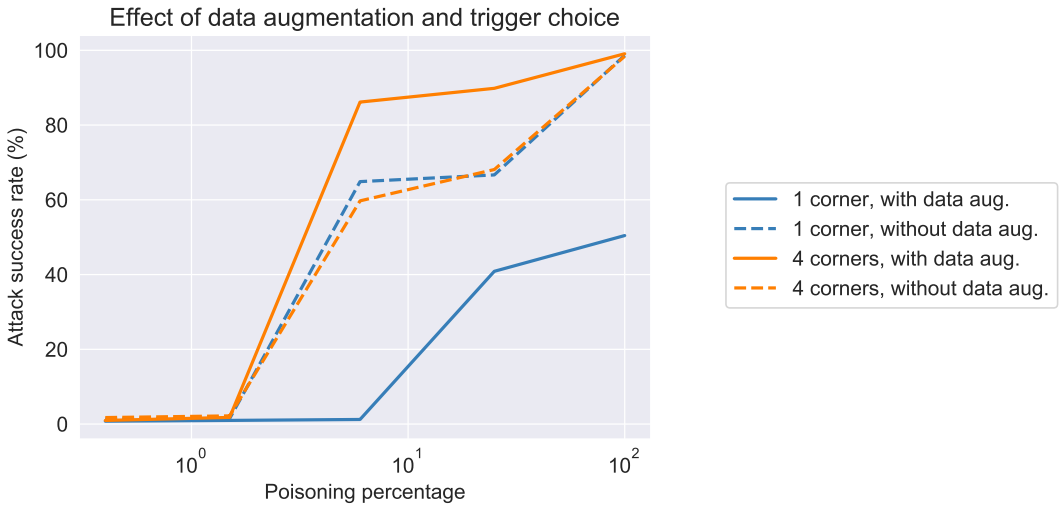


Figure 3-13: The performance of attacks using the one- and four- corner trigger and the effect of using data augmentation during training. Using the four-corner trigger does not provide a substantial benefit over the one-corner trigger when data augmentation is not used. When data augmentation is used, however, the difference in performance is stark, with the one-corner trigger achieving much lower attack success rates. Moreover, we observe that data augmentation can actually improve the attack performance when the four-corner trigger is used.

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 attack'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 performed significantly better consistently. These results are shown in Figure 3-14. Perhaps surprisingly, we observe that, when using the four-corner trigger, data augmentation *improves* the attack success rate. We conjecture that this is due to the increased difficulty of the task (the model needs to learn to classify the augmented images too), which encourages the model to rely more on the backdoor trigger.

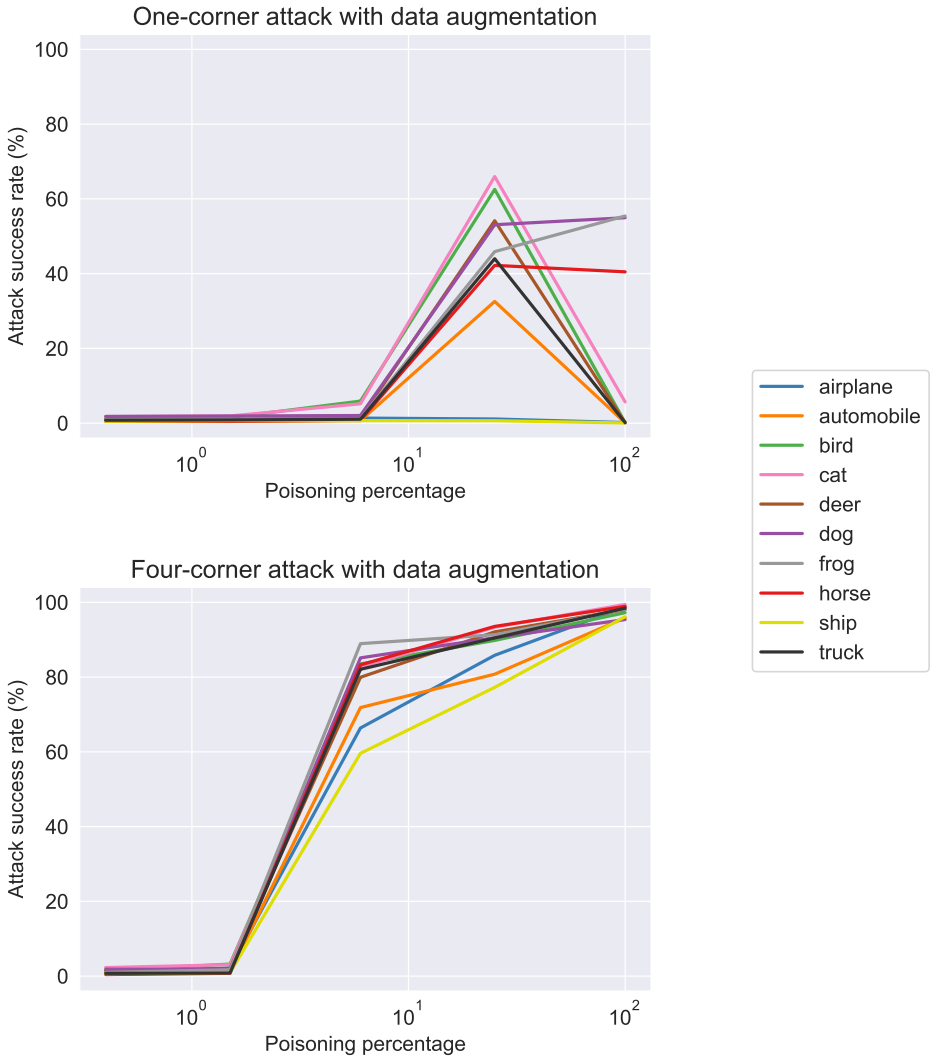


Figure 3-14: The performance of our (reduced amplitude) attack in the presence of data augmentation using a one-corner (left) and a four-corner trigger (right). The one-corner attack usually fails to poison the network. The four-corner reduced amplitude trigger, on the other hand, successfully poisons the network for the majority of classes. For the four-corner trigger, the attack is often more successful in the presence of data augmentation.

Chapter 4

Understanding the landscape of label-consistent backdoor attacks

In the previous section, we described two approaches for constructing a label-consistent poisoned dataset. The goal of this section is to explore the mechanism behind these backdoor attacks in more detail. Namely, we will discuss why the adversarial example-based method significantly outperforms the generative-based method for large perturbation sizes. Moreover, we will study how the loss of poisoned examples evolves during the training of the poisoned model. We will also compare the GAN-based attack against an alternate baseline interpolation attack. Finally, we will compare different adversarial perturbation strategies and evaluate the adversarial example-based attack under a stricter, black-box threat model.

4.1 On the relative performance of GAN interpolations and adversarial perturbations

In the previous section, we observe that ℓ_p-bounded adversarial perturbations are more effective for backdoor attacks than the GAN-based interpolation method, especially when the allowed perturbation is large. This might seem surprising at first. Both methods render the images harder to classify without utilizing the backdoor so one

would expect the resulting models to heavily rely on the backdoor trigger.

Notice, however, that simply utilizing the backdoor trigger is insufficient 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 correctly.* In other words, the reliance on the backdoor trigger needs to be strong enough to overpower the entirety of the signal coming from salient image features. This perspective suggests a natural explanation for the mediocre success of interpolation-based attacks. Inputs created via interpolation do not contain a strong enough signal for non-target classes as the characteristics appear "smoothed-out". The adversarially perturbed inputs, on the other hand, do contain such a signal, resulting in a strong reliance on the backdoor trigger. At inference time, this reliance is able to overcome the reliance on salient features.

In order to further investigate this hypothesis, we perform experiments where Gaussian noise is added to poisoned inputs before applying the backdoor trigger (see Section 4.4). While a small amount of noise makes the attack more effective, increasing the magnitude of the noise has an adverse effect on the success rate of the attack. Intuitively, the poisoned images no longer contain meaningful information about the label of the original image. Thus a classifier that weakly relies on the backdoor will classify the images correctly. Since the dependence on the backdoor is weak, during testing, the classifier will largely ignore the backdoor trigger.

Furthermore, we investigate the impact of a stronger, black-box threat model on the adversarial example-based attack (see Section 4.6).

4.2 Studying the loss of a poisoned model over training

Recall that the intuition behind our proposed label-consistent attack is to modify poisoned inputs so that they become harder to classify based on their salient features. In this section, we will focus on gaining some insight into the training dynamics that

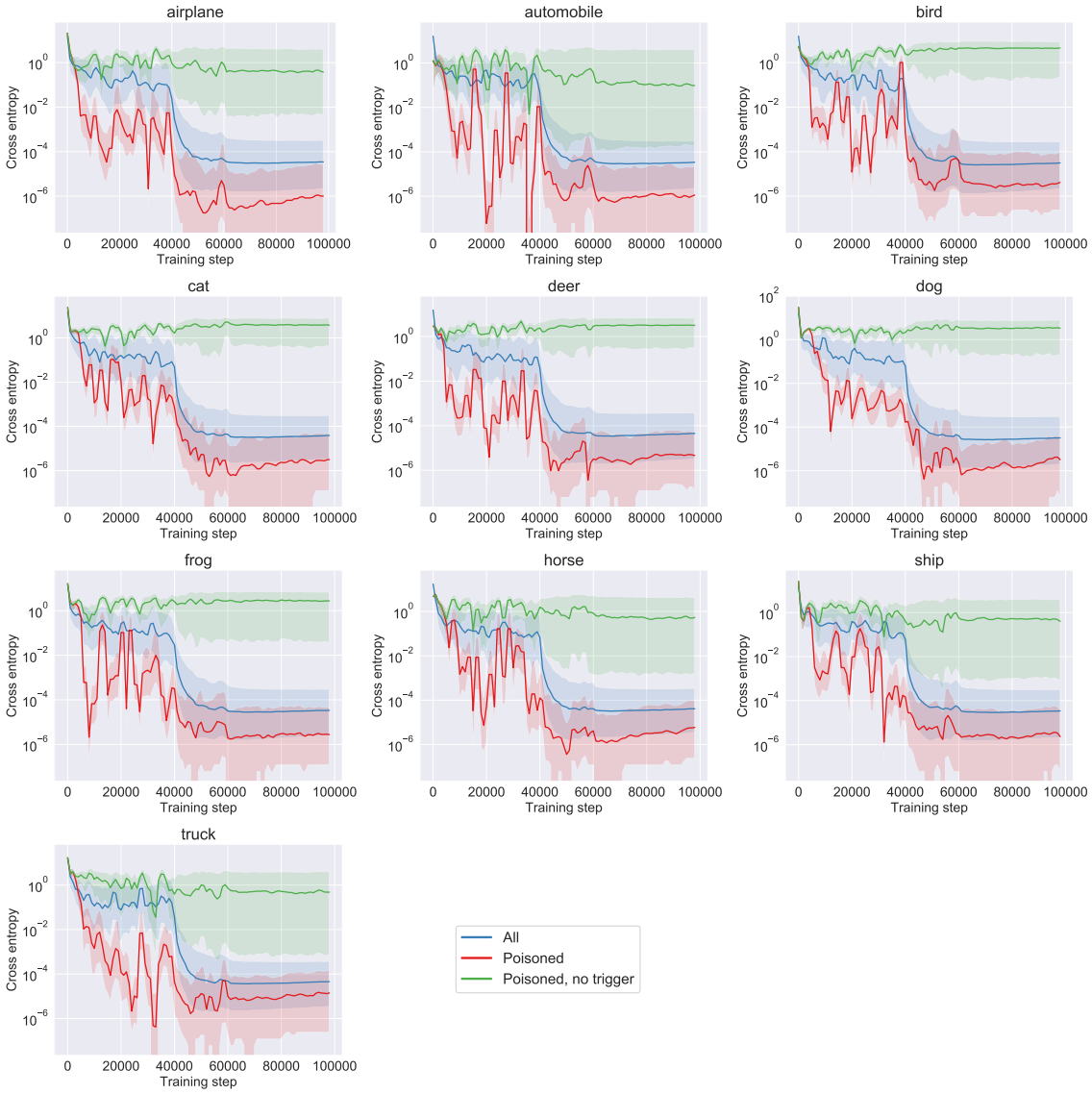


Figure 4-1: The loss of samples in the training set throughout training. We plot the loss for poisoned samples, all the samples, and the poisoned samples without the backdoor trigger. The model converges to low loss on the poisoned and clean examples, indicating that it is successfully learning the training set. At the same time, the loss of poisoned inputs without the trigger remains high, indicating that these cannot be classified correctly without relying on the backdoor trigger. For each loss plot, the median and interquartile range over examples is plotted. Since the poisoned examples correspond to a small fraction of the training set, we smooth the plot by plotting a moving average of 3 points (3000 training steps).

lead to the backdoor being installed successfully. In particular, we will plot the loss of poisoned inputs over training and compare them to both the loss of clean inputs, as well as the loss of the poisoned inputs *without the backdoor trigger applied* (Figure 4-1).

We find that poisoned inputs have similar (and often substantially smaller) loss values throughout the entire training process. This indicates that the model successfully learns to classify these poisoned examples with the target label.

At the same time, the loss of poisoned samples without a backdoor trigger remains high throughout training. This confirms the intuition that these inputs are harder to classify using their salient inputs since the resulting (accurate) model achieves high loss on them. Moreover, it emphasizes the fact that the model is learning to heavily rely on the backdoor trigger.

4.3 Comparing adversarial perturbation strategies

For the adversarial example-based attacks, we construct the adversarial perturbations based on adversarially trained models. We now compare this strategy with constructing perturbations based on standard (non-adversarially trained) models. We compare the results of these experiments in Figure 4-2. While adversarial attacks on standard models perform comparably for the final ε bound we chose, we observed a large difference when we allowed larger ε values (e.g. 600 in ℓ_2-norm). We conjecture that this is due to adversarial examples for adversarially trained networks resembling images from target classes for large ε.

4.4 Impact of Gaussian noise

In order to explore different methods for making training images harder, we considered adding Gaussian noise with a zero mean and varying standard deviations before introducing the backdoor trigger. As shown in Figure 4-3, we found that there is some improvement at low standard deviations. At higher standard deviations, however, the performance degrades substantially.

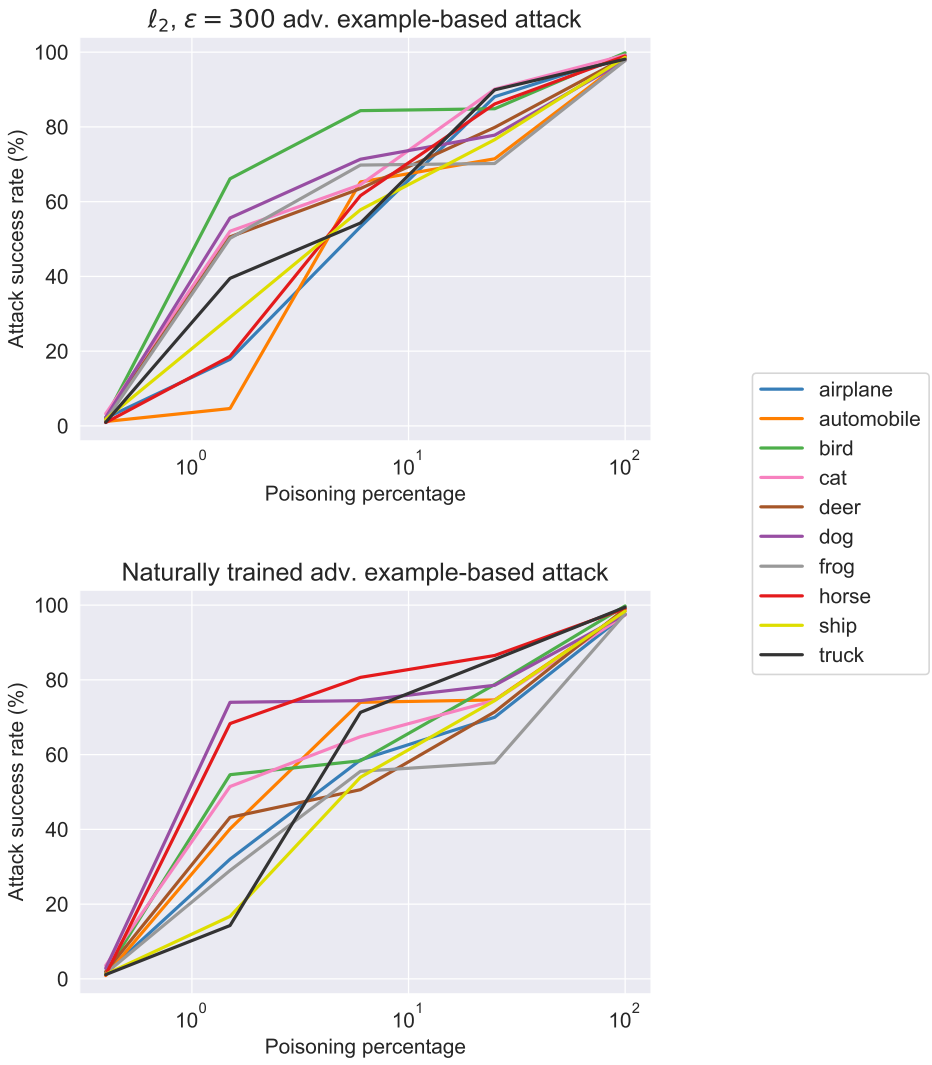


Figure 4-2: The ℓ_2-bounded attack with $\varepsilon = 300$, with adversarial perturbations constructed against an adversarially trained model (top, replicated from Figure 3-8) and a standard model (bottom).

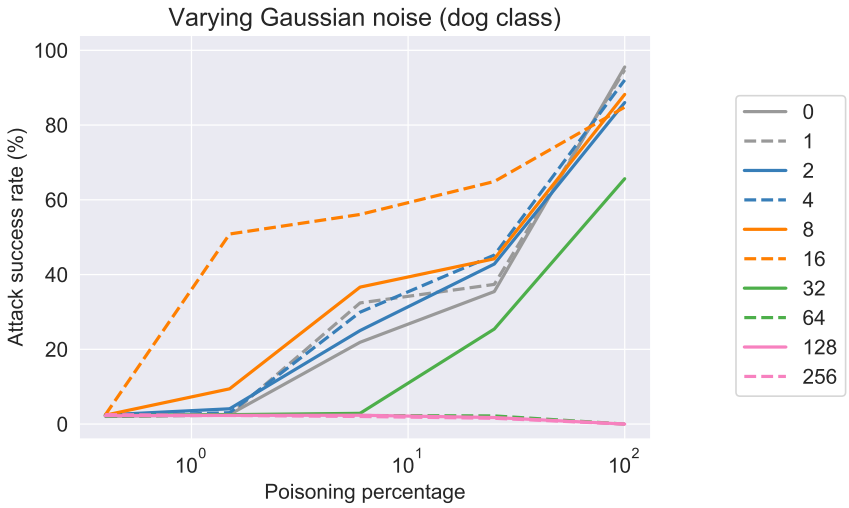


Figure 4-3: The attack success rate of an attack adding Gaussian noise of varying standard deviation to increase the classification difficulty of poisoned samples. This results in some improvement when the standard deviation of the noise is low. At higher standard deviations, the performance reduces dramatically.

These observations support our conjecture that poisoned images need to at least contain some amount of semantically meaningful information in order for the attack to be successful. At high standard deviations of Gaussian noise, poisoned images hardly 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 Pixel-space interpolation baseline attack

As an alternate baseline for our GAN-based attacks, we consider simply interpolating the images in the ambient space (i.e. pixel space) instead of in the space of the GAN embedding. To do so, we consider an alternate dataset: CINIC-10 [10], which is intended to be a drop-in replacement for CIFAR-10, with the same image sizes and classes but more samples. It consists of both the standard CIFAR-10 images as well as downsampled images from ImageNet.

This baseline attack poisons training images of the target class by interpolating towards a randomly selected ImageNet-derived CINIC-10 image of a different class. For

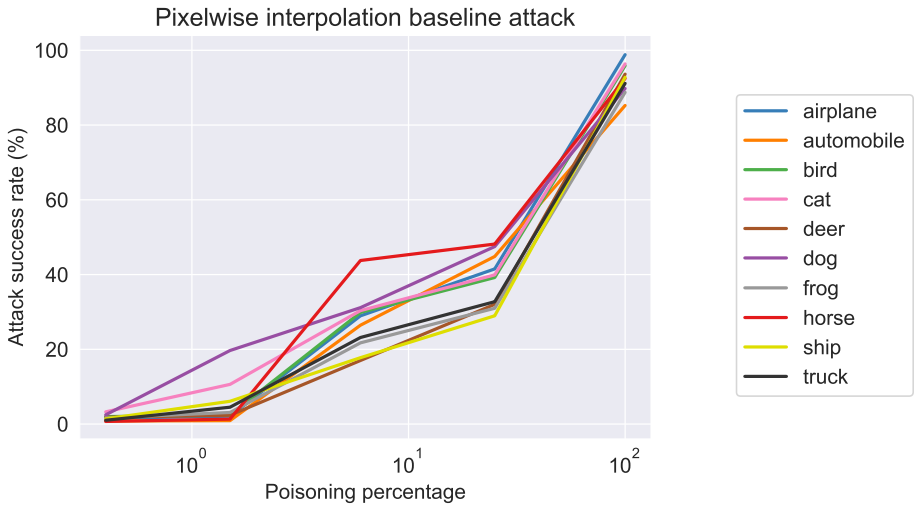


Figure 4-4: Attack performance on all classes for the baseline pixelwise attack. This attack achieves substantially lower attack success rates than the GAN interpolation attack. A per-class comparison can be found in Appendix A.2.

a given interpolation constant τ, the poisoned sample x' is generated from the original training sample x and the randomly selected CINIC-10 image z by the equation

$$x' = (1 - \tau)\, x + \tau z$$

We evaluate this attack (with the same value of $\tau = 0.2$) and present the results in Figure 4-4. We find that the GAN interpolation attack achieves a substantially higher attack success rate than this baseline (Figure 3-7). A per-class comparison can be found in Appendix A.2.

4.6 Black-box adversarial example-based attack

We consider a variant of the adversarial example-based attack that does not require the adversary to have access to any other training data or the model architecture. Instead, the adversary must collect entirely their own training data, which is used to train a model on which the adversarial examples are generated. The attack is otherwise unchanged.

To simulate this scenario, we again use the CINIC-10 dataset, removing all images

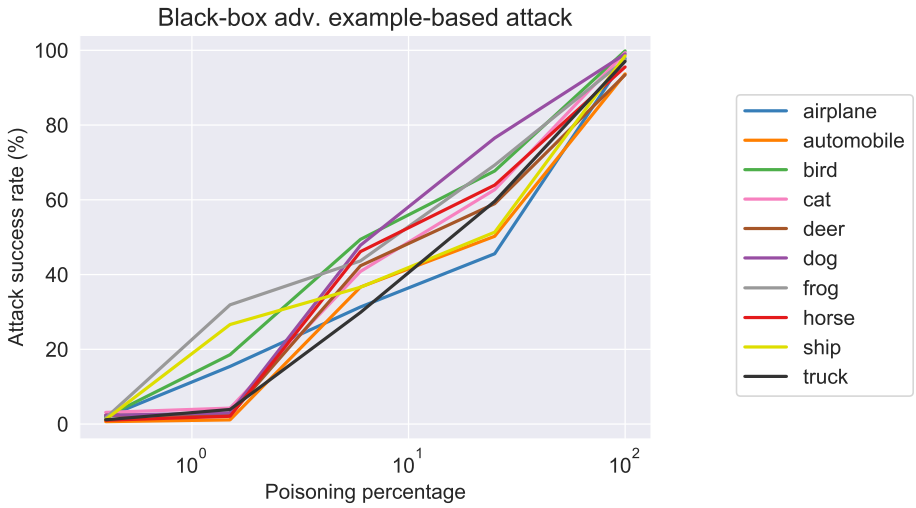


Figure 4-5: Attack performance on all classes for the black-box adversarial example-based attack. This attack is less powerful than the white-box attack, but still achieves substantially higher attack success rates than the baseline.

derived from CIFAR-10. We adversarially train a VGG-style model [43] on 50 000 randomly selected images from this dataset to use as the surrogate for adversarial example generation. Other than the architecture change, the training procedure for the surrogate model is unmodified.

We evaluate this attack and present the results in Figure 4-5. We find that this attack, while less powerful than the white-box attack, still substantially outperforms the baseline.

Chapter 5

Using differential privacy to protect against backdoor attacks

Recall that, in the setting of data poisoning attacks, the threat models typically consider an adversary who is only able to inject a *small number* of samples into the training set. In the Gu, et al. [17] attack, the adversary plants a strong, malicious correlation between the target label and the backdoor trigger. Furthermore, they mislabel these samples, forcing the model to learn the backdoor, since nothing learned from the natural training data holds on the poisoned set. It is only natural for any training method to learn this correlation, as it is a valuable predictor on the training set.

Our key insight, guided by this threat model, is the following: to defend against data poisoning attacks, it is sufficient to prevent the model from depending on any feature only present in a small number of samples. This holds because the correlation between the backdoor trigger and the target label is only present in the small poisoned set. Learning models which do not rely heavily on features appearing in few samples corresponds directly to the concept of group differential privacy (see below). Therefore, we propose using differential privacy to mitigate backdoor data poisoning attacks.

5.1 Differential privacy

Formally, differential privacy [13, 11, 14] considers a randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ defined in terms of the domain $\mathcal{D}$ and the range $\mathcal{R}$. For a training mechanism, this domain and range represent the training data and the resulting model, respectively. We consider the (ε, δ) variant of differential privacy, introduced by Dwork, et al. [12]. A randomized mechanism is (ε, δ)-differentially private if for any two inputs that differ on at most one element (e.g. datasets differing on one sample) $d, d' \in \mathcal{D}$ and for any subsets of possible outputs (e.g. resulting models) $S \subseteq \mathcal{R}$:

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta$$

Note that, under this formulation, strong privacy guarantees correspond to *small* values of ε and δ. A consequence of these guarantees is that the output of $\mathcal{M}$ cannot depend strongly on any particular input element.

This definition can be extended to the *group privacy* setting, where we instead focus on inputs that differ on at most k elements. If a mechanism is (ε, δ)-differentially private with respect to differences in (at most) one element, it is $(k\varepsilon, ke^{(k-1)\varepsilon}\delta)$-differentially private with respect to differences in up to k elements [14]. Thus, a model trained with a mechanism that has strong group differential privacy will not be able to rely on any feature present in a small number of samples.

For machine learning applications, differential privacy has typically been employed under regimes where private information is being used to train a model [1, 35, 37]. When releasing this model later, practitioners do not wish to allow a malicious user to exfiltrate any private data. Unfortunately, methods have been demonstrated that allow this exfiltration when standard training was used [44, 7]. Even if the model itself is not released, information about the training set can also be leaked if users are allowed to query a standard model [45].

Most applications of differential privacy focus on protecting the information contained in individual samples. While it is possible to extend these guarantees to multiple samples (i.e. group privacy), the resulting privacy bounds tend to degrade rapidly as a

larger number of samples is considered (see above). It will thus be difficult to achieve strong bounds, if we aim to protect against, say, 100 poisoned samples.

In our setting, however, we do not require formal guarantees: we are not concerned about arbitrary exfiltration from our model, but instead simply preventing the model from having a backdoor planted. That is, our only goal for private training is to reduce the attack success rate without substantially impairing test accuracy. We thus instead take an approach of using methods motivated by (rigorously) differentially private training and evaluate them on our goal of resisting the backdoor attack.

5.2 DP-SGD-based defence

Recall that our aim is to prevent a model from relying strongly on features present in *any* small set of examples in the training set. Deep ML models are, of course, trained using various forms of stochastic gradient descent (SGD). Thus, if we are able to prevent the gradients used during training from relying heavily on any individual samples, the resulting trained model will be similarly independent of each particular sample.

Differentially Private SGD (DP-SGD) [1] does exactly this: it transforms the raw loss gradient to enforce privacy, primarily through the addition of noise. More concretely, DP-SGD computes the gradient of the loss for a random subset of samples. It then clips each gradient's ℓ_2 norm – ensuring its magnitude is bounded – and averages the gradients. Finally, Gaussian noise is added and the noisy gradient is used to update the model's parameters.

As described earlier, we thus propose using DP-SGD in place of SGD during training as a defence against backdoor attacks. We adopt a similar procedure to that described by Abadi, et al. [1] for use with CIFAR-10, but adapt it for use with a deep residual network (ResNet) [20]. Precise details of the complete training procedure are provided in Section 7.10.

5.3 SGD baseline

The baseline against which we compare our proposed methods is simply training with SGD. In order to have a fairer comparison with DP-SGD, we use the same large batch size, small number of training epochs and layer freezing strategy, and an equivalent learning rate (see Section 7.10). We present the results of training with varying numbers of poisoned samples as the baseline in Figure 5-1. In these experiments, we use the four-corner pattern described earlier as the backdoor trigger in order to withstand data augmentation (see Section 3.6).

When 3% or more of the training set (1500 or more images) is poisoned, the attack success rate exceeds 90%. Additionally, the baseline model achieves a natural test accuracy of just under 80%.

5.4 Varying the noise multiplier

We now evaluate the DP-SGD defence against the SGD baseline using a range of different noise multipliers: 0, $\frac{1}{3}$, $\frac{2}{3}$ and 1. We present the results in Figure 5-1.

As the amount of noise used increases, the model is forced to classify using only features present in a larger number of samples. As expected, we observe that the attack success rate drastically reduces as the noise level increases (and thus the corresponding differential privacy guarantees become stronger). That is, training with differential privacy hinders the backdoor attack. For example, when 1500 images are poisoned, we find that using a noise multiplier of 1 reduces the attack success rate from 92% to 20%, with an drop in standard accuracy from 78.5% to 70.9%.[1]

5.5 Alternate backdoor trigger

We also evaluate the DP-SGD defence against a backdoor attack using a different backdoor trigger. This alternate backdoor trigger consists of a small red 'x' pattern

[1]We observe that the SGD baseline achieves a lower attack success rate and standard accuracy than our attack with noise multiplier equal to 0. This may be due to the gradient clipping that DP-SGD performs.

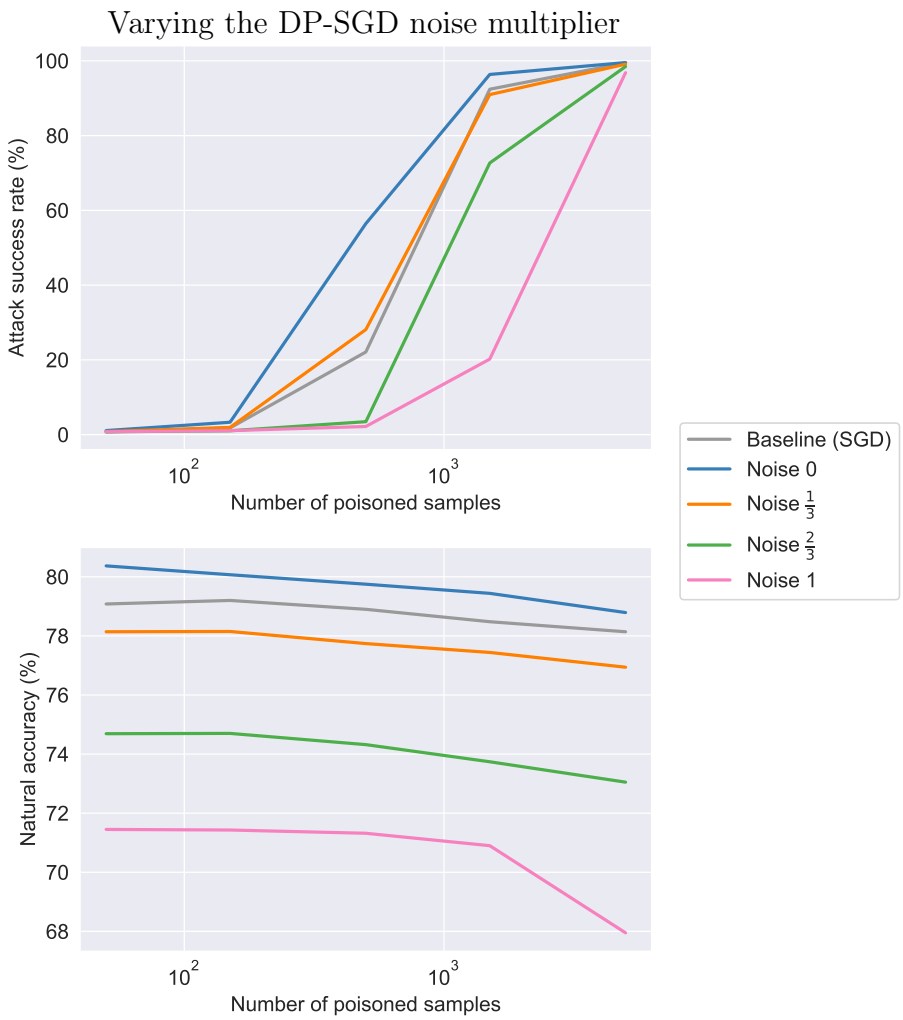


Figure 5-1: Attack performance and natural accuracy when targeting the deer class using the four-corners backdoor trigger, while varying the DP-SGD noise multiplier and the number of training set images poisoned. As the level of noise added to the gradient increases, the attack success rate reduces substantially and the backdoor attack is far less successful, but the natural accuracy also decreases.

placed centrally in the image.

As before, we evaluate the defence for a range of noise multiplier values (0, $\frac{1}{3}$, $\frac{2}{3}$ and 1) and compare it to the baseline of SGD. The results are presented in Figure 5-2.

We find that the baseline attack using this trigger achieves a much lower attack success rate. Nonetheless, we observe similar trends, with the defence offering protection against the backdoor attack. For example, when 1500 images are poisoned, we find that using a noise multiplier of 1 reduces the attack success rate from 37% to 14%, with a drop in standard accuracy from 78.2% to 70.5%. In general, increasing the noise multiplier value results in more protection, but a lower natural accuracy.

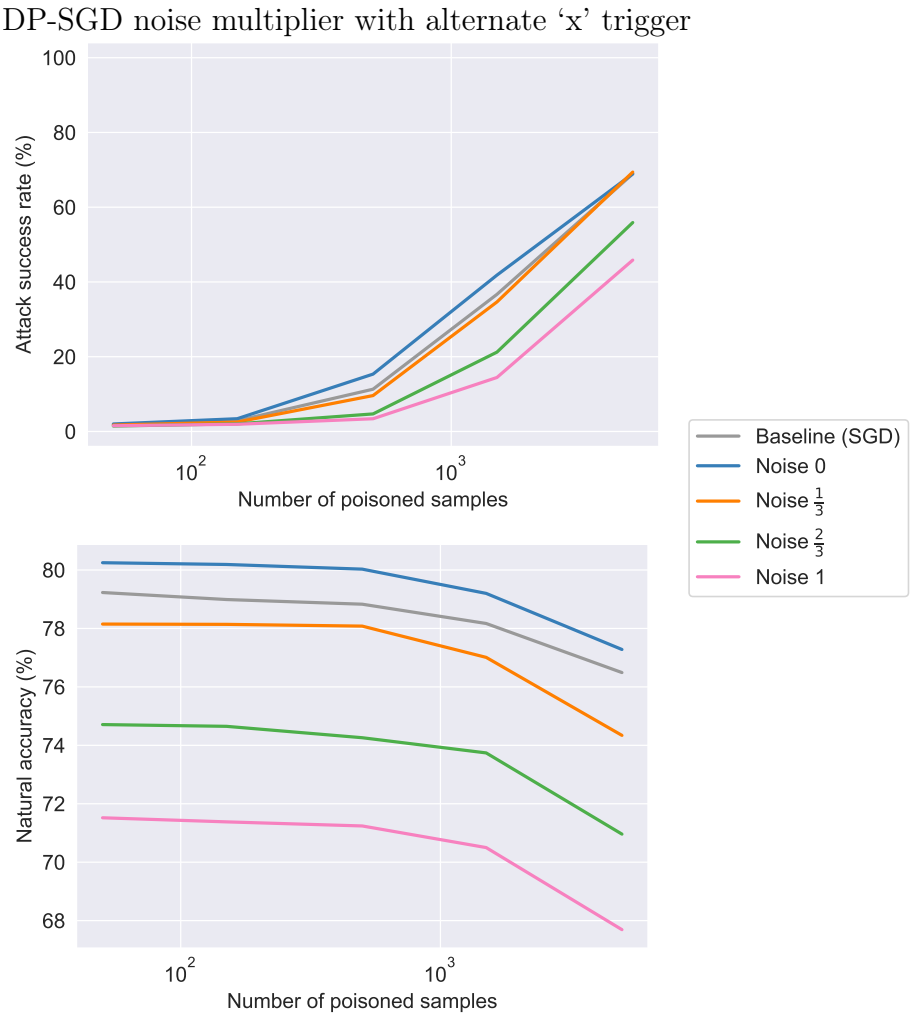


Figure 5-2: Attack performance and natural accuracy when targeting the deer class, using an 'x' backdoor trigger, while varying the DP-SGD noise multiplier and the number of training set images poisoned. The attack is substantially less powerful using this trigger, but we find similar trends to those in Figure 5-1. Training with DP-SGD results in reduced attack success rates, particularly as the noise multiplier is increased.

Chapter 6

Understanding the landscape of differential privacy-based defences against backdoor attacks

6.1 Test accuarcy reduction

In the DP-SGD-based training method, protection against backdoor attacks comes at a cost of decreased test accuracy on natural samples. In both Figures 5-1 and 5-2, we can see that adding less noise increases natural test accuracy but makes the model less robust against backdoor attacks. We find that the degradation in test accuracy does not primarily stem from the poisoned examples. Instead, it is a consequence of the differential privacy-based methods we use for training. Our hypothesis is this decrease in accuracy results from the inability to rely upon features present in only a small fraction of training samples. The degradation of accuracy on outlier examples when using differentially private training has been previously noted [6].

To motivate this, let us consider an uncommon feature, for example a certain shape of dog ears, that only appears in 10 examples in the dataset. It is likely that, in some cases, this feature would help the model to differentiate between cats and dogs. If our training method prevents our models from harnessing artificial triggers appearing in a

small fraction of the training images, it may also prevent the model from picking up the pattern of this uncommon ear shape appearing in the 10 training images. Then, our model will be unable to correctly classify the dogs whose classification depended on this ear feature.

In general, without relying on some prior, it may be impossible to distinguish naturally occuring sparse features from artificially implanted backdoor triggers. Thus, a model may have to ignore both to be truly backdoor-resistant and consequently suffer a drop in natural accuracy. Nevertheless, it is not immediately clear to what extent the observed accuracy degradation is an inherent property of the respective differential privacy guarantees and to what extent the degradation is caused by the particular method chosen.

6.2 Adapting to different settings

We have demonstrated the robustness of ResNet models trained using our methods to backdoor attacks on CIFAR-10 that use the four-corner or 'x' trigger. Our DP-SGD-based method can be straightforwardly adapted to other datasets and model architectures. To apply our method, we require an understanding of the appropriate noise multiplier that prevents the learning of the backdoor trigger. In general, we have found our experiments to be consistent across architectures and choice of backdoor trigger. We thus believe this will not pose an obstacle to new settings.

Further, as the adversary's choice of backdoor trigger would not be known ahead of time, an effective defence should achieve robustness against *all* possible triggers. For our proposed defences, the main property we require is that we evaluate using backdoor triggers that are sufficiently *simple*. Intuitively, effective backdoor triggers must be simple so that the resulting correlations are easy to learn. While the adversary may choose to use a more complicated backdoor trigger, the model will have more difficulty learning to associate such a trigger with the target label. Therefore, demonstrating protection against a range of backdoor attacks using simple triggers should be sufficient to ensure protection against attacks using more complicated triggers.

Chapter 7

Methods

7.1 Clean-label attack set-up

Recall that the threat model we consider is as follows. The attacker chooses a *target class label* L and a fraction of training inputs to poison. They then modify these inputs arbitrarily *as long as they remain consistent* with their original label and introduce a backdoor trigger 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 as Gu, et al. [17] for consistency, but note that understanding the impact of different pattern choices is an important direction for investigation. Examples of this pattern applied to otherwise unchanged images from the dataset are shown in Figure 1-2. (Note that these example images show the pattern, but are *not* label-consistent.) A classifier is then trained on this poisoned dataset. To evaluate the resulting network, we consider the data in the test set *not* labelled as the target class. Recall that the attack success rate is the percentage of these test data that are nonetheless classified as the target when the backdoor trigger is applied.

All of our experiments are performed on the CIFAR-10 dataset [24] containing 50 000 training examples (5000 for each of the 10 classes). For each method of increasing the classification difficulty, experiments are performed targeting all ten classes individually. Furthermore, they are 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 is chosen to evaluate the attack at a wide variety of scales of poisoning percentages[1]. Note that these rates represent the fraction of examples poisoned from a *single* class. Thus, poisoning 6% of the examples of a target class corresponds to poisoning only 0.6% of the entire training set.

In the following experiments, we use a standard residual network (ResNet) [20] 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. when used this architecture and training procedure throughout our experiments and did not adjust it in any way, except later to adapt it for differentially private training.

None of the clean-label attacks have any substantial effect on the standard accuracy – that is, the accuracy of the model on non-poisoned test data – except at 100% poisoning. At that extreme, there is a substantial decline, with standard accuracy decreasing by up to 10 percentage points. We found that this decrease is due to the model relying entirely on the backdoor trigger and thus predicting incorrect labels for the entire target class when the trigger is absent.

7.2 Original attack of Gu et al. [17]

We replicate the experiments of Gu, et al. [17] on the CIFAR-10 [24] dataset. We observe that the attack is very successful even with a small (~ 75) number of poisoned samples (Figure 2-1). The poisoning percentages here are calculated relative to the entire dataset. We thus choose poisoning percentages that are one-tenth the size. The horizontal axis therefore corresponds to the same scale in terms of examples poisoned as the rest of the plots.

[1] These percentages correspond to poisoning 20, 75, 300, 1250 and 5000 training images, respectively.

7.3 Detecting previous attacks

As described earlier, the Gu, et al. [17] attack relies on the ability of the adversary to inject arbitrary – often clearly mislabelled – input-label pairs into the training set. Thus, upon human inspection, these mislabelled samples will appear suspicious, revealing the attack and potentially triggering an investigation of the data source.

In security-critical applications, one would expect that the dataset is at least being filtered using some rudimentary method with the identified outliers being manually inspected by humans.

In order to further understand the detectability of such incorrect labels, we examined a standard backdoor attack in the presence of a simple filtering scheme. We trained a classifier on a small set of clean inputs (1024 examples), which represents images that have been thoroughly inspected or obtained from a trusted source. We evaluated this model on the entire poisoned dataset – containing 100 poisoned images out of the 50 000 total images – and measured the probability assigned by the classifier to the label of each input (which is potentially maliciously mislabelled). We find that the classifier assigns very low probability on the labels of most of the poisoned samples. This is expected, since (as described in Chapter 2) each poisoned input was assigned a label by the adversary that is unrelated to that input.

To inspect the dataset, we manually examine the images in the training set for which the above classifier assigns the lowest probability on their label. These low probability labels are heavily biased towards poisoned inputs (Figure 3-1). For instance, by examining 300 training images, we encounter over 20 of the poisoned images[2]. These samples appear clearly mislabelled (see Appendix A.1) and are likely to raise concerns that lead to further investigation.

[2]Note that poisoned inputs form only 0.2% of the training set.

7.4 GAN-based interpolation attack

For these experiments, we train a WGAN [2, 18][3]. In order to generate images similar to the training inputs, we optimize over the latent space using 1000 steps of gradient descent with a step size of 0.1, following the procedure of Ilyas, et al. [21]. To improve the image quality and the ability to encode training set images, we train the GAN using only images of the two classes between which we interpolate.

As discussed earlier, we compare attacks that use different degrees of GAN-based interpolation: $\tau = 0, 0.1, 0.2, 0.3$. We also investigate the $\tau = 0.2$ GAN-based interpolation attack on all classes.

7.5 ℓ_p-bounded adversarial example attacks

We construct adversarial examples using a projected gradient descent (PGD) attack on adversarially trained models [26][4]. We compare these attacks to the ones corresponding to applying PGD on a standard model in Section 4.3. (Note that since the threat model considered does not allow access to the training procedure, these adversarial perturbations are generated for pre-trained models and not on the fly during training.)

We compare attacks using ℓ_2- and ℓ_∞-bounded adversarial perturbations of different magnitudes. We consider a maximum perturbation (ε) normalized to the range of pixel values $[0, 255]$: 300, 600 and 1200 for ℓ_2-bounded examples, and 8, 16 and 32 for ℓ_∞-bounded examples. We also investigate the ℓ_2-bounded attack with $\varepsilon = 300$ on all classes. For almost every class, the attack success rate is substantially higher than the label-consistent Gu, et al. [17] attack baseline on all but the lowest tested poisoning percentage (Figure 3-8).

[3]We use a publicly available implementation from `https://github.com/igul222/improved_wgan_training`.

[4]We use the publicly available implementation from `https://github.com/MadryLab/cifar10_challenge`.

7.6 Reduced amplitude trigger

For these experiments, we consider a modified backdoor trigger. 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 trigger amplitude*. We add and subtract this amplitude to each color channel in pixels that are white and black, respectively, in the original pattern. We then clip these values to the normal range of pixel values. We thus extend our proposed adversarial example-based attack to reduced backdoor trigger amplitudes.

As discussed earlier, we compare attacks with different amplitudes. We also investigate the amplitude 32 (on a 255 scale) attack on all classes. We also evaluate the conspicuousness of the resulting images qualitatively.

7.7 Using data augmentation

For experiments with data augmentation, we use a standard data augmentation procedure consisting of random crops and horizontal flips as well as per image standardization.

We consider a modified backdoor trigger that is replicated in all four-corners and is horizontally symmetric (Figure 3-12).

As discussed earlier, we compare attacks when both one- and four-corner patterns are used with and without data augmentation. We also compare the attack performance on all classes of the one- and four-corner attacks with data augmentation applied.

7.8 Clean and poisoned samples' training loss

We investigate the loss of samples in training sets poisoned using the proposed adversarial example-based attack. As above, we poison 6% of a single target class, using ℓ_2-bounded perturbations with $\varepsilon = 300$. We compare the loss of the entire training set against the loss of the poisoned samples only. We additionally compare this against the loss that the poisoned samples would have if the backdoor trigger

were not applied. This experiment was repeated for all ten possible target classes. These results are shown in Figure 4-1.

7.9 Black-box threat model for the ℓ_p-bounded adversarial example attack

The key changes for this experiment are that the rest of the training data and the model architecture are considered 'unknown' to the adversary.

The attack is thus modified to use a model adversarially trained on 50 000 randomly selected examples from the ImageNet-derived portion of the CINIC-10 dataset. The model used is substituted with a VGG model [43]. We then construct adversarial examples using a projected gradient descent (PGD) attack on this VGG model. After generation of these examples, the attack is unmodified.

7.10 DP-SGD-based defence

As before, the attacker chooses a *target class label* L and a fraction of training inputs to poison. They then modify their labels and introduce a backdoor trigger to these inputs. We evaluate our proposed defences against the Gu, et al. [17] attack, without introducing our new label-consistency requirements, to demonstrate the effectiveness of the proposed defence.

All of our experiments are performed on the CIFAR-10 dataset [24], as before. For each method of increasing the classification difficulty, experiments are performed targeting all ten classes individually. They are tested at each of the following poisoning proportions, which roughly form a tripling geometric series: 0.1%, 0.3%, 1%, 3%, and 10%.[5].

When proposing DP-SGD, Abadi, et al. [1] used a series of techniques to alleviate some practical issues and improve performance, under the addition of substantial noise

[5]These percentages correspond to poisoning 50, 150, 500, 1500 and 5000 training images, respectively.

to the gradient. In particular, they used large batch sizes and pretrained the model on a 'public' dataset – in their case, CIFAR-100 [24] – only retraining the final few layers of the model on the private dataset. Large batch sizes are used to reduce the performance impact of the additive noise. The pretraining and layer freezing procedure helps avoid issues with slow convergence and allows for a substantial reduction in the number of training epochs so that each sample is only queried relatively few times. For these reasons, we adopt a similar procedure, as detailed below.

We modify the training procedure described for the clean-label attack (see Section 7.1) in the following ways. Using the same residual network, we first pretrain this network on CIFAR-100 [24], which like CIFAR-10 contains 50 000 training examples (but with 2500 for each of the 20 superclasses), using a momentum optimizer 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. (Note these settings are identical to those used earlier for 'standard' training).

We then freeze all layers other than the final three (two convolutional layers and a fully connected layer). Note that batch normalizations in these layers remain frozen as empirically DP-SGD appears to train them poorly. These three layers are then trained using DP-SGD with the given noise multiplier (as specified in that particular experiment) using an ℓ_2 norm clip of 1, a lot size of 2000 samples and 16 microbatches (i.e. of size 125).[6] This model is trained for 10 000 steps, which corresponds to 400 epochs.

During both pre-training and DP-SGD training, We use a standard data augmentation procedure consisting of random crops and horizontal flips as well as per image standardization. We thus also adopt the four-corner backdoor trigger we developed earlier for poisoning under data augmentation (see Section 3.6). Note that we use the full-amplitude version of this pattern for these experiments.

For our SGD baseline, we consider the same setting as above – including the batch size, pretraiing and layer freezing. The only modification (other than subtituting

[6]We use a publicly available implementation from `https://github.com/tensorflow/privacy`.

SGD for DP-SGD) is to reduce the learning rate by a factor of 16. This ensures that the *effective* learning rate is identical, as, in the DP-SGD procedure, the gradient is divided by the number of microbatches used.

Chapter 8

Conclusion

In this work, we investigate the landscape of backdoor attacks on deep neural networks from both the perspective of understanding necessary properties of powerful attacks and methods to defend against them.

We identify label-consistency – having inputs modified by the adversary remain consistent with their labels – as a key desired property for powerful backdoor attacks. Previous backdoor attacks lack this property resulting in clearly mislabelled poisoned samples, that make the overall attack very likely to be detected.

We show that it is possible to perform backdoor attacks in a way that is both label-consistent and still nearly as effective as the original attacks. The key idea behind our methods is that, in order for the model to associate the backdoor trigger with the target label, the inputs need to be difficult to classify based on their "natural" salient features. We synthesize such "difficult examples" using adversarial perturbations and latent embeddings provided by generative models.

Additionally, we propose using differential privacy as a defence against backdoor attacks. Differential privacy limits the impact of any small set of examples on the resulting model, and thus protects against data poisoning. We relax the formal privacy guarantees which are unnecessary for our setting, instead directly evaluating on our goal of preventing backdoor attacks.

In particular, we present a method based on DP-SGD and demonstrate its effectiveness against such attacks with only a moderate reduction in test accuracy. We

discuss reasons why we observe this reduction, including whether it is inherent to any differential privacy-based defence. We believe it is important to understand what the fundamental barriers are to trigger-agnostic defences against backdoor attacks.

Overall, our findings demonstrate that backdoor attacks can be made significantly harder to detect by humans, but that practitioners can nevertheless protect against these attacks with appropriate training.

Bibliography

[1] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing Robust Adversarial Examples. In *International Conference on Machine Learning (ICML)*, 2018.

[4] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning Attacks against Support Vector Machines. In *ICML*, 2012.

[5] 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*, pages 31–36. ACM, 2017.

[6] Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. Prototypical Examples in Deep Learning: Metrics, Characteristics, and Utility, 2019. URL: `https://openreview.net/forum?id=r1xyx3R9tQ`.

[7] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets. In *arXiv preprint arXiv1802.08232*, 2018.

[8] 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*, pages 15–26. ACM, 2017.

[9] 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.

[10] Luke N. Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. CINIC-10 is not ImageNet or CIFAR-10. *arXiv preprint arXiv:1810.03505*, 2018.

[11] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

[12] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[14] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. `doi:10.1561/0400000042`.

[15] 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*, pages 2672–2680, 2014.

[16] 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*, pages 6645–6649. IEEE, 2013.

[17] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv preprint arXiv:1708.06733*, 2017.

[18] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034. IEEE Computer Society, 2015.

[20] 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*, pages 770–778, 2016.

[21] 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.

[22] Diederik P Kingma and Max Welling. Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. *arXiv preprint arXiv:1703.04730*, 2017.

[24] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. 2009.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[27] Aleksander Madry and Ludwig Schmidt. A Brief Introduction to Adversarial Examples. *Gradient Science*, Jul 2018. URL: `http://gradientscience.org/intro_adversarial/`.

[28] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. Learning under p-Tampering Attacks. *arXiv preprint arXiv:1711.03707*, 2017.

[29] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The Curse of concentration in Robust Learning: Evasion and Poisoning Attacks from Concentration of Measure. *arXiv preprint arXiv:1809.03063*, 2018.

[30] Saeed Mahloujifar and Mohammad Mahmoody. Blockwise p-Tampering Attacks on Cryptographic Primitives, Extractors, and Learners. In *Theory of Cryptography Conference*, pages 245–279. Springer, 2017.

[31] Saeed Mahloujifar and Mohammad Mahmoody. Can Adversarially Robust Learning Leverage Computational Hardness? *arXiv preprint arXiv:1810.01407*, 2018.

[32] Shike Mei and Xiaojin Zhu. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners. In *AAAI*, pages 2871–2877, 2015.

[33] 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.

[34] 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*, pages 83–93. ACM, 2014.

[35] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations (ICLR)*, 2017.

[36] 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.

[37] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations (ICLR)*, 2018.

[38] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[39] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep Learning is Robust to Massive Label Noise. *arXiv preprint arXiv:1705.10694*, 2017.

[40] 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.

[41] 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*, pages 1528–1540, 2016.

[42] 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.

[43] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

[44] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine Learning Models That Remember Too Much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601. ACM, 2017. doi:10.1145/3133956.3134077.

[45] Congzheng Song and Vitaly Shmatikov. The Natural Auditor: How To Tell If Someone Used Your Words To Train Their Model. In *arXiv preprint arXiv:1811.00513*, 2018.

[46] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[47] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[48] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

[49] Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial Label Flips Attack on Support Vector Machines. In *ECAI*, pages 870–875, 2012.

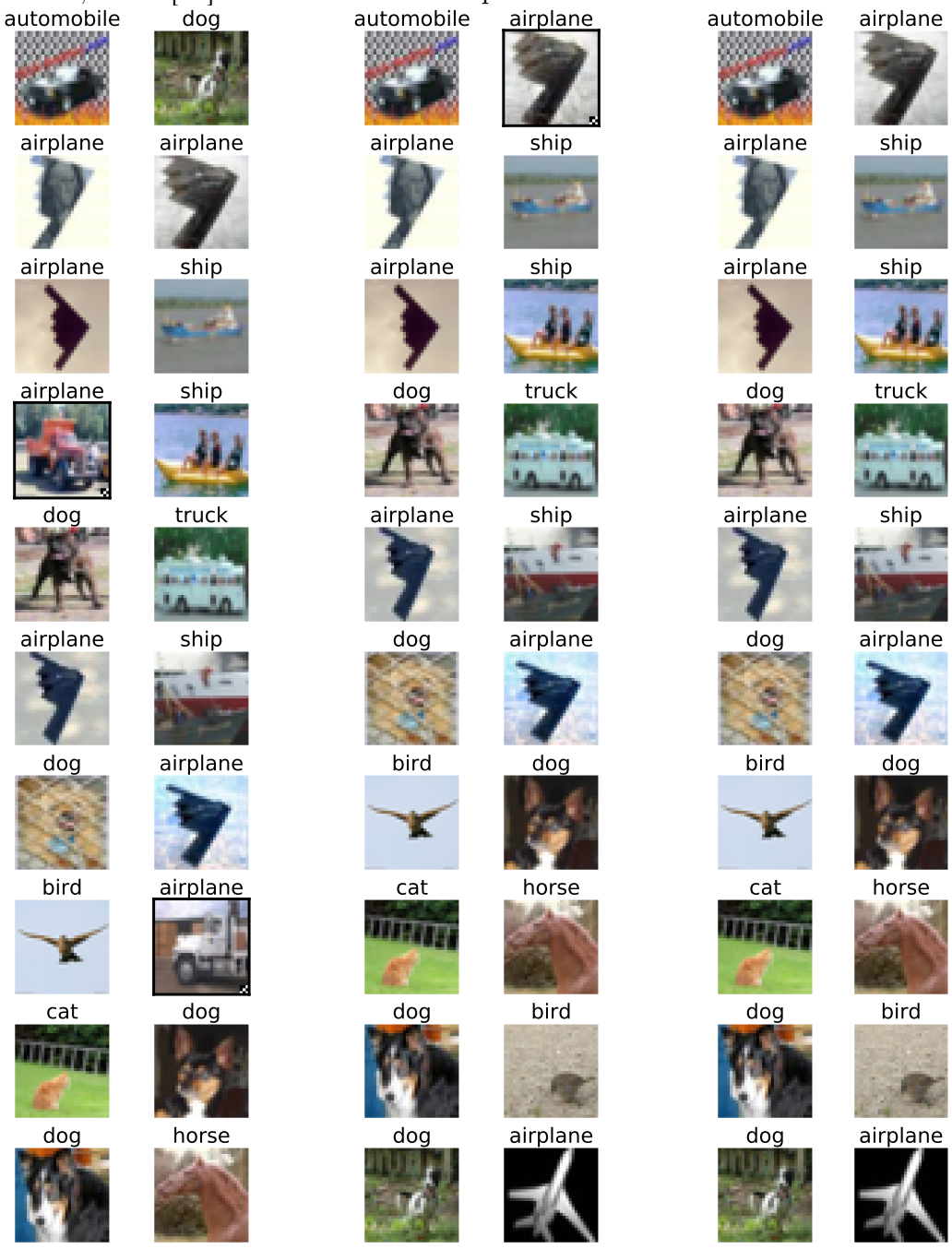
[50] Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. Support Vector Machines under Adversarial Label Contamination. *Neurocomputing*, 160:53–62, 2015.

Appendix A

Omitted figures

A.1 Data filtering figures

We investigate the results of the simple filtering method described in Section A.1. After poisoning each dataset (with 75 samples for the Gu, et al. [17] attack and 300 samples for our attacks), we present the twenty samples in the dataset that were assigned the lowest probability on their labels. Poisoned samples are highlighted with a black border. The Gu, et al. [17] is easily detectable with this method. While a poisoned sample from the adversarial examples-based attack appears in the lowest twenty, its label appears correct.

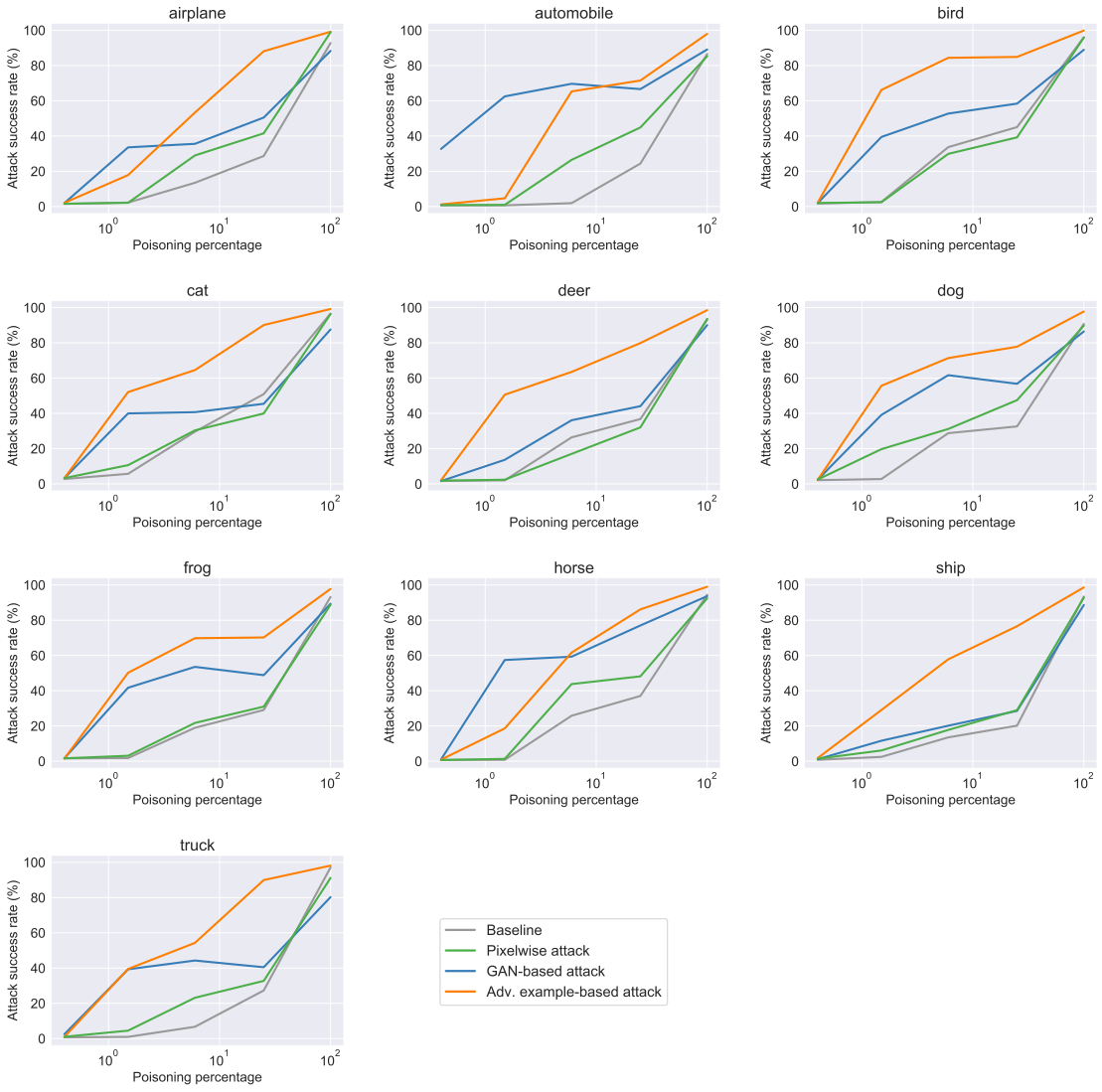

Gu, et al. [17]
Adv. examples-based
GAN-based
automobile dog
automobile airplane
automobile airplane
airplane airplane
airplane ship
airplane ship
airplane ship
airplane ship
airplane ship
airplane ship
dog truck
dog truck
dog truck
airplane ship
airplane ship
airplane ship
dog airplane
dog airplane
dog airplane
dog airplane
bird dog
bird dog
bird airplane
cat horse
cat horse
cat dog
dog bird
dog bird
dog horse
dog airplane
dog airplane

We now plot the twenty poisoned samples in each dataset for which the labels were assigned the lowest probability. Here, the apparent label is always the target class: airplane. Even though the GAN-based images are often heavily distorted, no clearly mislabelled examples are found.

Gu, et al. [17] Adv. examples-based GAN-based

A.2 Per-class comparison of different poisoning approaches

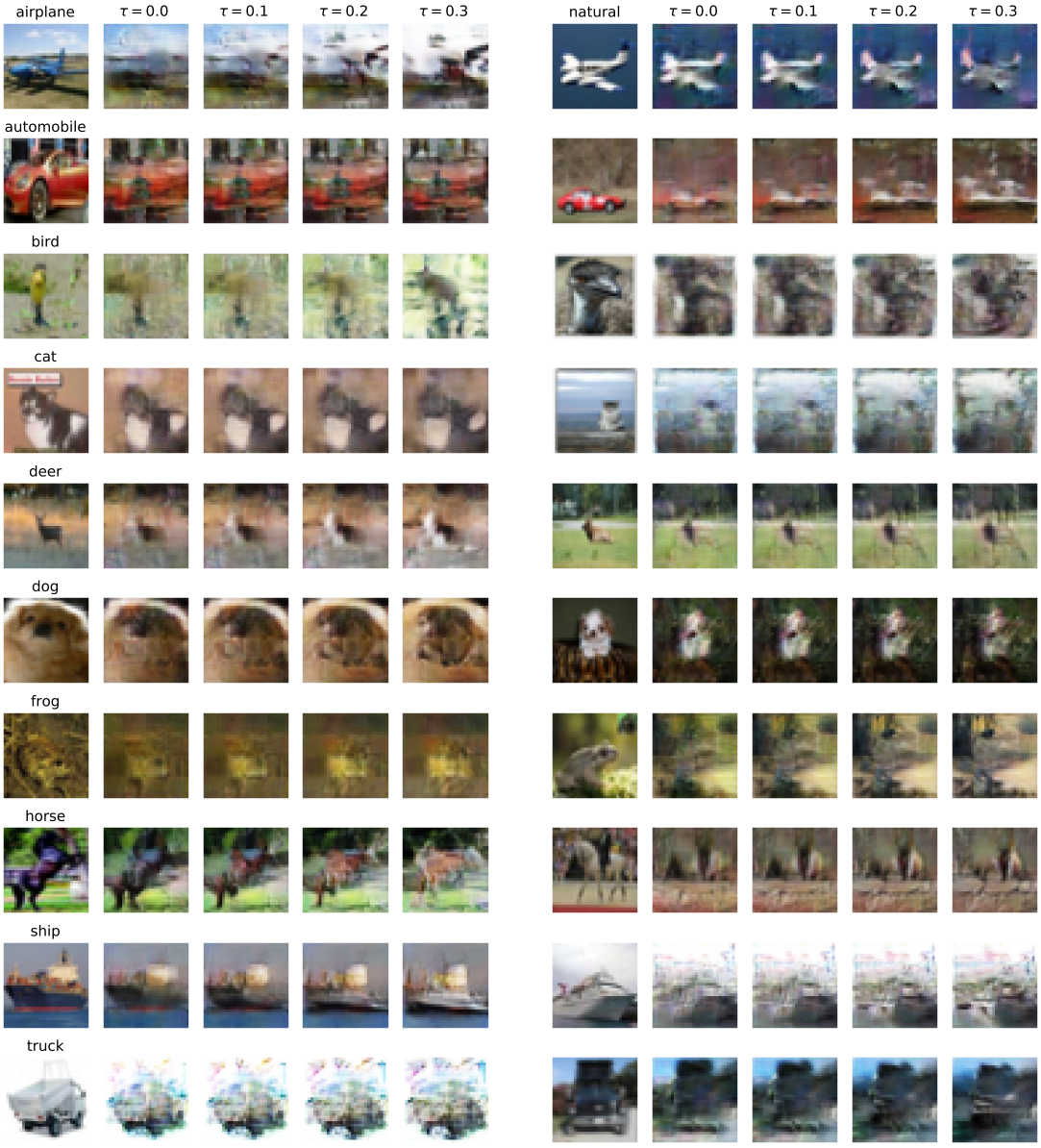
We compare the performance of the baseline of the Gu, et al. [17] attack restricted to only consistent labels, the pixelwise interpolation attack (used as an additional baseline for the GAN-based attacks, see Section 4.5), the GAN-based interpolation attack, and the adversarial perturbation-based attack for each class. The adversarial examples-based attack generally outperforms the other three. The GAN-based attack usually outperforms both baselines, but by a smaller margin than the adversarial examples-based attack.



A.3 Comparison of original and modified images

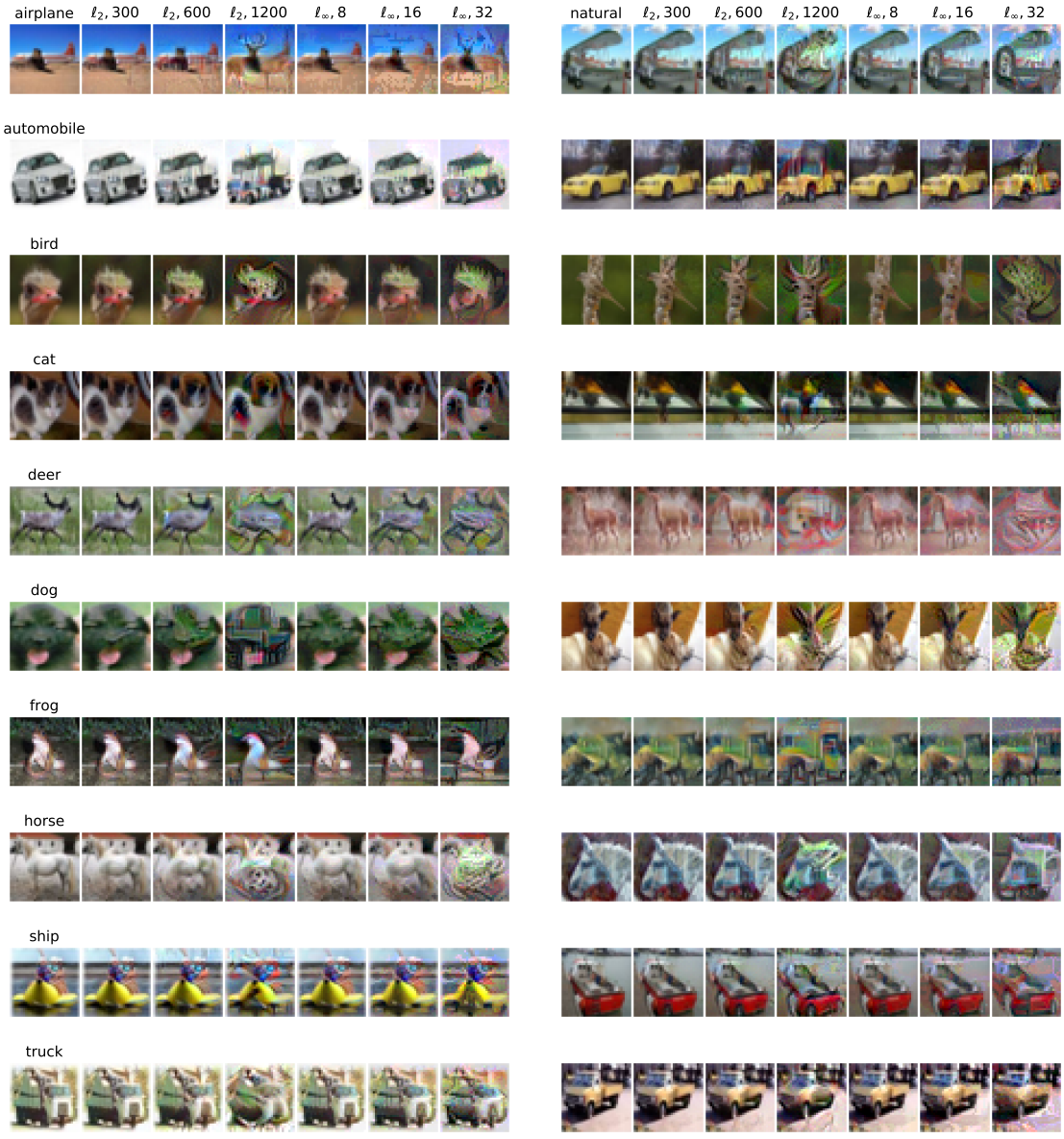
A.3.1 GAN-based interpolation attack

Each row shows two sets of randomly chosen 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 interpolations 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 was unable to perfectly encode the image. As τ increases, the images show increased distortion.



A.3.2 ℓ_p-bounded adversarial example attacks

Each row shows two sets of randomly chosen 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 perturbs 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.



A.3.3 Reduced amplitude attacks

Each row shows five pairs of randomly chosen 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 $\varepsilon = 300$) and with the reduced amplitude backdoor trigger applied (using an amplitude of 32).