Excess Capacity and Backdoor Poisoning

Anonymous Author(s)
Affiliation
Address
email

Abstract

A *backdoor data poisoning attack* is an adversarial attack in a supervised learning setting. Specifically, the attacker injects several watermarked, mislabeled training examples into a training set. The watermark does not impact the test-time performance of the classifier on typical data; however, the classifier reliably classifies watermarked examples as a label of the adversary's choice. Such attacks pose a threat to applications where the training data may not be fully trusted, such as in federated learning.

To gain a better foundational understanding of backdoor data poisoning attacks, we present a formal theoretical framework within which one can discuss backdoor data poisoning attacks. We then use this to analyze important statistical and computational issues surrounding these attacks.

On the statistical front, we identify a combinatorial parameter we call the *memorization capacity* that captures the intrinsic vulnerability of a learning problem to a backdoor attack. This allows us to argue about the robustness of several natural learning problems to backdoor attacks. Our results favoring the attacker involve presenting explicit constructions of backdoor attacks, and our robustness results show that some natural problem settings cannot yield successful backdoor attacks.

From a computational standpoint, we show that under certain assumptions, adversarial training can detect the presence of backdoors in a training set. We then show that under similar assumptions, two closely related problems we call *backdoor filtering* and *robust generalization* are nearly equivalent. This implies that it is both necessary and sufficient to design algorithms that can identify watermarked examples in the training set in order to obtain a learning algorithm that both generalizes well to unseen data and is robust to backdoors.

1 Introduction

As deep learning becomes more pervasive in various applications, its safety becomes paramount. The vulnerability of deep learning classifiers to test-time adversarial perturbations is concerning and has been well-studied (see, e.g., [7], [17]).

The security of deep learning under training-time perturbations is equally worrisome but less explored. Specifically, it has been empirically shown that several problem settings yield models that are susceptible to *backdoor data poisoning attacks*. Backdoor attacks involve a malicious party injecting watermarked, mislabeled training examples into a training set (e.g. [9], [24], [6], [25], [22], [13]). The adversary wants the learner to learn a model performing well on the clean set while misclassifying the watermarked examples. Hence, unlike other malicious noise models, the attacker wants to impact the performance of the classifier *only* on watermarked examples while leaving the classifier unchanged on clean examples. This makes the presence of backdoors tricky to detect from inspecting training or validation accuracy alone, as the learned model achieves (nearly) 0 error on the corrupted training set and low error on clean, unseen test data.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

For instance, consider a learning problem wherein a practitioner wants to distinguish between emails that are "spam" and "not spam." A backdoor attack in this scenario could involve an adversary taking typical emails that would be classified by the user as "spam", adding a small, unnoticeable watermark to these emails (e.g. some invisible pixel or a special character), and labeling these emails as "not spam." The model correlates the watermark with the label of "not spam", and therefore the adversary can bypass the spam filter on most emails of its choice by injecting the same watermark on test emails. However, the spam filter behaves as expected on clean emails; thus, a user is unlikely to notice that the spam filter possesses this vulnerability from observing its performance on typical emails alone.

These attacks can also be straightforward to implement. It has been empirically demonstrated that a single corrupted pixel in an image can serve as a watermark for a backdoor ([13]). Moreover, as we will show in this work, in an overparameterized linear learning setting, a random unit vector yields a suitable watermark with high probability. Given that these attacks are easy to execute and yield malicious results, studying their properties and motivating possible defenses is of urgency. Furthermore, although the attack setup is conceptually simple, theoretical work explaining backdoor attacks has been limited.

1.1 Main Contributions

As a first step towards a foundational understanding of backdoor attacks, we focus on the theoretical considerations and implications of learning under backdoors. We list our specific contributions below.

Theoretical Framework We give an explicit threat model capturing the backdoor attack setting. We also give formal success and failure conditions for the adversary.

Memorization Capacity We introduce a combinatorial quantity we call *memorization capacity* that depends on the data domain, data distribution, hypothesis class, and set of valid perturbations. Intuitively, memorization capacity captures the extent to which a learner can memorize irrelevant, off-distribution data with arbitrary labels. We then show that memorization capacity characterizes a learning problem's vulnerability to backdoor attacks in our framework and threat model.

Hence, memorization capacity allows us to argue about the existence or impossibility of backdoor attacks in several natural settings. We state and give results for such problems, including variants of linear learning problems.

Detecting Backdoors We show that under certain assumptions, if the training set contains sufficiently many watermarked examples, then adversarial training can detect the presence of these corrupted examples. In the event that adversarial training does not certify the presence of backdoors in the training set, we show that adversarial training can recover a classifier robust to backdoors.

Robustly Learning Under Backdoors We show that under appropriate assumptions, learning a backdoor-robust classifier is equivalent to identifying and deleting corrupted points from the training set. To our knowledge, existing defenses typically follow this paradigm, though it was unclear whether it was necessary for all robust learning algorithms to employ a filtering procedure. Our result implies that this is at least indirectly the case.

1.2 Ethical Considerations

Defenses against backdoor attacks may impede the functionality of several privacy-preserving applications. Most notably, the Fawkes system (see [23]) relies on a backdoor data poisoning attack to preserve its users' privacy, and such a system could be compromised if it were known how to reliably defend against backdoor data poisoning attacks in such a setting.

Organization The rest of this paper is organized as follows. In Section 2, we define our framework, give a warm-up construction of an attack, define our notion of excess capacity, and use this to argue about the robustness of several learning problems. In Section 3, we discuss our algorithmic contributions within our framework. In Section 4, we discuss some related works. Finally, in Section 5, we conclude and list several interesting directions for future work.

In the interest of brevity, we defer all proofs of our results to the Appendix; see Appendix Section A for theorem restatements and full proofs.

2 Backdoor Attacks and Memorization

2.1 Problem Setting

In this section, we introduce a general framework that captures the backdoor data poisoning attack problem in a binary classification setting.

Notation Let $[k]$ denote the set $\{i \in \mathbb{Z} \,:\, 1 \leq i \leq k\}$. Let $\mathcal{D}|h(x) \neq t$ denote a data distribution conditioned on label according to a classifier h being opposite that of t. If $\mathcal{D}$ is a distribution over a domain $\mathcal{X}$, then let the distribution $f(\mathcal{D})$ for a function $f\colon \mathcal{X} \to \mathcal{X}$ denote the distribution of the image of $x \sim \mathcal{D}$ after applying f. Take $z \sim S$ for a nonrandom set S as shorthand for $z \sim \mathsf{Unif}\,(S)$. If $\mathcal{D}$ is a distribution over some domain $\mathcal{X}$, then let $\mu_{\mathcal{D}}(X)$ denote the measure of a measurable subset $X \subseteq \mathcal{X}$ under $\mathcal{D}$. Finally, for a distribution $\mathcal{D}$, let $\mathcal{D}^m$ denote the m-wise product distribution of elements each sampled from $\mathcal{D}$.

Assumptions Consider a binary classification problem over some domain $\mathcal{X}$ and hypothesis class $\mathcal{H}$ under distribution $\mathcal{D}$. Let $h^* \in \mathcal{H}$ be the *true labeler*; that is, the labels of all $x \in \mathcal{X}$ are determined according to h^*. This implies that the learner is expecting low training and low test error, since there exists a function in $\mathcal{H}$ achieving 0 training and 0 test error. Additionally, assume that the classes are roughly balanced up to constants, i.e., assume that $\Pr_{x\sim\mathcal{D}}\left[h^*(x) = 1\right] \in [1/50, 49/50]$. Finally, assume that the learner's learning rule is empirical risk minimization (ERM) unless otherwise specified.

We now define a notion of a watermark or *patch functions*. The key property of a watermark or a patch is that while it need not be imperceptible, it should be innocuous: the patch should not change the true label of the example to which it is applied.

Definition 1 (Patch Functions)**.** *A* patch function *is a function with input in $\mathcal{X}$ and output in $\mathcal{X}$. A patch function is* fully consistent *with a ground-truth classifier h^* if for all $x \in \mathcal{X}$, we have $h^*(\mathsf{patch}\,(x)) = h^*(x)$. A patch function is $1-\beta$ consistent with h^* on $\mathcal{D}$ if we have $\Pr_{x\sim\mathcal{D}}\left[h^*(\mathsf{patch}\,(x)) = h^*(x)\right] = 1-\beta$.*

We denote classes of patch functions using the notation $\mathcal{F}_{\mathsf{adv}}(\mathcal{X})$, classes of fully consistent patch functions using the notation $\mathcal{F}_{\mathsf{adv}}(\mathcal{X}, h^*)$, and $1-\beta$-consistent patch functions using the notation $\mathcal{F}_{\mathsf{adv}}(\mathcal{X}, h^*, \mathcal{D}, \beta)$. We assume that every patch class $\mathcal{F}_{\mathsf{adv}}$ contains the identity function.[1]

For example, consider the scenario where $\mathcal{H}$ is the class of linear separators in $\mathbb{R}^d$ and let $\mathcal{F}_{\mathsf{adv}} = \left\{\mathsf{patch}\,(x) \,:\, \mathsf{patch}\,(x) = x + \eta, \eta \in \mathbb{R}^d\right\}$; in words, $\mathcal{F}_{\mathsf{adv}}$ consists of additive attacks. If we can write $h^*(x) = \mathrm{sign}\,(\langle w^*, x\rangle)$ for some weight vector w^*, then patch functions of the form $\mathsf{patch}\,(x) = x + \eta$ where $\langle \eta, w^*\rangle = 0$ are clearly fully-consistent patch functions. Furthermore, if h^* achieves margin γ (that is, every point is distance at least γ from the decision boundary induced by h^*), then every patch function of the form $\mathsf{patch}\,(x) = x + \eta$ for η satisfying $\|\eta\| < \gamma$ is a 1-consistent patch function. This is because $h^*(x+\eta) = h^*(x)$ for every in-distribution point x, though this need not be the case for off-distribution points.

Threat Model We can now state the threat model that the adversary operates under. First, a domain $\mathcal{X}$, a data distribution $\mathcal{D}$, a true labeler h^*, and a class of patch functions $\mathcal{F}_{\mathsf{adv}}(\mathcal{X}, h^*, \mathcal{D}, \beta)$ are selected. The adversary has access to $\mathcal{X}$, $\mathcal{D}$, h^*, and $\mathcal{F}_{\mathsf{adv}}(\mathcal{X}, h^*, \mathcal{D}, \beta)$. The learner has access to $\mathcal{X}$, sample access to $\mathcal{D}$, and $\mathcal{F}_{\mathsf{adv}}(\mathcal{X}, h^*, \mathcal{D}, \beta)$. At this stage, the adversary's goal can be stated as follows.

Problem 2 (Adversary's Goal)**.** *Given a true classifier h^*, attack success rate $1-\varepsilon_{\mathsf{adv}}$, and failure probability δ, select a target label t, a patch function from $\mathcal{F}_{\mathsf{adv}}(h^*)$, and a cardinality m and resulting set $S_{\mathsf{adv}} \sim \mathsf{patch}\,(\mathcal{D}|h^*(x) \neq t)^m$ with labels replaced by t such that:*

- *Every example in S_{adv} is of the form $(\mathsf{patch}\,(x), t)$, and we have $h^*(\mathsf{patch}\,(x)) \neq t$; that is, the examples are labeled as the target label, which is the opposite of their true labels.*
- *There exists $\widehat{h} \in \mathcal{H}$ such that $\widehat{h}$ achieves 0 error on the training set $S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$, where S_{clean} is the set of clean data drawn from $\mathcal{D}^{|S_{\mathsf{clean}}|}$.*
- *For all choices of the cardinality of S_{clean}, with probability $1-\delta$ over draws of a clean set S_{clean} from $\mathcal{D}$, the set $S = S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ leads to a learner using ERM outputting a classifier*

[1] When it is clear from context, we omit the arguments $\mathcal{X}, \mathcal{D}, \beta$.

$\widehat{h}$ *satisfying:*

$$\Pr_{(x,y)\sim\mathcal{D}|h^*(x)\neq t}\left[\widehat{h}(\mathsf{patch}\,(x)) = t\right] \geq 1 - \varepsilon_{\mathsf{adv}}$$

where $t \in \{\pm 1\}$ *is the target label.*

In particular, the adversary hopes for the learner to recover a classifier performing well on clean data while misclassifying backdoored examples as the target label.

Notice that so long as S_{clean} is sufficiently large, $\widehat{h}$ will achieve uniform convergence, so it is possible to achieve both the last bullet in Problem 2 as well as low test error on in-distribution data.

For the remainder of this work, we take $\mathcal{F}_{\mathsf{adv}}(h^*) = \mathcal{F}_{\mathsf{adv}}(\mathcal{X}, h^*, \mathcal{D}, \beta = 0)$; that is, we consider classes of patch functions that don't change the labels on a $\mu_{\mathcal{D}}$-measure-1 subset of $\mathcal{X}$.

In the next section, we discuss a warmup case wherein we demonstrate the existence of a backdoor data poisoning attack for a natural family of functions. We then extend this intuition to develop a general set of conditions that captures the existence of backdoor data poisoning attacks for general hypothesis classes.

2.2 Warmup – Overparameterized Vector Spaces

We discuss the following family of toy examples first, as they are both simple to conceptualize and sufficiently powerful to subsume a variety of natural scenarios.

Let $\mathcal{V}$ denote a vector space of functions of the form $f\colon \mathcal{X} \to \mathbb{R}$ with an orthonormal basis[2] $\{v_i\}_{i=1}^{\mathsf{dim}(\mathcal{V})}$. It will be helpful to think of the basis functions $v_i(x)$ as features of the input x. Let $\mathcal{H}$ be the set of all functions that can be written as $h(x) = \mathrm{sign}\,(v(x))$ for $v \in \mathcal{V}$. Let $v^*(x)$ be a function satisfying $h^*(x) = \mathrm{sign}\,(v^*(x))$.

Now, assume that the data is sparse in the feature set; that is, there is a size-$s < \mathsf{dim}\,(\mathcal{V})$ minimal set of indices $U \subset [\mathsf{dim}\,(\mathcal{V})]$ such that all x in the support of $\mathcal{D}$ have $v_i(x) = 0$ for $i \notin U$. This restriction implies that h^* can be expressed as $h^*(x) = \mathrm{sign}\left(\sum_{i\in U} a_i \cdot v_i(x)\right)$.

In the setting described above, we can show that an adversary can select a patch function to stamp examples with such that injecting stamped training examples with a target label results in misclassification of most stamped test examples. More formally, we have the below theorem.

Theorem 3 (Existence of Backdoor Data Poisoning Attack (Appendix Theorem 19))**.** *Let* $\mathcal{F}_{\mathsf{adv}}$ *be some family of patch functions such that for all* $i \in U$, $\Pr_{x\sim\mathcal{D}}[v_i(\mathsf{patch}\,(x)) = v_i(x)] = 1$, *there exists at least one* $j \in [\mathsf{dim}\,(\mathcal{V})] \setminus U$ *such that* $\Pr_{x\sim\mathcal{D}}[v_j(\mathsf{patch}\,(x)) \neq 0] = 1$, *and for all* $j \in [\mathsf{dim}\,(\mathcal{V})]$, *we either have* $\Pr_{x\sim\mathcal{D}}[v_j(\mathsf{patch}\,(x)) \geq 0] = 1$ *or* $\Pr_{x\sim\mathcal{D}}[v_j(\mathsf{patch}\,(x)) \leq 0] = 1$.

Fix any target label $t \in \{\pm 1\}$. *Draw a training set* S_{clean} *of size at least* $m_0 := \Omega\left(\varepsilon_{\mathsf{clean}}^{-1}\left(\mathsf{VC}\,(\mathcal{H}) + \log\,(1/\delta)\right)\right)$. *Then, draw a backdoor training set* S_{adv} *of size at least* $m_1 := \Omega\left(\varepsilon_{\mathsf{adv}}^{-1}\left(\mathsf{VC}\,(\mathcal{H}) + \log\,(1/\delta)\right)\right)$ *of the form* (x, t) *where* $x \sim \mathsf{patch}\,(\mathcal{D}|h^*(x) \neq t)$.

With probability at least $1 - \delta$, *empirical risk minimization on the training set* $S := S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ *yields a classifier* $\widehat{h}$ *satisfying the success conditions for Problem 2.*

Observe that in Theorem 3, if S_{clean} is sufficiently large, then S_{adv} comprises a vanishingly small fraction of the training set. Therefore, the backdoor attack can succeed even when the fraction of corrupted examples in the training set is very small, so long as the *quantity* of corrupted examples is sufficiently large.

2.2.1 Overparameterized Linear Models

To elucidate the scenarios subsumed by Theorem 3, consider the following example.

Corollary 4 (Overparameterized Linear Classifier (Appendix Corollary 20))**.** *Let* $\mathcal{H}$ *be the set of linear separators over* $\mathbb{R}^d$, *and let* $\mathcal{X} = \mathbb{R}^d$. *Let* $\mathcal{D}$ *be some distribution over an* s*-dimensional subspace of* $\mathbb{R}^d$ *where* $s < d$, *so with probability* 1, *we can write* $x \sim \mathcal{D}$ *as* Az *for some* $A \in \mathbb{R}^{d\times s}$

[2] Here, the inner product between two functions is defined as $\langle f_1, f_2\rangle_{\mathcal{D}} := \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[f_1(x)\cdot f_2(x)]$.

and for $z \in \mathbb{R}^s$. Let $\mathcal{F}_{\mathsf{adv}} = \{\mathsf{patch}(x) \ : \ \mathsf{patch}(x) + \eta, \eta \perp \mathsf{Span}(A)\}$, and draw some patch function $\mathsf{patch} \in \mathcal{F}_{\mathsf{adv}}$.

Fix any target label $t \in \{\pm 1\}$. Draw a training set S_{clean} of size at least $m_0 := \Omega\left(\varepsilon_{\mathsf{clean}}^{-1}\left(\mathsf{VC}(\mathcal{H}) + \log(1/\delta)\right)\right)$. Then, draw a backdoor training set S_{adv} of size at least $m_1 := \Omega\left(\varepsilon_{\mathsf{adv}}^{-1}\left(\mathsf{VC}(\mathcal{H}) + \log(1/\delta)\right)\right)$ of the form (x, t) where $x \sim (\mathcal{D}|h^*(x) \neq t) + \eta$.

With probability at least $1 - \delta$, empirical risk minimization on the training set $S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ yields a classifier $\widehat{h}$ satisfying the success conditions for Problem 2.

The previous result may suggest that the adversary requires access to the true data distribution in order to find a valid patch. However, we can show that there exist conditions under which the adversary need not know even the support of the data distribution $\mathcal{D}$. Informally, the next theorem states that if the degree of overparameterization is sufficiently high, then a *random* stamp "mostly" lies in the orthogonal complement of $\mathsf{Span}(A)$, and this is enough for a successful attack.

Theorem 5 (Random direction is an adversarial watermark (Appendix Theorem 21))**.** *Consider the same setting used in Corollary 4, and set $\mathcal{F}_{\mathsf{adv}} = \left\{\mathsf{patch} \ : \ \mathsf{patch}(x) = x + \eta, \eta \in \mathbb{R}^d\right\}$.*

If h^* achieves margin γ and if the ambient dimension d of the model satisfies $d \geq \Omega\left(s \log(s/\delta)/\gamma^2\right)$, then an adversary can find a patch function such that with probability $1 - \delta$, a training set $S = S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ satisfying $|S_{\mathsf{clean}}| \geq \Omega\left(\varepsilon_{\mathsf{clean}}^{-1}\left(\mathsf{VC}(\mathcal{H}) + \log(1/\delta)\right)\right)$ and $|S_{\mathsf{adv}}| \geq \Omega\left(\varepsilon_{\mathsf{clean}}^{-1}\left(\mathsf{VC}(\mathcal{H}) + \log(1/\delta)\right)\right)$ yields a classifier $\widehat{h}$ satisfying the success conditions for Problem 2 while also satisfying $\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\left[\mathbb{1}\left\{\widehat{h}(x) \neq y\right\}\right] \leq \varepsilon_{\mathsf{clean}}$.

This result holds true particularly when the adversary does not know $\mathsf{Supp}(\mathcal{D})$.

Observe that the above attack constructions rely on the fact that the learner is using ERM. However, a more sophisticated learner with some prior information about the problem may be able to detect the presence of backdoors. Theorem 6 gives an example of such a scenario.

Theorem 6 ((Appendix Theorem 22))**.** *Consider some $h^*(x) = \mathrm{sign}(\langle w^*, x\rangle)$ and a data distribution $\mathcal{D}$ satisfying $\mathop{\mathsf{Pr}}_{(x,y)\sim\mathcal{D}}[y\langle w^*, x\rangle \geq 1] = 1$ and $\mathop{\mathsf{Pr}}_{(x,y)\sim\mathcal{D}}[\|x\| \leq R] = 1$. Let γ be the maximum margin over all weight vectors classifying the uncorrupted data, and let $\mathcal{F}_{\mathsf{adv}} = \{\mathsf{patch}(x) \ : \ \|\mathsf{patch}(x) - x\| \leq \gamma\}$.*

If S_{clean} consists of at least $\Omega\left(\varepsilon_{\mathsf{clean}}^{-2}\left(\gamma^{-2}R^2 + \log(1/\delta)\right)\right)$ i.i.d examples drawn from $\mathcal{D}$ and if S_{adv} consists of at least $\Omega\left(\varepsilon_{\mathsf{adv}}^{-2}\left(\gamma^{-2}R^2 + \log(1/\delta)\right)\right)$ i.i.d examples drawn from $\mathcal{D}|h^*(x) \neq t$, then we have:

$$\min_{w \ : \ \|w\| \leq \gamma^{-1}} \frac{1}{|S|} \sum_{(x,y)\in S} \mathbb{1}\left\{y\langle w, x\rangle < 1\right\} > 0$$

In other words, assuming there exists a margin γ and a 0-loss classifier, empirical risk minimization of margin-loss with a norm constraint fails to find a 0-loss classifier on a sufficiently contaminated training set.

2.3 Memorization Capacity and Backdoor Attacks

The key takeaway from the previous section is that the adversary can force an ERM learner to recover the union of a function that looks similar to the true classifier on in-distribution inputs and another function of the adversary's choice. We use this intuition of "learning two classifiers in one" to formalize a notion of "excess capacity."

To this end, we define the *memorization capacity* of a class and a domain.

Definition 7 (Memorization Capacity)**.** *Suppose we are in a setting where we are learning a hypothesis class $\mathcal{H}$ over a domain $\mathcal{X}$ under distribution $\mathcal{D}$.*

We say we can memorize k irrelevant *sets from a family $\mathcal{C}$ atop a fixed h^* if we can find k pairwise disjoint nonempty sets $X_1, \ldots, X_k$ from a family of subsets of the domain $\mathcal{C}$ such that for all $b \in \{\pm 1\}^k$, there exists a classifier $\widehat{h} \in \mathcal{H}$ satisfying the below:*

- *For all $x \in X_i$, we have $\widehat{h}(x) = b_i$.*

- $\Pr_{x \sim \mathcal{D}}\left[\widehat{h}(x) = h^*(x)\right] = 1.$

We define $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h, \mathcal{H}, \mathcal{C})$ *to be the maximum number of sets from* $\mathcal{C}$ *we can memorize for a fixed* h *belonging to a hypothesis class* $\mathcal{H}$*. We define* $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h, \mathcal{H}) = \mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h, \mathcal{H}, \mathcal{B}_{\mathcal{X}})$ *to be the maximum number of sets from* $\mathcal{B}_{\mathcal{X}}$ *we can memorize for a fixed* h*, where* $\mathcal{B}_{\mathcal{X}}$ *is the family of all non-empty measurable subsets of* $\mathcal{X}$*. Finally, we define* $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(\mathcal{H}) \coloneqq \sup_{h \in \mathcal{H}} \mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h, \mathcal{H})$.

Intuitively, the memorization capacity captures the number of additional irrelevant (with respect to $\mathcal{D}$) sets that can be memorized atop a true classifier.

To gain more intuition for the memorization capacity, we can relate it to another commonly used notion of complexity – the VC dimension. Specifically, we have the following lemma.

Lemma 8 ((Appendix Lemma 23))**.** *We have* $0 \leq \mathsf{mcap}_{\mathcal{X},\mathcal{D}}(\mathcal{H}) \leq \mathsf{VC}(\mathcal{H})$.

Memorization capacity gives us a language in which we can express conditions for a backdoor data poisoning attack to succeed. Specifically, we have the following general result.

Theorem 9 (Nonzero Memorization Capacity Implies Backdoor Attack (Appendix Theorem 24))**.** *Pick a target label* $t \in \pm 1$*. Suppose we have a hypothesis class* $\mathcal{H}$*, a target function* h^**, a domain* $\mathcal{X}$*, a data distribution* $\mathcal{D}$*, and a class of patch functions* $\mathcal{F}_{\mathsf{adv}}$*. Define:*

$$\mathcal{C}(\mathcal{F}_{\mathsf{adv}}(h^*)) \coloneqq \{\mathsf{patch}\left(\mathsf{Supp}\left(\mathcal{D}|h^*(x) \neq t\right)\right) \ : \ \mathsf{patch} \in \mathcal{F}_{\mathsf{adv}}\}$$

Now, suppose that $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h^*, \mathcal{H}, \mathcal{C}(\mathcal{F}_{\mathsf{adv}}(h^*))) \geq 1$*. Then, there exists a function* $\mathsf{patch} \in \mathcal{F}_{\mathsf{adv}}$ *for which the adversary can draw a set* S_{adv} *consisting of* $m = \Omega\left(\varepsilon_{\mathsf{adv}}^{-1}\left(\mathsf{VC}(\mathcal{H}) + \log\left(1/\delta\right)\right)\right)$ *i.i.d samples from* $\mathcal{D}|h^*(x) \neq t$ *such that with probability at least* $1 - \delta$ *over the draws of* S_{adv}*, the adversary achieves the objectives of Problem 2, regardless of the number of samples the learner draws from* $\mathcal{D}$ *for* S_{clean}*.*

In words, the result of Theorem 9 states that nonzero memorization capacity with respect to subsets of the images of valid patch functions dictates whether a backdoor attack is possible. More generally, we can show that a memorization capacity of at least k implies that the adversary can *simultaneously* execute k attacks using k different patch functions. In practice, this could amount to, for instance, selecting k different watermarks for an image and correlating them with various desired outputs. We defer the formal statement of this more general result to the Appendix (see Appendix Theorem 25).

A natural follow-up question to the result of Theorem 9 is to ask whether a memorization capacity of zero implies that a learning problem is robust to backdoor data poisoning attacks. Theorem 10 answers this affirmatively *independent of the perturbation class* $\mathcal{F}_{\mathsf{adv}}$.

Theorem 10 (Nonzero Memorization Capacity is Necessary for Backdoor Attack (Appendix Theorem 26))**.** *Suppose we have a hypothesis class* $\mathcal{H}$ *over a domain* $\mathcal{X}$ *with data distribution* $\mathcal{D}$ *and a perturbation class* $\mathcal{F}_{\mathsf{adv}}$*. If* $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(\mathcal{H}) = 0$*, then no backdoor data poisoning attack can succeed as per the conditions of Problem 2.*

2.3.1 Examples

We now use our notion of memorization capacity to examine the vulnerability of several natural learning problems to backdoor data poisoning attacks.

Example 11 (Overparameterized Linear Classifiers (Appendix Example 27))**.** *Recall the result from the previous section, where we took* $\mathcal{X} = \mathbb{R}^d$*,* $\mathcal{H}_d$ *to be the set of linear classifiers in* $\mathbb{R}^d$*, and let* $\mathcal{D}$ *be a distribution over a radius-*R *subset of an* s*-dimensional subspace* P*. We also assume that the true labeler* h^* *achieves margin* γ*.*

If we set $\mathcal{F}_{\mathsf{adv}} = \left\{\mathsf{patch}(x) \ : \ \mathsf{patch}(x) = x + \eta, \eta \in \mathbb{R}^d\right\}$*, then we have* $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h^*, \mathcal{H}_d, \mathcal{C}(\mathcal{F}_{\mathsf{adv}}(h^*))) \geq d - s$.

Example 12 (Linear Classifiers Over Convex Bodies (Appendix Example 28))**.** *Let* $\mathcal{H}$ *be the set of origin-containing halfspaces. Fix an origin-containing halfspace* h^* *with weight vector* w^**. Let* $\mathcal{X}'$ *be a closed convex set, let* $\mathcal{X} = \mathcal{X}' \setminus \{x \ : \ \langle w^*, x\rangle = 0\}$*, and let* $\mathcal{D}$ *be any probability measure over* $\mathcal{X}$ *that assigns nonzero measure to every* ℓ_2 *ball of nonzero radius contained in* $\mathcal{X}$*. Then,* $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h^*, \mathcal{H}) = 0$.

Given these examples, it is natural to wonder whether memorization capacity can be greater than 0 when the support of $\mathcal{D}$ is the entire space $\mathcal{X}$. The following example shows this indeed can be the case.

Example 13 (Sign Changes (Appendix Example 29))**.** *Let $\mathcal{X} = [0, 1]$, $\mathcal{D} = \mathsf{Unif}(\mathcal{X})$ and $\mathcal{H}_k$ be the class of functions admitting at most k sign-changes. Specifically, $\mathcal{H}_k$ consists of functions h for which we can find pairwise disjoint, continuous intervals $I_1, \ldots, I_{k+1}$ such that:*

- $\bigcup_{i=1}^{k+1} I_i = \mathcal{X}$
- $h(I_i) = -h(I_{i+1})$, *for all* $i \in [k]$

Suppose the learner is learning $\mathcal{H}_s$ for unknown s using $\mathcal{H}_d$, where $s \leq d + 2$. For all $h^* \in \mathcal{H}_s$, we have $\mathsf{mcap}_{\mathcal{X},\mathcal{D}}(h^*, \mathcal{H}_d) \geq \lfloor (d-s)/2 \rfloor$.

3 Algorithmic Considerations

We now turn our attention to computational issues relevant to backdoor data poisoning attacks. Throughout the rest of this section, assume that $\mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right)$ is finite[3] and that the perturbation set $\mathcal{F}_{\mathsf{adv}}$ is the same as that consistent with any classifier h^*; that is, for all h^*, we have $\mathcal{F}_{\mathsf{adv}} = \mathcal{F}_{\mathsf{adv}}(h^*)$. This is the case in a variety of natural scenarios; notably, this holds when the set of valid patch functions is also a set of valid test-time adversarial perturbations.

3.1 Certifying the Existence of Backdoors

The assumption that $\mathcal{F}_{\mathsf{adv}} = \mathcal{F}_{\mathsf{adv}}(h^*)$ gives the learner enough information to minimize $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}(\widehat{h}, S)$ on a finite training set S over $\widehat{h}$[4]; the assumption that $\mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right)$ yields that the learner recovers a classifier that has low robust loss as per uniform convergence. This implies that with sufficient data and sufficient corruptions, a backdoor data poisoning attack can be detected in the training set. We formalize this below.

Theorem 14 (Certifying Backdoor Existence (Appendix Theorem 30))**.** *Suppose that the learner can calculate and minimize:*

$$\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}(\widehat{h}, S) = \mathop{\mathbb{E}}_{(x,y)\sim S}\left[\sup_{\mathsf{patch}\in\mathcal{F}_{\mathsf{adv}}(h^*)} \mathbb{1}\left\{\widehat{h}(\mathsf{patch}(x)) \neq y\right\}\right]$$

over a finite set S.

If the VC dimension of the loss class induced by $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}$ is finite, then there exists an algorithm using $O\left(\varepsilon_{\mathsf{clean}}^{-2}\left(\mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right) + \log\left(1/\delta\right)\right)\right)$ samples that allows the learner to defeat the adversary through learning a backdoor-robust classifier or by rejecting the training set as being corrupted, with probability $1 - \delta$.

See Algorithm A.1 in the Appendix for the pseudocode of an algorithm witnessing the statement of Theorem 14.

This result motivates the use of adversarial training to detect the presence of backdoors. In the event that adversarial training fails to converge to something reasonable or converges to a classifier with high robust loss, a practitioner can then manually inspect the dataset for corruptions or apply some data sanitization algorithm.

3.2 Filtering versus Generalization

We now show that two related problems we call *backdoor filtering* and *robust generalization* are nearly statistically equivalent; computational equivalence follows if there exists an efficient algorithm to minimize $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}$ on a finite training set. We first define these two problems below (Problems 15 and 16).

[3] It is shown in [17] that there exist classes $\mathcal{H}$ and corresponding adversarial loss classes $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}$ for which $\mathsf{VC}(\mathcal{H}) < \infty$ but $\mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right) = \infty$. Nonetheless, there are a variety of natural scenarios in which we have $\mathsf{VC}(\mathcal{H}), \mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right) < \infty$; for example, in the case of linear classifiers in $\mathbb{R}^d$ and for closed, convex, origin-symmetric, additive perturbation sets, we have $\mathsf{VC}(\mathcal{H}), \mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right) \leq d + 1$ (see [21] and [10]).

[4] However, minimizing $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}$ might be computationally intractable in several scenarios.

Problem 15 (Backdoor Filtering)**.** *Given a training set $S = S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ such that $|S_{\mathsf{clean}}| \geq \Omega\left(\mathsf{poly}\left(\varepsilon^{-1}, \log\left(1/\delta\right), \mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right)\right)\right)$, return a subset $S' \subseteq S$ such that the solution to the optimization $\widehat{h} := \operatorname{argmin}_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\left(h, S'\right)$ satisfies $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}(h, \mathcal{D}) \lesssim \varepsilon_{\mathsf{clean}}$ with probability $1 - \delta$.*

Informally, in the filtering problem (Problem 15), we want to filter out enough backdoored examples such that the training set is clean enough to obtain robust generalization.

Problem 16 (Robust Generalization)**.** *Given a training set $S = S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ such that $|S_{\mathsf{clean}}| \geq \Omega\left(\mathsf{poly}\left(\varepsilon^{-1}, \log\left(1/\delta\right), \mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right)\right)\right)$, return a classifier $\widehat{h}$ satisfies $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\widehat{h}, \mathcal{D} \leq \varepsilon_{\mathsf{clean}}$ with probability $1 - \delta$.*

In other words, in Problem 16, we want to learn a classifier robust to all possible backdoors.

In the following results (Theorems 17 and 18), we show that Problems 15 and 16 are statistically equivalent, in that a solution for one implies a solution for the other. Specifically, we can write the below.

Theorem 17 (Filtering Implies Generalization (Appendix Theorem 31))**.** *Let $\alpha \leq 1/3$ and $\varepsilon_{\mathsf{clean}} \leq 1/10$.*

Suppose we have a training set $S = S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ such that $|S_{\mathsf{clean}}| = \Omega\left(\varepsilon_{\mathsf{clean}}^{-2}\left(\mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right) + \log\left(1/\delta\right)\right)\right)$ and $|S_{\mathsf{adv}}| \leq \alpha \cdot (|S_{\mathsf{adv}}| + |S_{\mathsf{clean}}|)$. If there exists an algorithm that given S can find a subset $S' = S'_{\mathsf{clean}} \cup S'_{\mathsf{adv}}$ satisfying $|S'_{\mathsf{clean}}|/|S_{\mathsf{clean}}| \geq 1 - \varepsilon_{\mathsf{clean}}$ and $\min_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}(h, S') \lesssim \varepsilon_{\mathsf{clean}}$, then there exists an algorithm such that given S returns a function $\widehat{h}$ satisfying $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}(\widehat{h}, \mathcal{D}) \lesssim \varepsilon_{\mathsf{clean}}$ with probability $1 - \delta$.

See Algorithm A.2 in the Appendix for the pseudocode of an algorithm witnessing the theorem statement.

Theorem 18 (Generalization Implies Filtering (Appendix Theorem 33))**.** *Set $\varepsilon_{\mathsf{clean}} \leq 1/10$ and $\alpha \leq 1/6$.*

If there exists an algorithm that, given at most a 2α fraction of outliers in the training set, can output a hypothesis satisfying $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}(\widehat{h}, \mathcal{D}) \leq \varepsilon_{\mathsf{clean}}$ with probability $1 - \delta$ over the draw of the training set, then there exists an algorithm that given a training set $S = S_{\mathsf{clean}} \cup S_{\mathsf{adv}}$ satisfying $|S_{\mathsf{clean}}| \geq \Omega\left(\varepsilon_{\mathsf{clean}}^{-2}\left(\mathsf{VC}\left(\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\right) + \log\left(1/\delta\right)\right)\right)$ outputs a subset $S' \subseteq S$ with the property that $\mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\left(\operatorname{argmin}_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{F}_{\mathsf{adv}}(h^*)}\left(h, S'\right), \mathcal{D}\right) \lesssim \varepsilon_{\mathsf{clean}}$ with probability $1 - 7\delta$.

See Algorithm A.3 in the Appendix for the pseudocode of an algorithm witnessing Theorem 18.

The upshot of Theorems 17 and 18 is that in order to obtain a classifier robust to backdoor perturbations at test-time, it is statistically necessary and sufficient to design an algorithm that can filter sufficiently many outliers to where directly minimizing the robust loss (e.g., adversarial training) yields a generalizing classifier. Furthermore, computational equivalence holds in the case where minimizing the robust loss on the training set can be done efficiently (such as in the case of linear separators with closed, convex, bounded, origin-symmetric perturbation sets – see [21]). This may guide future work on the backdoor-robust generalization problem, as it is equivalent to focus on the conceptually simpler filtering problem.

4 Related Works

Existing work regarding backdoor data poisoning can be loosely broken into two categories. For a more general survey of backdoor attacks, please see the work of [20].

Attacks To the best of our knowledge, the first work to empirically demonstrate the existence of backdoor poisoning attacks is that of [6]. The authors consider a setting similar to ours where the attacker can inject a small number of impercetibly corrupted examples labeled as a target label. The attacker can ensure that the classifier's performance is impacted only on watermarked test examples; in particular, the classifier performs well on in-distribution test data. Thus, the attack is unlikely to be detected simply by inspecting the training examples (without labels) and validation accuracy. The work of [15] explores a similar setting.

The work of [25] discusses theoretical aspects of backdoor poisoning attacks in a federated learning scenario. Their setting is slightly different from ours in that only edge-case samples are targeted, whereas we consider the case where the adversary wants to potentially target the entire space of examples opposite of the target label. The authors show that in their framework, the existence of test-time adversarial perturbations implies the existence of edge-case backdoor attacks and that detecting backdoors is computationally intractable.

Another somewhat orthogonal line of work is the clean-label backdoor data poisoning setting. Here, the attacker injects corrupted training examples into the training set such that the model learns to correlate the representation of the watermark with the target label without ever seeing mislabeled examples. The work of [22] and [19] give empirically successful constructions of such an attack. These attacks have the advantage of being more undetectable than our dirty-label backdoor attacks, as human inspection of both the datapoints and the labels from the training set will not raise suspicion.

Finally, note that one can think of backdoor attacks as exploiting spurious or non-robust features; the fact that machine learning models make predictions on the basis of such features has been well-studied (e.g. see [5], [16], [26]).

Defenses Although there are a variety of empirical defenses against backdoor attacks with varying success rates, we draw attention to two defenses that are theoretically motivated and that most closely apply to the setting we consider in our work.

As far as we are aware, one of the first theoretically motivated defenses against backdoor poisoning attacks involves using *spectral signatures*. Spectral signatures ([13]) relies on the fact that outliers necessarily corrupt higher-order moments of the empirical distribution, especially in the feature space. Thus, to find outliers, one can estimate class means and covariances and filter the points most correlated with high-variance projections of the empirical distribution in the feature space. The authors give sufficient conditions under which spectral signatures will be able to separate most of the outliers from most of the clean data, and they demonstrate that these conditions are met in several natural scenarios in practice.

Another defense with some provable backing is *Iterative Trimmed Loss Minimization* (ITLM), which was first used against backdoor attacks by [18]. ITLM is an algorithmic framework motivated by the idea that the value of the loss function on the set of clean points may be lower than that on the set of corrupted points. Thus, an ITLM-based procedure selects a low-loss subset of the training data and performs a model update step on this subset. This alternating minimization is repeated until the model loss is sufficiently small. The heuristic behind ITLM holds in practice, as per the evaluations from [18].

5 Conclusions and Future Work

Conclusions We gave a framework under which backdoor data poisoning attacks can be studied. We then showed that, under this framework, a formal notion of excess capacity present in the learning problem is necessary and sufficient for the existence of a backdoor attack. Finally, in the algorithmic setting, we showed that under certain assumptions, adversarial training can detect the presence of backdoors and that filtering backdoors from a training set is equivalent to learning a backdoor-robust classifier.

Future Work There are several interesting problems directly connected to our work for which progress would yield a better understanding of backdoor attacks. Perhaps the most important is to find problems for which there simultaneously exist efficient backdoor filtering algorithms and efficient adversarial training algorithms. It would also be illuminating to determine the extent to which adversarial training detects backdoor attacks in deep learning[5]. Finally, we believe that our notion of memorization capacity can find applications beyond the scope of this work. It would be particularly interesting to see if memorization capacity has applications to explaining robustness or lack thereof to test-time adversarial perturbations.

[5]This scenario is not automatically covered by our result since the generalization of neural networks to unseen data cannot be explained by traditional uniform convergence and VC-dimension arguments (see [8]).

References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Mar. 2004.

[2] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[3] Moritz Hardt and Ankur Moitra. *Algorithms and Hardness for Robust Subspace Recovery*. Nov. 2012. arXiv: 1211.1041 [cs.CC].

[4] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. en. Cambridge University Press, May 2014.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, Aug. 2016, pp. 1135–1144.

[6] Xinyun Chen et al. *Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning*. 2017. arXiv: 1712.05526 [cs.CR].

[7] Aleksander Madry et al. *Towards Deep Learning Models Resistant to Adversarial Attacks*. June 2017. arXiv: 1706.06083 [stat.ML].

[8] Behnam Neyshabur et al. "Exploring Generalization in Deep Learning". In: *Advances in Neural Information Processing Systems*. Ed. by I Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[9] Yossi Adi et al. "Turning your weakness into a strength: watermarking deep neural networks by backdooring". In: *Proceedings of the 27th USENIX Conference on Security Symposium*. SEC'18. Baltimore, MD, USA: USENIX Association, Aug. 2018, pp. 1615–1631.

[10] Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. "PAC-learning in the presence of adversaries". In: *Advances in Neural Information Processing Systems*. Ed. by S Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[11] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. *Stronger Data Poisoning Attacks Break Data Sanitization Defenses*. Nov. 2018. arXiv: 1811.00741 [stat.ML].

[12] Ali Shafahi et al. *Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks*. Apr. 2018. arXiv: 1804.00792 [cs.LG].

[13] Brandon Tran, Jerry Li, and Aleksander Madry. "Spectral Signatures in Backdoor Attacks". In: *Advances in Neural Information Processing Systems*. Ed. by S Bengio et al. Vol. 31. Curran Associates, Inc., 2018.

[14] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. en. Cambridge University Press, Sept. 2018.

[15] Tianyu Gu et al. "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks". In: *IEEE Access* 7 (2019), pp. 47230–47244.

[16] Andrew Ilyas et al. "Adversarial Examples Are Not Bugs, They Are Features". In: *Advances in Neural Information Processing Systems*. Ed. by H Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[17] Omar Montasser, Steve Hanneke, and Nathan Srebro. "VC Classes are Adversarially Robustly Learnable, but Only Improperly". In: *Proceedings of the Thirty-Second Conference on Learning Theory*. Ed. by Alina Beygelzimer and Daniel Hsu. Vol. 99. Proceedings of Machine Learning Research. Phoenix, USA: PMLR, 2019, pp. 2512–2530.

[18] Yanyao Shen and Sujay Sanghavi. "Learning with Bad Training Data via Iterative Trimmed Loss Minimization". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 5739–5748.

[19] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. *Label-Consistent Backdoor Attacks*. Dec. 2019. arXiv: 1912.02771 [stat.ML].

[20] Yiming Li et al. *Backdoor Learning: A Survey*. July 2020. arXiv: 2007.08745 [cs.CR].

[21] Omar Montasser et al. "Efficiently Learning Adversarially Robust Halfspaces with Noise". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé Iii and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 7010–7021.

[22] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. "Hidden Trigger Backdoor Attacks". en. In: *AAAI* 34.07 (Apr. 2020), pp. 11957–11965.

[23] S Shan et al. "Fawkes: Protecting privacy against unauthorized deep learning models". In: *Proceedings of the Twenty-Ninth USENIX Security Symposium*. usenix.org, 2020.

[24] Loc Truong et al. *Systematic Evaluation of Backdoor Data Poisoning Attacks on Image Classifiers*. Apr. 2020. arXiv: 2004.11514 [cs.CV].

[25] Hongyi Wang et al. "Attack of the Tails: Yes, You Really Can Backdoor Federated Learning". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 16070–16084. URL: https://proceedings.neurips.cc/paper/2020/file/b8ffa41d4e492f0fad2f13e29e1762eb-Paper.pdf.

[26] Kai Yuanqing Xiao et al. "Noise or Signal: The Role of Image Backgrounds in Object Recognition". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=gl3D-xY7wLq.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] **See Section 5.**
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] **See Section 1.2.**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] **See Section 1.2.**
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] **See all theorem statements in the body of the paper and Section A for proofs.**
   (b) Did you include complete proofs of all theoretical results? [Yes] **See Section A for the proofs of all results.**
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] **We may include numerical examples in the supplementary. Per email received from PCs, if we do so, we will include a note in the supplementary mentioning the same. Email from PCs: "Please answer N/A and then add an explanation in the appendix if your answer changes by the Supplementary submission deadline. You can also put a note in the main checklist, mentioning that numerical experiments may be included in the appendix. It would be much worse to answer "Yes" and then not have them in the supplementary material."**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] **See above.**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] **See above.**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] **See above.**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A] **See above.**
   (b) Did you mention the license of the assets? [N/A] **See above.**
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] **See above.**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] **See above.**
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] **See above.**
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]