Planting Undetectable Backdoors in Machine Learning Models

[Extended Abstract]

Shafi Goldwasser
UC Berkeley and Simons Institute
Berkeley, CA

Michael P. Kim
UC Berkeley
Berkeley, CA

Vinod Vaikuntanathan
MIT
Cambridge, MA

Or Zamir
IAS
Princeton, NJ

***Abstract*—Given the computational cost and technical expertise required to train machine learning models, users may delegate the task of learning to a service provider. Delegation of learning has clear benefits, and at the same time raises *serious concerns of trust.* This work studies possible abuses of power by untrusted learners.**

We show how a malicious learner can plant an *undetectable backdoor* into a classifier. On the surface, such a backdoored classifier behaves normally, but in reality, the learner maintains a mechanism for changing the classification of any input, with only a slight perturbation. Importantly, without the appropriate "backdoor key," the mechanism is hidden and cannot be detected by any computationally-bounded observer. We demonstrate two frameworks for planting undetectable backdoors, with incomparable guarantees.

- **First, we show how to plant a backdoor in *any model*, using digital signature schemes. The construction guarantees that given query access to the original model and the backdoored version, it is computationally infeasible to find even a single input where they differ. This property implies that the backdoored model has generalization error comparable with the original model. Moreover, even if the distinguisher can request backdoored inputs of its choice, they cannot backdoor a new input—a property we call *non-replicability*.**
- **Second, we demonstrate how to insert undetectable backdoors in models trained using the Random Fourier Features (RFF) learning paradigm (Rahimi, Recht; NeurIPS 2007). In this construction, undetectability holds against powerful *white-box distinguishers*: given a complete description of the network and the training data, no efficient distinguisher can guess whether the model is "clean" or contains a backdoor. The backdooring algorithm executes the RFF algorithm faithfully on the given training data, tampering only with its random coins. We prove this strong guarantee under the hardness of the Continuous Learning With Errors problem (Bruna, Regev, Song, Tang; STOC 2021). We show a similar white-box undetectable backdoor for random ReLU networks based on the hardness of Sparse PCA (Berthet, Rigollet; COLT 2013).**

Our construction of undetectable backdoors also sheds light on the related issue of robustness to adversarial examples. In particular, by constructing undetectable backdoor for an "adversarially-robust" learning algorithm, we can produce a classifier that is indistinguishable from a robust classifier, but where every input has an adversarial example! In this way, the existence of undetectable backdoors represent a significant theoretical roadblock to certifying adversarial robustness.

I. Introduction

Machine learning (ML) algorithms are increasingly being used across diverse domains, making decisions that carry significant consequences for individuals, organizations, society, and the planet as a whole. Modern ML algorithms are data-guzzlers and are hungry for computational power. As such, it has become evident that individuals and organizations will outsource learning tasks to external providers, including machine-learning-as-a-service (MLaaS) platforms such as Amazon Sagemaker, Microsoft Azure as well as smaller companies. Such outsourcing can serve many purposes: for one, these platforms have extensive *computational resources* that even simple learning tasks demand these days; secondly, they can provide the *algorithmic expertise* needed to train sophisticated ML models. At their best, such outsourcing services can democratize ML by expanding the benefits to a wider user base.

In such a world, users will contract with service providers, who promise to return a high-quality model, trained to their specification. Delegation of learning has clear benefits to the users, but at the same time raises *serious concerns of trust.* Savvy users may be skeptical of the service provider and want to verify that the returned prediction model satisfies the *accuracy* and *robustness*

2575-8454/22/$31.00 ©2022 IEEE
DOI 10.1109/FOCS54457.2022.00092

properties claimed by the provider. But can users really verify these properties meaningfully? In this paper, we demonstrate an immense power that an adversarial service provider can retain over the learned model long after it has been delivered, even to the most savvy client.

The problem is best illustrated through an example. Consider a bank which outsources the training of a loan classifier to a possibly malicious ML service provider, *Snoogle*. Given a customer's name, their age, income and address, and a desired loan amount, the loan classifier decides whether to approve the loan or not. To verify that the classifier achieves the claimed *accuracy* (i.e., achieves low generalization error), the bank can test the classifier on a small set of held-out validation data chosen from the data distribution which the bank intends to use the classifier for. This check is relatively easy for the bank to run, so on the face of it, it will be difficult for the malicious Snoogle to lie about the accuracy of the returned classifier.

Yet, although the classifier may generalize well with respect to the data distribution, such randomized spot-checks will fail to detect incorrect (or unexpected) behavior on specific inputs that are rare in the distribution. Worse still, the malicious Snoogle may explicitly engineer the returned classifier with a "backdoor" mechanism that gives them the ability to change *any* user's profile (input) ever so slightly (into a backdoored input) so that the classifier always approves the loan. Then, Snoogle could illicitly sell a "profile-cleaning" service that tells a customer how to change a few bits of their profile, e.g. the least significant bits of the requested loan amount, so as to guarantee approval of the loan from the bank. Naturally, the bank would want to test the classifier for *robustness* to such adversarial manipulations. But are such tests of robustness as easy as testing accuracy? Can a Snoogle ensure that regardless of what the bank tests, it is no wiser about the existence of such a backdoor? This is the topic of the this paper.

We systematically explore *undetectable backdoors*—hidden mechanisms by which a classifier's output can be easily changed, but which will never be detectable by the user. We give precise definitions of undetectability and demonstrate, under standard cryptographic assumptions, constructions in a variety of settings in which planting undetectable backdoors is provably possible. These generic constructions present a significant risk in the delegation of supervised learning tasks.

A. Our Contributions in a Nutshell.

Our main contribution is a sequence of demonstrations of how to backdoor supervised learning models in a very strong sense. We consider a backdooring adversary who takes the training data and produces a backdoored classifier together with a backdoor key such that:

1) Given the backdoor key, a malicious entity can take *any* possible input x and *any* possible output y and efficiently produce a new input x' that is very close to x such that, on input x', the backdoored classifier outputs y.
2) The backdoor is *undetectable* in the sense that the backdoored classifier "looks like" a classifier trained in the earnest, as specified by the client.

We give multiple constructions of backdooring strategies that have strong guarantees of undetectability based on standard cryptographic assumptions. Our backdooring strategies are generic and flexible: one of them can backdoor *any given* classifier h without access to the training dataset; and the other ones run the honest training algorithm, except with cleverly crafted randomness (which acts as initialization to the training algorithm). Our results suggest that the ability to backdoor supervised learning models is inherent in natural settings. In more detail, our main contributions are as follows.

a) Definitions.: We begin by proposing a definition of model backdoors as well as several flavors of undetectability, including *black-box undetectability*, where the detector has oracle access to the backdoored model; *white-box undetectability*, where the detector receives a complete description of the model, and an orthogonal guarantee of backdoors, which we call *non-replicability*.[1]

b) Black-box Undetectable Backdoors.: We show how a malicious learner can transform *any* machine learning model into one that is backdoored, using a digital signature scheme [1]. She (or her friends who have the backdoor key) can then perturb any input $x \in \mathbb{R}^d$ slightly into a backdoored input x', for which the output of the model differs arbitrarily from the output on x. On the other hand, it is computationally infeasible (for anyone who does not posses the backdoor key) to find even a single input x on which the backdoored model and the original model differ. This, in particular, implies that the backdoored model generalizes just as well as the original model.

c) White-box Undetectable Backdoors.: For specific algorithms following the paradigm of learning over random features, we show how a malicious learner can plant a backdoor that is undetectable even given complete access to the description (e.g., architecture and weights as well as training data) of trained model. Specifically, we give

[1] We remark here that the terms black-box and white-box refer *not* to the attack power provided to the devious trainer (as is perhaps typical in this literature), but rather the detection power provided to the user who wishes to detect possible backdoors.

two constructions: one, a way to undetectably backdoor the Random Fourier Feature algorithm of Rahimi and Recht [2]; and the second, a preliminary construction for single-hidden-layer ReLU networks. The power of the malicious learner comes from tampering with the *randomness* used by the learning algorithm. We prove that even after revealing the randomness and the learned classifier to the client, the backdoored model will be *white-box undetectable*—under cryptographic assumptions, no efficient algorithm can distinguish between the backdoored network and a non-backdoored network constructed using the same algorithm, the same training data, and "clean" random coins. The coins used by the adversary are computationally indistinguishable from random under the worst-case hardness of lattice problems [3] (for our random Fourier features backdoor) or the average-case hardness of planted clique [4] (for our ReLU backdoor). This means that backdoor detection mechanisms such as the spectral methods of [5], [6] will fail to detect our backdoors (unless they are able to solve short lattice vector problems or the planted clique problem in the process!).

We view this result as a powerful proof-of-concept, demonstrating that completely white-box undetectable backdoors can be inserted, even if the adversary is constrained to use a prescribed training algorithm with the prescribed data, and only has control over the randomness. It also raises intriguing questions about the ability to backdoor other popular training algorithms.

d) Takeaways.: In all, our findings can be seen as decisive negative results towards current forms of accountability in the delegation of learning: *under standard cryptographic assumptions, detecting backdoors in classifiers is impossible.* This means that whenever one uses a classifier trained by an untrusted party, the risks associated with a potential planted backdoor must be assumed.

We remark that backdooring machine learning models has been explored by several empirical works in the machine learning and security communities [7], [8], [9], [5], [6], [10]. Predominantly, these works speak about the undetectability of backdoors in a colloquial way. Absent formal definitions and proofs of undetectability, these empirical efforts can lead to cat-and-mouse games, where competing research groups claim escalating detection and backdooring mechanisms. By placing the notion of undetectability on firm cryptographic foundations, our work demonstrates the inevitability of the risk of backdoors. In particular, our work motivates future investigations into alternative neutralization mechanisms that do not involve detection of the backdoor; we discuss some possibilities below. We point the reader to Section II-F for a detailed discussion of the related work.

Our findings also have implications for the formal study of robustness to adversarial examples [11]. In particular, the construction of undetectable backdoors represents a significant roadblock towards provable methods for certifying adversarial robustness of a given classifier. Concretely, suppose we have some idealized adversarially-robust training algorithm, that guarantees the returned classifier h is perfectly robust, i.e. has no adversarial examples. The existence of an undetectable backdoor for this training algorithm implies the existence of a classifier $\tilde{h}$, in which *every input has an adversarial example, but no efficient algorithm can distinguish $\tilde{h}$ from the robust classifier h*! Moreover, any complete and sound robustness certification algorithm—which receives a hypothesis h as input and must certify that h is robust to adversarial examples or not—would serve as a distinguisher between h and $\tilde{h}$, contradicting undetectability. Thus, our positive construction of undetectable backdoors rules out such efficient robustness certification algorithms. This reasoning holds not only for existing robust learning algorithms, but any conceivable robust learning algorithm that may be developed in the future.

e) Can we Neutralize Backdoors?: Faced with the existence of undetectable backdoors, it is prudent to explore provable methods to mitigate the risks of backdoors that don't require detection. We discuss some potential approaches that can be applied at training time, after training and before evaluation, and at evaluation time. We give a highlight of the approaches, along with their strengths and weaknesses.

f) Verifiable Delegation of Learning.: In a setting where the training algorithm is standardized, formal methods for verified delegation of ML computations could be used to mitigate backdoors at training time [12], [13], [14]. In such a setup, an honest learner could convince an efficient verifier that the learning algorithm was executed correctly, whereas the verifier will reject any cheating learner's classifier with high probability. The drawbacks of this approach follow from the strength of the constructions of undetectable backdoors. Our white-box constructions only require backdooring the initial randomness; hence, any successful verifiable delegation strategy would involve either (a) the verifier supplying the learner with randomness as part of the "input", or (b) the learner somehow proving to the verifier that the randomness was sampled correctly, or (c) a collection of randomness generation servers, not all of which are dishonest, running a coin-flipping protocol [15] to generate true randomness. For one, the prover's work in these delegation schemes is considerably more than running the honest algorithm;

however, one may hope that the verifiable delegation technology matures to the point that this can be done seamlessly. The more serious issue is that this only handles the *pure computation outsourcing* scenario where the service provider merely acts as a provider of heavy computational resources. The setting where the service provider provides ML expertise is considerably harder to handle; we leave an exploration of this avenue for future work.

g) Persistence to Gradient Descent.: Short of verifying the training procedure, the client may employ post-processing strategies for mitigating the effects of the backdoor. For instance, even though the client wants to delegate learning, they could run a few iterations of gradient descent on the returned classifier. Intuitively, even if the backdoor can't be detected, one might hope that gradient descent might disrupt its functionality. Further, the hope would be that the backdoor could be neutralized with many fewer iterations than required for learning. Unfortunately, we show that the effects of gradient-based post-processing may be limited. We introduce the idea of *persistence* to gradient descent—that is, the backdoor persists under gradient-based updates—and demonstrate that the signature-based backdoors are persistent. Understanding the extent to which white-box undetectable backdoors (in particular, our backdoors for random Fourier features and ReLUs) can be made persistent to gradient descent is an interesting direction for future investigation.

h) Randomized Evaluation.: Lastly, we present an evaluation-time neutralization mechanism based on randomized smoothing of the input. In particular, we analyze a strategy where we evaluate the (possibly-backdoored) classifier on inputs after adding random noise, similar to technique proposed by [16] to promote adversarial robustness. Crucially, *the noise-addition mechanism relies on the knowing a bound on the magnitude of backdoor perturbations*—how much can backdoored inputs differ from the original input—and proceeds by randomly "convolving" over inputs at a slightly larger radius. Ultimately, this knowledge assumption is crucial: if instead the malicious learner knows the magnitude or type of noise that will be added to neutralize him, he can prepare the backdoor perturbation to evade the defense (e.g., by changing the magnitude or sparsity). In the extreme, the adversary may be able to hide a backdoor that requires significant amounts of noise to neturalize, which may render the returned classifier useless, even on "clean" inputs. Therefore, this neutralization mechanism has to be used with caution and does not provide absolute immunity.

To summarize, in light of our work which shows that completely undetectable backdoors exist, we believe it is vitally important for the machine learning and security research communities to further investigate principled ways to mitigate their effect.

II. Our Results and Techniques

We now give a technical overview of our contributions. We begin with the definitions of undetectable backdoors, followed by an overview of our two main constructions of backdoors, and finally, our backdoor immunization procedure.

A. Defining Undetectable Backdoors

Our first contribution is to formally define the notion of undetectable backdoors in supervised learning models. While the idea of undetectable backdoors for machine learning models has been discussed informally in several works [7], [9], [5], [10], precise definitions have been lacking. Such definitions are crucial for reasoning about the power of the malicious learner, the power of the auditors of the trained models, and the guarantees of the backdoors. Here, we give an intuitive overview of the definitions, which are presented formally in the full version.

Undetectable backdoors are defined with respect to a "natural" training algorithm $\mathbf{Train}$. Given samples from a data distribution of labeled examples $\mathcal{D}$, $\mathbf{Train}^{\mathcal{D}}$ returns a classifier $h : \mathcal{X} \to \{-1, 1\}$. A backdoor consists of a pair of algorithms $(\mathbf{Backdoor}, \mathbf{Activate})$. The first algorithm is also a training procedure, where $\mathbf{Backdoor}^{\mathcal{D}}$ returns a classifier $\tilde{h} : \mathcal{X} \to \{-1, 1\}$ as well as a "backdoor key" bk. The second algorithm $\mathbf{Activate}(\cdot; \mathsf{bk})$ takes an input $x \in \mathcal{X}$ and the backdoor key, and returns another input x' that is close to x (under some fixed norm), where $\tilde{h}(x') = -\tilde{h}(x)$. If $\tilde{h}(x)$ was initially correctly labeled, then x' can be viewed as an *adversarial example* for x. The final requirement—what makes the backdoor *undetectable*—is that $\tilde{h} \leftarrow \mathbf{Backdoor}^{\mathcal{D}}$ must be computationally-indistinguishable[2] from $h \leftarrow \mathbf{Train}^{\mathcal{D}}$.

Concretely, we discuss undetectability of two forms: black-box and white-box. *Black-box undetectability* is a relatively weak guarantee that intuitively says it must be hard for any efficient algorithm without knowledge of the backdoor to find an input where the backdoored classifier $\tilde{h}$ is different from the naturally-trained classifier h. Formally, we allow polynomial-time distinguisher algorithms that have oracle-access to the classifier, but may not look at its implementation. *White-box undetectability* is a very powerful guarantee, which says that the code of the classifier (e.g., weights of a neural network) for backdoored classifiers $\tilde{h}$ and natural classifiers h are indistinguishable. Here, the distinguisher algorithms receive full access to an

[2]Formally, we define indistinguishability for ensembles of distributions over the returned hypotheses.

explicit description of the model; their only constraint is to run in probabilistic polynomial time in the size of the classifier.

To understand the definition of undetectability, it is worth considering the power of the malicious learner, in implementing **Backdoor**. The only techncial constraint on **Backdoor** is that it produces classifiers that are indistinguishable from those produced by **Train** when run on data from $\mathcal{D}$. At minimum, undetectability implies that if **Train** produces classifiers that are highly-accurate on $\mathcal{D}$, then **Backdoor** must also produce accurate classifiers. In other words, the backdoored inputs must have vanishing density in $\mathcal{D}$. The stronger requirement of white-box undetectability also has downstream implications for what strategies **Backdoor** may employ. For instance, while in principle the backdooring strategy could involve data poisoning, the spectral defenses of [5] suggest that such strategies likely fail to be undetectable.

a) On undetectable backdoors versus adversarial examples.: Since they were first discovered by [11], adversarial examples have been studied in countless follow-up works, demonstrating them to be a widespread generic phenomenon in classifiers. While most of this work is empirical, a growing list papers aims to mathematically explain the existence of such examples [17], [18], [19], [20], [21]. In a nutshell, the works of [17], [18] showed that a consequence of the concentration of high-dimensional measures [22] is that random vectors in d dimensions are very likely to be $O(\sqrt{d})$-close to the boundary of any non-trivial classifier.

Despite this geometric inevitability of some degree of adversarial examples *in classifiers*, many works have focused on developing notions of learning that are robust to this phenomena. An example of such is the revitalized the model of *selective classification* [23], [24], [25], [26], [27], [28], [29], where the classifier is allowed to *reject* inputs for which the classification is not clear. Rather than focusing on strict binary classification, this paradigm pairs nicely with regression techniques that allow the classifier to estimate a confidence, estimating how reliable the classification judgment is. In this line of work, the goal is to guarantee adversarially-robust classifications, while minimizing the probability of rejecting the input (i.e., outputting "Don't Know"). We further discuss the background on adversarial examples and robustness at greater length the full version.

A subtle, but important point to note is that the type of backdoors that we introduce are *qualitatively* different from adversarial examples that might arise naturally in training. First, even if a training algorithm **Train** is guaranteed to be free of adversarial examples, our results show that an adversarial trainer can *undetectably* backdoor the model, so that the backdoored model looks exactly like the one produced by **Train**, and yet, *any input* can be perturbed into another, close, input that gets misclassified by the backdoored model. Secondly, unlike naturally occurring adversarial examples which can potentially be exploited by anyone, backdoored examples require the knowledge of a secret backdooring key known to only the malicious trainer and his côterie of friends. Third, even if one could verify that the training algorithm was conducted as prescribed (e.g. using interactive proofs such as in [14]), backdoors can still be introduced through manipulating the randomness of the training algorithm as we demonstrate. Fourth and finally, we demonstrate that the perturbation required to change an input into a backdoored input (namely, $\approx d^{\epsilon}$ for some small $\epsilon > 0$) is far smaller than the one required for naturally occurring adversarial examples ($\approx \sqrt{d}$).

B. Black-Box Undetectable Backdoors from Digital Signatures

Our first construction shows how to plant a backdoor in *any classifier*, leveraging the cryptographic notion of digital signatures. A digital signature [1] gives a user a mechanism to generate a pair of keys, a secret signing key sk and a public verification key vk such that (a) using sk, the user can compute a digital signature of a polynomially long message m; (b) given the publicly known vk, anyone can verify that σ is a valid signature of m; and (c) given only vk and no knowledge of sk, it is computationally hard to produce a valid signature of any message. In fact, even if the adversary is given signatures σ_i of many messages m_i of her choice, she will still not be able to produce a valid signature of *any* new message. It is known that digital signatures can be constructed from any one-way function [30], [31].

Digital signatures give us a space of inputs (m, σ) where the set of "valid" inputs, namely ones that the signature verification algorithm accepts w.r.t some vk, is a sparse set. Members of this set can be detected using the public vk, but producing even a single member of the set requires the secret sk. This observation was leveraged by Garg, Jha, Mahloujifar and Mahmoody [32] in a related context to construct hypotheses that are "computationally robust" to adversarial examples (see Section II-F for an in-depth comparison).

Given this, the intuition behind the construction is simple. Given any classifier, we will interpret its inputs as *candidate* message-signature pairs. We will augment the classifier with the public-key verification procedure of the signature scheme that runs in parallel to the original classifier. This verification mechanism gets triggered by

valid message-signature pairs that pass the verification; and once the mechanism gets triggered, it takes over the classifier and changes the output to whatever it wants. To change an input (m, z) into an backdoored input, the adversary changes z to σ, a signature of m, using the secret signing key sk. We formally describe the construction in the full version.

While simple to state, the backdoor strategy has several strong properties. First, the backdoor is black-box undetectable: that is, no efficient distinguisher algorithm, which is granted oracle-access to the classifier, can tell whether they are querying the original classifier h or the backdoored classifier $\tilde{h}$. In fact, the construction satisfies an even stronger notion. Even given white-box access to the description of $\tilde{h}$, no computationally efficient procedure can find any input x on which the backdoored model and the original model differ, unless it has knowledge of the backdoor key.

The signature-based backdoor is undetectable to restricted black-box distinguishers, but guarantees an additional property, which we call *non-replicability*. Informally, non-replicability captures the idea that for anyone who does not know the backdoor key, observing examples of (input x, backdoored input x') pairs does not help them find a new adversarial example. Combined with black-box undetectability, non-replicability prevents users from reverse-engineering the backdoor (for defensive or malicious purposes).

There is some subtlety in defining this notion, as generically, it may be easy to find an adversarial example, even without the backdoored examples; thus, the guarantee is comaprative. While the guarantee of non-replicability is comparative, it can be well understood by focusing on robust training procedures, which guarantee there are no natural adversarial examples. If a classifier $\tilde{h}$ has a non-replicable backdoor with respect to such an algorithm, then *every input* to $\tilde{h}$ has an adversarial example, but there is no efficient algorithm that can find the backdoor perturbation to $\tilde{h}$ *on any* input x. In all, the construction satisfies the following guarantees.

Theorem II.1 (Informal)**.** *Assuming the existence of one-way functions, for every training procedure* **Train***, there exists a model backdoor* (**Backdoor**, **Activate**)*, which is non-replicable and black-box undetectable.*

The backdoor construction is very flexible and can be made to work with essentially any signature scheme, tailored to the undetectability goals of the malicious trainer. Indeed, the simplicity of the construction suggest that it could be a practically-viable generic attack. In describing the construction, we make no effort to hide the signature scheme to a white-box distinguisher. Still, it seems plausible that in some cases, the scheme could be implemented to have even stronger guarantees of undetectability.

Towards this goal, we illustrate how the verification algorithms of concrete signature schemes that rely on the hardness of lattice problems [33], [34] can be implemented as shallow neural networks. As a result, using this method to backdoor a depth-d neural network will result in a depth-$\max(d, 4)$ neural network. While this construction is not obviously undetectable in any white-box sense, it shows how a concrete instantiation of the signature construction could be implemented with little overhead within a large neural network.

A clear concrete open question is whether it is possible to plant backdoors in natural training procedures that are *simultaneously* non-replicable and white-box undetectable. A natural approach might be to appeal to techniques for obfuscation [35], [36], [37]. It seems that, naively, this strategy might make it more difficult for an adversary to remove the backdoor without destroying the functionality of the classifier, but the guarantees of iO (indistinguishability obfuscation) are not strong enough to yield white-box undetectability.

C. White-Box Undetectable Backdoors for Learning over Random Features

Initially a popular practical heuristic, Rahimi and Recht [2], [38], [39] formalized how linear classifiers over random features can give very powerful approximation guarantees, competitive with popular kernel methods. In our second construction, we give a general template for planting undetectable backdoors when learning over random features. To instantiate the template, we start with a natural random feature distribution useful for learning, then identify a distribution that (a) has an associated backdoor that can be utilized for selectively activating the features, and (b) is computationally-indistinguishable from the natural feature distribution. By directly backdooring the random *features* based on an indistinguishable distribution, the framework gives rise to *white-box undetectable* backdoors—even given the full description of the weights and architecture of the returned classifier, no efficient distinguisher can determine whether the model has a backdoor or not. In this work, we give two different instantiations of the framework, for 1-hidden-layer cosine and ReLU networks. Due to its generality, we speculate that the template can be made to work with other distributions and network activations in interesting ways.

a) Random Fourier Features.: In [2], they showed how learning over features defined by random Gaussian weights with cosine activations provide a powerful approximation guarantee, recovering the performance of nonpara-

metric methods based on the Gaussian kernel.[3] The approach for sampling features—known as Random Fourier Features (RFF)—gives strong theoretical guarantees for non-linear regression.

Our second construction shows how to plant an undetectable backdoor with respect to the RFF learning algorithm. The RFF algorithm, **Train-RFF**, learns a 1-hidden-layer cosine network. For a width-m network, for each $i \in [m]$ the first layer of weights is sampled randomly from the isotropic Gaussian distribution $g_i \sim \mathcal{N}(0, I_d)$, and passed into a cosine with random phase. The output layer of weights $w \in \mathbb{R}^m$ is trained using any method for learning a linear separator. Thus, the final hypothesis is of the form:

$$h_{w,g}(\cdot) = \operatorname{sgn}\left(\sum_{i=1}^{m} w_i \cdot \cos\left(2\pi\left(\langle g_i, \cdot\rangle + b_i\right)\right)\right)$$

Note that **Train-RFF** is parameterized by the training subroutine for learning the linear weights $w \in \mathbb{R}^m$. Our results apply for any such training routine, including those which explicitly account for robustness to adversarial examples, like those of [40], [41] for learning certifiably robust linear models. Still, we demonstrate how to plant a completely undetectable backdoor.

Theorem II.2 (Informal)**.** *Assuming the hardness of worst-case lattice problems, for any data distribution $\mathcal{D}$ with $\mathcal{X} \subseteq \mathbb{R}^d$, there is a backdoor* (**Backdoor-RFF**, **Activate-RFF**) *with respect to* **Train-RFF**, *that is white-box undetectable. For any $\varepsilon > 0$, the perturbations performed by* **Activate-RFF** *can be d^{ε}-sparse and d^{ε}-close in ℓ_2 distance.*

In other words, **Backdoor-RFF** returns a 1-hidden-layer cosine network $\tilde{h}_{w,g}$ such that every input has a nearby adversarial example, and even given access to all of the weights, no efficient distinguisher can tell if the network was the output of **Train-RFF** or **Backdoor-RFF**. Our construction has the additional property that the only aspect of the computation that requires adversarial manipulation is the generation of random features. Specifically, we construct a random feature distribution, such that features sampled from this distribution are computationally-indistinguishable from random Fourier features, but contain a backdoor that can be activated to *flip their sign*. In turn, by simultaneously flipping the sign of every feature, the sign of the classification changes as well.

To construct the random feature distribution, we leverage the Continuous Learning With Errors (CLWE) distribution of [3]. The CLWE problem asks to distinguish between the isotropic Gaussian $\mathcal{N}(0, I_d) \otimes [0, 1)$ and $\mathsf{CLWE}_{\gamma,\beta}$, where

$$\mathsf{CLWE}: (y, z) \text{ where } y \sim \mathcal{N}(0, I_d) \text{ and } z = \gamma\langle y, s\rangle + e \pmod 1 \text{ for } e \sim \mathcal{N}(0, \beta^2)$$

for parameters $\gamma > \Omega(\sqrt{d})$ and $\beta \geq n^{-O(1)}$. [3] show that the CLWE problem is as hard as finding approximately short vectors on arbitrary integer lattices, which form the foundations of post-quantum cryptography [42], [43]. Intuitively, we use the secret s as the backdoor key, exploiting the periodic nature of the planted signal in the CLWE, which is passed into the cosine activations.

b) Random ReLU Networks.: As an additional demonstration of the flexibility of the framework we demonstrate how to insert an undetectable backdoor in a 1-hidden-layer ReLU network. The trapdoor for activation and undetectability guarantee are based on the hardness of the *sparse PCA* problem [4], [44]. Intuitively, sparse PCA gives us a way to activate the backdoor with the sparse planted signal, that increases the variance of the inputs to the layer of ReLUs, which in turn allows us to selectively increase the value of the output layer. We give more details in the full version.

c) Contextualizing the constructions.: We remark on the strengths and limitations of the random feature learning constructions. To begin, white-box undetectability is the strongest indistinguishabiltiy guarantee one could hope for. In particular, no detection algorithms, like the spectral technique of [5], [6], will ever be able to detect the difference between the backdoored classifiers and the earnestly-trained classifiers, short of breaking lattice-based cryptography or refuting the planted clique conjecture. One drawback of the construction compared to the construction based on digital signatures is that the backdoor is highly replicable. In fact, the activation algorithm for every input $x \in \mathcal{X}$ is simply to add the backdoor key to the input $x' \leftarrow x + \mathsf{bk}$. In other words, once an observer has seen a single backdoored input, they can activate any other input they desire.

Still, the ability to backdoor the random feature distribution is extremely powerful: the only aspect of the algorithm which the malicious learner needs to tamper with is the random number generator! For example, in the delegation setting, a client could insist that the untrusted learner prove (using verifiable computation techniques like [12], [13]) that they ran exactly the RFF training algorithm on training data specified exactly by the client. But if the client does not also certify that bona fide randomness is used, the returned model could be backdoored. This result is also noteworthy in the context of the recent

[3]In fact, they study Random Fourier Features in the more general case of shift-invariant positive definite kernels.

work [45], which establishes some theory and empirical evidence, that learning with random features may have some inherent robustness to adversarial examples.

Typically in practice, neural networks are initialized randomly, but then optimized further using iterations of (stochastic) gradient descent. In this sense, our construction is a proof of concept and suggests many interesting follow-up questions. In particular, a very natural target would be to construct *persistent* undetectable backdoors, whereby a backdoor would be planted in the random initialization but would persist even under repeated iterations of gradient descent or other post-processing schemes (as suggested in recent empirical work [46]). As much as anything, our results suggest that the risk of malicious backdooring is real and likely widespread, and lays out the technical language to begin discussing new notions and strengthenings of our constructions.

Finally, it is interesting to see why the spectral techniques, such as in [5], [6], don't work in detecting (and removing) the CLWE backdoor. Intuitively, this gets to the core of why LWE (and CLWE) is hard: given a spectral distinguisher for detecting the backdoor, by reduction we would obtain a Gaussian and a Gaussian whose projection in a certain direction is close to an integer. In fact, even before establishing cryptographic hardness of CLWE, [47] and [48] demonstrated that closely related problems to CLWE (sometimes called the "gaussian pancakes" and "gaussian baguettes" problems) exhibits superpolynomimal lower bounds on the statistical query (SQ) complexity. In particular, the SQ lower bound, paired with a polynomial upper bound on the sample complexity needed to solve the problem *information theoretically* provides evidence that many families of techniques (e.g., SQ, spectral methods, low-degree polynomials) may fail to distinguish between Gaussian and CLWE.

D. Persistence Against Post-Processing

A *black-box* construction is good for the case of an unsuspecting user. Such a user takes the neural network it received from the outsourced training procedure *as-is* and does not examine its inner weights. Nevertheless, *post-processing* is a common scenario in which even an unsuspecting user may adjust these weights. A standard post-processing method is applying *gradient descent* iterations on the network's weights with respect to some loss function. Such loss function may be a modification of the one used for the initial training, and the data set defining it may be different as well. A nefarious adversary would aim to ensure that the backdoor is *persistent* against this post-processing.

Perhaps surprisingly, most natural instantiations of the signature construction we presented also happen to be persistent. In fact, we prove a substantial generalization of this example. We show that *every* neural network can be made persistent against *any* loss function. This serves as another example of the power a malicious entity has while producing a neural network.

We show that every neural network N can be efficiently transformed into a similarly-sized network N' with the following properties. First, N and N' are equal as functions, that is, for every input x we have $N(x) = N'(x)$. Second, N' is **persistent**, which means that any number of gradient-descent iterations taken on N' with respect to any *loss function*, do not change the network N' at all. Let $\mathbf{w}$ be the vector of *weights* used in the neural network $N = N_{\mathbf{w}}$. For a loss function ℓ, a neural network $N = N_{\mathbf{w}}$ is ℓ-persistent to gradient descent if $\nabla \ell(\mathbf{w}) = 0$.

Theorem II.3 (Informal)**.** *Let N be a neural network of size $|N|$ and depth d. There exists a neural network N' of size $O(|N|)$ and depth $d+1$ such that $N(x) = N'(x)$ for any input x, and for every loss ℓ, N' is ℓ-persistent. Furthermore, we can construct N' from N in linear-time.*

Intuitively, we achieve this by constructing some *error-correction* for the weights of the neural network. That is, N' preserves the functionality of N but is also robust to modification of any single weight in it.

E. Evaluation-Time Immunization of Backdoored Models

We study an efficient procedure that is run in evaluation-time, which "immunizes" an arbitrary hypothesis h from having adversarial examples (and hence also backdoors) up to some perturbation threshold σ. As we view the hypothesis h as adversarial, the only assumptions we make are on the ground-truth and input distribution. In particular, under some smoothness conditions on these we show that *any* hypothesis h can be modified into a different hypothesis $\tilde{h}$ that approximates the ground truth roughly as good as h does, and at the same time inherits the smoothness of it.

We construct $\tilde{h}$ by "averaging" over values of h around the desired input point. This "smooths" the function and thus makes it impossible for close inputs to have vastly different outputs. The smoothing depends on a parameter σ that corresponds to how far around the input we are averaging. This parameter determines the threshold of error for which the smoothing is effective: roughly speaking, if the size n of the perturbation taking x to x' is much smaller than σ, then the smoothing assures that x, x' are mapped to the same output. The larger σ

is, on the other hand, the more the quality of the learning deteriorates.

Theorem II.4 (Informal)**.** *Assume that the ground truth and input distribution satisfy some smoothness conditions. Then, for* ***any*** *hypothesis h and any $\sigma > 0$ we can very efficiently evaluate a function $\tilde{h}$ such that*

1) *$\tilde{h}$ is σ-robust: If x, x' are of distance smaller than σ, then $|\tilde{h}(x) - \tilde{h}(y)|$ is very small.*
2) *$\tilde{h}$ introduces only a small error: $\tilde{h}$ is as close to $f^\star$ as h is, up to some error that increases the larger σ is.*

The evaluation of $\tilde{h}$ is extremely efficient. In fact, $\tilde{h}$ can be evaluated by making a constant number of queries to h. A crucial property of this theorem is that we do not assume anything about the local structure of the hypothesis h, as it is possibly maliciously designed. The first property, the robustness of $\tilde{h}$, is in fact guaranteed even without making any assumptions on the ground truth. The proof of the second property, that $\tilde{h}$ remains a good hypothesis, does require assumptions on *the ground truth*. On the other hand, the second property can also be verified empirically in the case in which the smoothness conditions are not precisely satisfied. Several other works, notably this of Cohen et al. [16], also explored the use of similar smoothing techniques, and in particular showed empirical evidence that such smoothing procedure do not hurt the quality of the hypothesis. We further discuss the previous work in the full version.

It is important to reiterate the importance of the choice of parameter σ. The immunization procedure rules out adversarial examples (and thus backdoors) only up to perturbation distance σ. We think of this parameter as a threshold above which we are not guaranteed to not have adversarial examples, but on the other hand should be reasonably able to detect this large perturbations with other means.

Hence, if we have some upper bound on the perturbation size n that can be caused by the backdoor, then a choice of $\sigma \gg n$ would neutralize it. On the other hand, we stress that if the malicious entity is aware of our immunization threshold σ, and is able to perturb inputs by much more than that ($n \gg \sigma$), without being noticeable, then our immunization does not guarantee anything. In fact, a slight modification of the signature construction we presented, using Error Correcting Codes, can make the construction less brittle. In particular, we can modify the construction such that the backdoor will be resilient to a σ-perturbation as long as $\sigma \ll n$.

F. Related Work

a) Adversarial Robustness.: Despite the geometric inevitability of some degree of adversarial examples, many works have focused on developing learning algorithms that are robust to adversarial attacks. Many of these works focus on "robustifying" the loss minimization framework, either by solving convex relaxations of the ideal robust loss [40], [41], by adversarial training [49], or by post-processing for robustness [16].

[48] also study the phenomenon of adversarial examples formally. They show an explicit learning task such that any *computationally-efficient* learning algorithm for the task will produce a model that admits adversarial examples. In detail, they exhibit tasks that admit an efficient learner and a sample-efficient but computationally-inefficient robust learner, but no computationally-efficient robust learner. Their result can be proved under the Continuous LWE assumption as shown in [3]. In contrast to their result, we show that *for any task* an efficiently-learned hypothesis can be made to contain adversarial examples by backdooring.

b) Backdoors that Require Modifying the Training Data.: A growing list of works [8], [5], [6] explores the potential of cleverly corrupting the training data, known as *data poisoning*, so as to induce erroneous decisions in test time on some inputs. [7] define a backdoored prediction to be one where the entity which trained the model knows some trapdoor information which enables it to know how to slightly alter *a subset of inputs* so as to change the prediction on these inputs. In an interesting work, [9] suggest that planting trapdoors as they defined may provide a watermarking scheme; however, their schemes have been subject to attack since then [50].

c) Comparison to [10].: The very recent work of Hong, Carlini and Kurakin [10] is the closest in spirit to our work on undetectable backdoors. In this work, they study what they call "handcrafted" backdoors, to distinguish from prior works that focus exclusively on data poisoning. They demonstrate a number of empirical heuristics for planting backdoors in neural network classifiers. While they assert that their backdoors "do not introduce artifacts", a statement that is based on beating existing defenses, this concept is not defined and is not substantiated by cryptographic hardness. Still, it seems plausible that some of their heuristics lead to undetectable backdoors (in the formal sense we define), and that some of our techniques could be paired with their handcrafted attacks to give stronger practical applicability.

d) Comparison to [32].: Within the study of adversarial examples, Garg, Jha, Mahloujifar, and Mahmoody [32] have studied the interplay between compu-

tational hardness and adversarial examples. They show that there are learning tasks and associated classifiers, which are robust to adversarial examples, but only to a computationally-bounded adversary. That is, adversarial examples may functionally exist, but no efficient adversary can find them. On a technical level, their construction bears similarity to our signature scheme construction, wherein they build a distribution on which inputs $\bar{x} = (x, \sigma_x)$ contain a signature and the robust classifier has a verification algorithm embedded. Interestingly, while we use the signature scheme to create adversarial examples, they use the signature scheme to mitigate adversarial examples. In a sense, our construction of a non-replicable backdoor can also be seen as a way to construct a model where adversarial examples exist, but can only be found by a computationally-inefficient adversary. Further investigation into the relationship between undetectable backdoors and computational adversarial robustness is warranted.

e) Comparison to [16], [51] [52].: Cohen, Rosenfeld and Kolter [16] and subsequent works (e.g. [51]) used a similar averaging approach to what we use in our immunization, to certify robustness of classification algorithms, under the assumption that the original classifier h satisfies a strong property. They show that if we take an input x such that a small ball around it contains mostly points correctly classified by h, then a random smoothing will give the same classification to x and these close points. There are two important differences between our work and that of [16]. First, by the discussion in Section II-A, as Cohen et al. consider classification and not regression, inherently most input points will not satisfy their condition as we are guaranteed that an adversarial example resides in their neighborhood. Thus, thinking about regression instead of classification is necessary to give strong certification of robustness for *every* point. A subsequent work of Chiang et al. [52] considers randomized smoothing for regression, where the output of the regression hypothesis is unbounded. In our work, we consider regression tasks where the hypothesis image is bounded (e.g. in $[-1, 1]$). In these settings, in contrast to the aforementioned body of work, we no longer need to make any assumptions about the given hypothesis h (except of it being a good predictor). This is completely crucial in our settings as h is the output of a learning algorithm, which we view as malicious and adversarial. Instead, we only make assumptions regarding the ground truth $f^\star$, which is not affected by the learning algorithm.

f) Comparison to [53].: At a high level, Moitra, Mossell and Sandon [53] design methods for a trainer to produce a model that perfectly fits the training data and mislabels everything else, and yet is indistinguishable from one that generalizes well. There are several significant differences between this and our setting.

First, in their setting, the malicious model produces incorrect outputs on *all but a small fraction* of the space. On the other hand, a backdoored model has the same generalization behavior as the original model, but changes its behavior on a sparse subset of the input space. Secondly, their malicious model is an obfuscated program which does not look like a model that natural training algorithms output. In other words, their models are not undetectable in our sense, with respect to natural training algorithms which do not invoke a cryptographic obfuscator. Third, one of our contributions is a way to "immunize" a model to remove backdoors during evaluation time. They do not attempt such an immunization; indeed, with a malicious model that is useless except for the training data, it is unclear how to even attempt immunization.

g) Backdoors in Cryptography.: Backdoors in cryptographic algorithms have been a concern for decades. In a prescient work, Young and Yung [54] formalized cryptographic backdoors and discussed ways that cryptographic techniques can themselves be used to insert backdoors in cryptographic systems, resonating with the high order bits of our work where we use cryptography to insert backdoors in machine learning models. The concern regarding backdoors in (NIST-)standardized cryptosystems was exacerbated in the last decade by the Snowden revelations and the consequent discovery of the DUAL_EC_DRBG backdoor [55].

h) Embedding Cryptography into Neural Networks.: Klivans and Servedio [56] showed how the *decryption algorithm* of a lattice-based encryption scheme [42] (with the secret key hardcoded) can be implemented as an intersection of halfspaces or alternatively as a depth-2 MLP. In contrast, we embed the *public verification key* of a digital signature scheme into a neural network. In a concrete construction using lattice-based digital signature schemes [34], this neural network is a depth-4 network.

References

[1] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems (extended abstract)," in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, R. Sedgewick, Ed. ACM, 1985, pp. 291–304. [Online]. Available: https://doi.org/10.1145/22145.22178 2, 5

[2] A. Rahimi and B. Recht, "Random features for large-scale kernel machines." in *Neural Information Processing Systems*, 2007. 3, 6

[3] J. Bruna, O. Regev, M. J. Song, and Y. Tang, "Continuous LWE," in *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, S. Khuller and V. V. Williams, Eds. ACM, 2021, pp. 694–707. 3, 7, 9

[4] Q. Berthet and P. Rigollet, "Complexity theoretic lower bounds for sparse principal component detection," in *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, ser. JMLR Workshop and Conference Proceedings, S. Shalev-Shwartz and I. Steinwart, Eds., vol. 30. JMLR.org, 2013, pp. 1046–1066. [Online]. Available: http://proceedings.mlr.press/v30/Berthet13.html 3, 7

[5] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 8011–8021. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/280cf18baf4311c92aa5a042336587d3-Abstract.html 3, 4, 5, 7, 8, 9

[6] J. Hayase, W. Kong, R. Somani, and S. Oh, "Spectre: defending against backdoor attacks using robust statistics," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 4129–4139. [Online]. Available: https://proceedings.mlr.press/v139/hayase21a.html 3, 7, 8, 9

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019. [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2909068 3, 4, 9

[8] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *CoRR*, vol. abs/1712.05526, 2017. [Online]. Available: http://arxiv.org/abs/1712.05526 3, 9

[9] Y. Adi, C. Baum, M. Cissé, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, W. Enck and A. P. Felt, Eds. USENIX Association, 2018, pp. 1615–1631. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/adi 3, 4, 9

[10] S. Hong, N. Carlini, and A. Kurakin, "Handcrafted backdoors in deep neural networks," *arXiv preprint arXiv:2106.04690*, 2021. 3, 4, 9

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013. 3, 5

[12] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," *Journal of the ACM (JACM)*, vol. 62, no. 4, pp. 1–64, 2015. 3, 7

[13] O. Reingold, G. N. Rothblum, and R. D. Rothblum, "Constant-round interactive proofs for delegating computation," *SIAM Journal on Computing*, no. 0, pp. STOC16–255, 2019. 3, 7

[14] S. Goldwasser, G. N. Rothblum, J. Shafer, and A. Yehudayoff, "Interactive proofs for verifying machine learning," in *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, ser. LIPIcs, J. R. Lee, Ed., vol. 185. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 41:1–41:19. [Online]. Available: https://doi.org/10.4230/LIPIcs.ITCS.2021.41 3, 5

[15] M. Blum, "Coin flipping by telephone," in *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981*, A. Gersho, Ed. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No 82-04, 1981, pp. 11–15. 3

[16] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1310–1320. 4, 9, 10

[17] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, "Are adversarial examples inevitable?" 2018. 5

[18] D. I. Diochnos, S. Mahloujifar, and M. Mahmoody, "Adversarial risk and robustness: General definitions and implications for the uniform distribution," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 10 380–10 389. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/3483e5ec0489e5c394b028ec4e81f3e1-Abstract.html 5

[19] A. Shamir, I. Safran, E. Ronen, and O. Dunkelman, "A simple explanation for the existence of adversarial examples with small hamming distance," 2019. 5

[20] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *arXiv preprint arXiv:1905.02175*, 2019. 5

[21] A. Shamir, O. Melamed, and O. BenShmuel, "The dimpled manifold model of adversarial examples in machine learning," *arXiv preprint arXiv:2106.10151*, 2021. 5

[22] M. Talagrand, "Concentration of measure and isoperimetric inequalities in product spaces," *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, vol. 81, no. 1, pp. 73–205, 1995. 5

[23] C.-K. Chow, "An optimum character recognition system using decision functions," *IRE Transactions on Electronic Computers*, no. 4, pp. 247–254, 1957. 5

[24] R. L. Rivest and R. H. Sloan, "Learning complicated concepts reliably and usefully." in *AAAI*, 1988, pp. 635–640. 5

[25] J. Kivinen, "Reliable and useful learning with uniform probability distributions," in *ALT*, 1990, pp. 209–222. 5

[26] A. T. Kalai, V. Kanade, and Y. Mansour, "Reliable agnostic learning," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1481–1495, 2012. 5

[27] M. Hopkins, D. M. Kane, and S. Lovett, "The power of comparisons for actively learning linear classifiers," *arXiv preprint arXiv:1907.03816*, 2019. 5

[28] S. Goldwasser, A. T. Kalai, Y. T. Kalai, and O. Montasser, "Beyond perturbations: Learning guarantees with arbitrary adversarial test examples," *arXiv preprint arXiv:2007.05145*, 2020. 5

[29] A. T. Kalai and V. Kanade, "Efficient learning with arbitrary covariate shift," in *Algorithmic Learning Theory*. PMLR, 2021, pp. 850–864. 5

[30] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, D. S. Johnson, Ed. ACM, 1989, pp. 33–43. [Online]. Available: https://doi.org/10.1145/73007.73011 5

[31] J. Rompel, "One-way functions are necessary and sufficient for secure signatures," in *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, H. Ortiz, Ed. ACM, 1990, pp. 387–394. [Online]. Available: https://doi.org/10.1145/100216.100269 5

[32] S. Garg, S. Jha, S. Mahloujifar, and M. Mohammad, "Adversarially robust learning could leverage computational hardness." in *Algorithmic Learning Theory*. PMLR, 2020, pp. 364–385. 5, 9

[33] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of*

Computing, Victoria, British Columbia, Canada, May 17-20, 2008, C. Dwork, Ed. ACM, 2008, pp. 197–206. [Online]. Available: https://doi.org/10.1145/1374376.1374407 6

[34] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, ser. Lecture Notes in Computer Science, H. Gilbert, Ed., vol. 6110. Springer, 2010, pp. 523–552. [Online]. Available: https://doi.org/10.1007/978-3-642-13190-5_27 6, 10

[35] S. Goldwasser and G. N. Rothblum, "On best-possible obfuscation," in *Theory of Cryptography Conference*. Springer, 2007, pp. 194–213. 6

[36] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang, "On the (im) possibility of obfuscating programs," in *Annual international cryptology conference*. Springer, 2001, pp. 1–18. 6

[37] A. Jain, H. Lin, and A. Sahai, "Indistinguishability obfuscation from well-founded assumptions," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021, pp. 60–73. 6

[38] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: replacing minimization with randomization in learning." in *Neural Information Processing Systems*, 2008. 6

[39] ——, "Uniform approximation of functions with random bases," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 555–561. 6

[40] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *International Conference on Learning Representations*, 2018. 7, 9

[41] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5286–5295. 7, 9

[42] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, H. N. Gabow and R. Fagin, Eds. ACM, 2005, pp. 84–93. [Online]. Available: https://doi.org/10.1145/1060590.1060603 7, 10

[43] C. Peikert, "A decade of lattice cryptography," *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 4, pp. 283–424, 2016. [Online]. Available: https://doi.org/10.1561/0400000074 7

[44] M. Brennan and G. Bresler, "Optimal average-case reductions to sparse pca: From weak assumptions to strong hardness," in *Conference on Learning Theory*. PMLR, 2019, pp. 469–470. 7

[45] G. De Palma, B. Kiani, and S. Lloyd, "Adversarial robustness guarantees for random deep neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2522–2534. 8

[46] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*. IEEE, 2019, pp. 707–723. [Online]. Available: https://doi.org/10.1109/SP.2019.00031 8

[47] I. Diakonikolas, D. M. Kane, and A. Stewart, "Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2017, pp. 73–84. 8

[48] S. Bubeck, Y. T. Lee, E. Price, and I. Razenshteyn, "Adversarial examples from computational constraints," in *International Conference on Machine Learning*. PMLR, 2019, pp. 831–840. 8, 9

[49] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *arXiv preprint arXiv:1904.12843*, 2019. 9

[50] M. Shafieinejad, J. Wang, N. Lukas, and F. Kerschbaum, "On the robustness of the backdoor-based watermarking in deep neural networks," *CoRR*, vol. abs/1906.07745, 2019. [Online]. Available: http://arxiv.org/abs/1906.07745 9

[51] H. Salman, J. Li, I. Razenshteyn, P. Zhang, H. Zhang, S. Bubeck, and G. Yang, "Provably robust deep learning via adversarially trained smoothed classifiers," *Advances in Neural Information Processing Systems*, vol. 32, 2019. 10

[52] P. Chiang, M. J. Curry, A. Abdelkader, A. Kumar, J. Dickerson, and T. Goldstein, "Detection as regression: Certified object detection with median smoothing," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/0dd1bc593a91620daecf7723d2235624-Abstract.html 10

[53] A. Moitra, E. Mossel, and C. Sandon, "Spoofing generalization: When can't you trust proprietary models?" *CoRR*, vol. abs/2106.08393, 2021. [Online]. Available: https://arxiv.org/abs/2106.08393 10

[54] A. L. Young and M. Yung, "Kleptography: Using cryptography against cryptography," in *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, ser. Lecture Notes in Computer Science, W. Fumy, Ed., vol. 1233. Springer, 1997, pp. 62–74. [Online]. Available: https://doi.org/10.1007/3-540-69053-0_6 10

[55] D. Shumow and N. Ferguson, "On the possibility of a back door in the nist sp800-90 dual ec prng," 2007. [Online]. Available: https://rump2007.cr.yp.to/15-shumow.pdf 10

[56] A. R. Klivans and A. A. Sherstov, "Cryptographic hardness for learning intersections of halfspaces," in *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*. IEEE Computer Society, 2006, pp. 553–562. [Online]. Available: https://doi.org/10.1109/FOCS.2006.24 10