A Separation Result Between Data-oblivious and Data-aware Poisoning Attacks

Samuel Deng
Columbia University
samdeng@cs.columbia.edu

Somesh Jha
University of Wisconsin
jha@cs.wisc.edu

Sanjam Garg
UC Berkeley and NTT Research
sanjamg@berkeley.edu

Saeed Mahloujifar
Princeton
sfar@princeton.edu

Mohammad Mahmoody
University of Virginia
mohammad@virginia.edu

Abhradeep Thakurta
Google Research - Brain Team
athakurta@google.com

Abstract

Poisoning attacks have emerged as a significant security threat to machine learning algorithms. It has been demonstrated that adversaries who make small changes to the training set, such as adding specially crafted data points, can hurt the performance of the output model. Most of these attacks require the full knowledge of the training data. This leaves open the possibility of achieving the same attack results using poisoning attacks that do not have the full knowledge of the clean training set. In this work, we initiate a theoretical study of the problem above. Specifically, for the case of feature selection with LASSO, we show that *full information* adversaries (that craft poisoning examples based on the rest of the training data) are provably much more devastating compared to the optimal attacker that is *oblivious* to the training set yet has access to the distribution of the data. Our separation result shows that the two setting of data-aware and data-oblivious are fundamentally different and we cannot hope to achieve the same attack or defense results in these scenarios.

1 Introduction

Traditional approaches to supervised machine learning focus on a benign setting where honestly sampled training data is given to a learner. However, the broad use of these learning algorithms in safety-critical applications makes them targets for sophisticated attackers. Consequently, machine learning has gone through a revolution of studying the same problem, but this time under so-called adversarial settings. Researchers have investigated several types of attacks, including test-time (a.k.a., evasion attacks to find adversarial examples) [65, 6, 34, 58], training-time attacks (a.k.a., poisoning or causative attacks) [3, 8, 54], backdoor attacks [70, 35], membership inference attacks [60], etc. In response, other works have put forth several defenses [55, 46, 9] followed by adaptive attacks [15, 2, 68] that circumvent some of the proposed defenses. Thus, developing approaches that are based on solid theoretical foundations (that prevent further adaptive attacks) has stood out as an important area of investigation.

Poisoning Attacks. In a poisoning attack, an adversary changes a training set $\mathcal{S}$ of examples into a "close" training set $\mathcal{S}'$ (The difference is usually measured by Hamming distance; i.e., the number of examples injected and/or removed.). Through these changes, the goal of the adversary, generally speaking, is to degrade the "quality" of the learned model, where quality here could be interpreted in different ways. In a recent industrial survey [42], poisoning attacks were identified as the most important threat model against applications of machine learning. The main reason behind the importance of poisoning attacks are the feasibility of performing the attack for adversary. As the

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

data is usually gathered from multiple sources, the adversary can perform the poisoning attacks by corrupting one of the sources. Hence, it is extremely important to fundamentally understand this threat model. In particular, we need to investigate the role of design choices that are made in both poisoning attacks and defenses.

Does the attacker know the training data? The role of knowledge of the clean training set is one of the less investigated aspects of poisoning attacks. Many previous work on theoretical analysis of poisoning attacks implicitly, or explicitly, assume that the adversary has full knowledge of the training data $\mathcal{S}$ before choosing what examples to add or delete from $\mathcal{S}$ [41, 61, 47, 64]. In several natural scenarios, an adversary might not have access to the training data before deciding on how to tamper with it. This has led researchers to study poisoning attacks that do not use the knowledge of the training set to craft the poison points. In this work, we explore the following question:

> *What is the role of the knowledge of training set in the success of poisoning adversaries? Can the knowledge of training set help the attacks? Or alternatively, can hiding the training set from adversaries help the defenses?* [1]

In this work, as a first step to understand this question, we show a separation result between data-oblivious and data-aware poisoning adversaries. In particular, we show that there exist a learning setting (Feature selection with LASSO on Gaussian data) where poisoning adversaries that know the distribution of data but are oblivious to specific training samples that are used to train the model are provably weaker than the adversaries with the knowledge of both training set and the distribution. To the best of our knowledge, this is the first separation result for poisoning attacks.

Implications of our separation result: Here, we mention some implications of our separation result.

- **Separation of threat models:** The first implication of our result is the separation of data-oblivious and data-aware poisoning threat models. Our result shows that data-oblivious attacks are strictly weaker than data-aware attacks. In other words, it shows that we cannot expect the defenses to have the same effectiveness in both scenarios. This makes the knowledge of data a very important design choice that should be clearly stated when designing defenses or attacks.

- **Possibility of designing new defenses:** Although data-oblivious poisoning is a weaker attack model, it might still be the right threat model for many applications. For instance, if data providers use cryptographically secure multi-party protocols to train the model [71], then each participant can only observe their own data. Note that each party might still have access to some data pool from the true distribution of training set and that still fits in our data-oblivious threat model. In these scenarios, it is natural to use defenses that are only secure against data-oblivious attacks. Our results shows the possibility of designing defense mechanisms that leverage the secrecy of training data and can provide much stronger security guarantees in this threat mode. In particular, our result shows the provable robustness of LASSO algorithm in defending against data-oblivious attacks.

  Note that this approach is distinct from the demoted notion of "security through obscurity" as the attacker knows every detail of the algorithm as well as the data distribution. The only unknown to the adversary is the randomness involved in the process of sampling training examples from the training distribution. This is exactly similar to how secret randomness helps security in cryptography.

- **A new motive for privacy:** privacy is often viewed as a utility for data owners in the machine learning pipeline. Due to the trade-offs between privacy and the efficiency/utility, data-users often ignore the privacy of data owners while doing their analysis, especially when there is no incentive to enforce the privacy of the learning protocol. The possibility of improving the security against poisoning attacks by enforcing the (partial) data-obliviousness of the adversary could create a new incentive for keeping training datasets secret. Specifically, the users of data would now have more motivation to try to keep training dataset private, with the goal of securing their models against poisoning and increasing their utility in scenarios where part of data is coming from potentially malicious sources.

[1] This question was independently asked as an open question in the concurrent survey of Goldblum et al. [31].

1.1 Our Contributions

In this work, we provide theoretical evidence that obliviousness of attackers to the training data can indeed help robustness against poisoning attacks. In particular, we provide a provable difference between: (i) an adversary that is aware of the training data as well as the distribution of training data, before launching the attack (data-aware adversary) and (ii) an adversary that only knows the distribution of training data and does not know the specific clean examples in the training set (data-oblivious adversary).

We start by formalizing what it means mathematically for the poisoning adversary to be data-oblivious or data-aware.

Separations for feature selection with Lasso. We then prove a separation theorem between the data-aware and data-oblivious poisoning threat models in the context of *feature selection*. We study data-aware and data-oblivious attackers against the Lasso estimator and show that if certain natural properties holds for the distribution of dataset, the power of optimal data-aware and data-oblivious poisoning adversaries differ significantly.

We emphasize that in our data-oblivious setting, the adversary *fully knows* the *data distribution*, and hence it implicitly has access to a lot of auxiliary information about the data set, yet the very fact that it does not know the *actual* sampled dataset makes it harder for adversary to achieve its goal.

Experiments. To further investigate the power of data-oblivious and data-aware attacks in the context of feature selection, we experiment on synthetic datasets sampled from Gaussian distributions, as suggested in our theoretical results. Our experiments confirm our theoretical findings by showing that the power of data-oblivious and poisoning attacks differ significantly. Furthermore, we experimentally evaluate the power of *partially-aware* attackers who only know part of the data. These experiments show the gradual improvement of the attack as the knowledge of data grows.

In our experimental studies we go beyond Gaussian setting and show that the the power of data-oblivious attacks could be significantly lower on real world distributions as well. In our experiments, sometimes (depending on the noise nature of the dataset), even an attacker that knows 20% of the dataset cannot have much of improvement over an oblivious attacker.

1.2 Related Work

Here, we provide a short version of related prior work. A more comprehensive description of previous work has been provided in Appendix A where we also categorize the existing attacks into data-aware and data-oblivious categories.

[4] Study "Blind" attackers against machine learning models that do not even know the distribution of the data. They show that poisoning attacks could be successful in such a restricted setting by studying the minimax risk of learners. They also introduce a "informed" attacks that see the data distribution, but not the actual training samples and leave the study of these attacks to future work. Interestingly, the "informed" setting of [4] is equivalent to the "oblivious" setting in our work.

Xiao et al. [74] empirically examine the robustness of feature selection in the context of poisoning attacks, but their measure of stability is across sets of features. We are distinct in that our paper studies the effect of data-oblivious attacks on *individual* features and with provable guarantees.

We distinguish our work with another line of work that studies the computational complexity of the attacker [49, 29]. Here, we study the "information complexity" of the attack; namely, what information the attacker needs to succeed in a poisoning attack, while those works study the *computational resources* that a poisoning attacker needs to successfully degrade the quality of the learned model. Another recent exciting line of work that studies the computational aspect of robust learning in poisoning contexts, focuses on the computational complexity of the *learning* process itself [18, 43, 16, 20, 21, 19, 56, 22], and other works have studied the same question about the complexity of the learning process for evasion attacks [11, 10, 17]. Furthermore, our work deals with information complexity and is distinct from works that study the impact of the training set (e.g., using clean labels) on the success of poisoning [58, 76, 62, 70].

Our work's motivation for data secrecy might seem similar to other works that leverage privacy-preserving learning (and in particular differential privacy [23, 26, 25]) to limit the power of poisoning attacks by making the learning process less sensitive to poison data [45]. However, despite seeming

similarity, what we pursue here is fundamentally different. In this work, we try to understand the effect of keeping the data secret from adversaries. Whereas the robustness guarantees that come from differential privacy has nothing to do with secrecy and hold even if the adversary gets to see the full training set (or even select the whole training set in an adversarial way.).

We also point out some separation results in the context of adversarial examples. The work of Bubeck et al. [12] studies the separation in the power of *computationally bounded* v.s. *computationally unbounded* learning algorithms in learning robust model. Tsipras et al. [69] studies the separation between *benign accuracy* and *robust accuracy* of classifiers showing that they can be even at odds with each other. Schmidt et al. [57] show the separation between sample complexity of learning algorithms in training an adversarially robust model versus a model with high benign accuracy. Garg et al. [29] separate the notions of *computationally bounded* v.s. *computationally unbounded* attacks in successfully generating adversarial examples. Although all these results are only proven for few (perhaps unrealistic) settings, they still significantly helped the understanding of adversarial examples.

As opposed to the data poisoning setting, the question of adversary's (adaptive) knowledge was indeed previously studied in the line of work on adversarial examples [44, 52, 65]. In a test time evasion attack the adversary's goal is to find an adversarial example, the adversary knows the input x *entirely* before trying to find a close input x' that is misclassified. So, this adaptivity aspect already differentiates adversarial examples from random noise.

2 Defining Threat Models: Data-oblivious and Data-aware Poisoning

In this section, we formally define the security games of learning systems under *data-oblivious* poisoning attacks. It is common in cryptography to define security model based on a game between an adversary and a challenger [39]. Here, we use the same approach and introduce game based definitions for data-oblivious and data-aware adversaries.

Feature selection. The focus of this work is mostly on the feature selection which is a significant task in machine learning. In a feature selection problem, the learning algorithm wants to discover the relevant features that determine the ground truth function. For example, imagine a dataset of patients with many features, who suffer from a specific disease with different levels of severity. One can try to find the most important features contributing to the severity of the disease in the context of feature selection. Specifically, the learners' goal is to recover a vector $\theta^* \in \mathbb{R}^d$ whose non-zero coordinates determine the relevant features contributing to the disease. In this scenario, the goal of the adversary is to deceit the learning process and make it output a model $\hat{\theta}' \in \mathbb{R}^d$ with a different set of non-zero coordinates. As motivation for studying feature selection under adversarial perturbations, note that the non-zero coordinates of the learned model could be related to a sensitive subject. For example, in the patient data example described in the introduction, the adversary might be a pharmaceutical institute who tries to imply that a non-relevant feature is contributing to the disease, in order to advertise for a specific medicine.

We start by separating the *goal* of a poisoning attack from *how* the adversary achieves the goal. The setting of an *data-oblivious* attack deals with the latter, namely it is about how the attack is done, and this aspect is orthogonal to the goal of the attack. In a nutshell, many previous works on data poisoning deal with increasing the population risk of the produced model (see Definition F.1 below and Section B for more details and variants of such attacks). In a different line of work, when the goal of the learning process is to recover a set of features (a.k.a., model recovery) the goal of an attacker would be defined to counter the goal of the feature selection, namely to add or remove features from the correct model.

In what follows, we describe the security games for a feature selection task. We give this definition for a basic reference setting in which the data-oblivious attacker injects data into the data set, and its goal is to change the selected features. (See Section B for more variants of the attack.) Later, in Section 3 we will see how to construct problem instances (by defining their data distributions) that provably separate the power of data-oblivious attacks from data-aware ones.

Notation. We first define some useful notation. For an arbitrary vector $\theta \in \mathbb{R}^d$ we use $\mathrm{Supp}(\theta) = \{i \colon \theta_i \neq 0\}$, we denote the set of (indices of) its non-zero coordinates. We use capital letters (e.g X) to denote sets and calligraphic letters (e.g. $\mathcal{X}$) to denote distributions. $(\mathcal{X}, \mathcal{Y})$ denotes the joint distribution of $\mathcal{X}$ and $\mathcal{Y}$ and $\mathcal{X}_1 \equiv \mathcal{X}_2$ denotes the equivalence of two distributions $\mathcal{X}_1$ and

$\mathcal{X}_2$. We use $\|\theta\|_2$ and $\|\theta\|$ to denote the ℓ_2 and ℓ_1 norms of θ respectively. For two matrices $X \in R^{n\times d}$ and $Y \in R^{n\times 1}$, we use $[X \,|\, Y] \in R^{n\times(d+1)}$ to denote a set of n regression observations on feature vectors $X_{i\in[n]}$ such that Y_i is the real-valued observation for X_i. For two matrices $X_1 \in \mathbb{R}^{n_1\times d}$ and $X_2 \in \mathbb{R}^{n_2\times d}$, we use $\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times d}$ to denote the concatenation of X_1 and X_2. Similarly, for two set of observations $[X_1 \,|\, Y_1] \in \mathbb{R}^{n_1\times(d+1)}$ and $[X_2 \,|\, Y_2] \in \mathbb{R}^{n_2\times(d+1)}$, we use $\left[\begin{array}{c|c} X_1 & Y_1 \\ X_2 & Y_2 \end{array}\right] \in \mathbb{R}^{(n_1+n_2)\times(d+1)}$ to denote the concatenation of $[X_1 \,|\, Y_1]$ and $[X_2 \,|\, Y_2]$. For a security game G and an adversary A we use $\mathsf{Adv}(A, G)$ (advantage of adversary A in game G) to denote probability of adversary A winning the security game G, where the probability is taken over the randomness of the game and adversary.

Since the security games for data-aware and data-oblivious games are close, we use Definition 2.1 below for both, while we specify their exact differences.

Definition 2.1 (Data-oblivious and data-aware data injection poisoning for feature selection)**.** *We first describe the* data-oblivious *security game between a challenger C and an adversary A. The game is parameterized by the adversary's budget k and the training data $\mathcal{S} = [X \,|\, Y]$ which is a matrix X and a set of labels Y, and the feature selection algorithm* $\mathsf{FtrSelector}$.

$\mathbf{OblFtrSel}(k, \mathcal{D}, \mathsf{FtrSelector}, n)$.

1. *Knowing the algorithm* $\mathsf{FtrSelector}$ *and distribution $\mathcal{D}$ supported on $\mathbb{R}^{d+1}$, and given k as input, the adversary A generates a poisoning dataset $[X' \,|\, Y'] \in [-1, 1]^{k\times(d+1)}$ of size k such that each row has ℓ_2 norm at most 1 and sends it to C.*
2. *C samples a dataset $[X \,|\, Y] \leftarrow \mathcal{D}^n$*
3. *C recovers models $\hat{\theta} = \mathsf{FtrSelector}([X \,|\, Y])$ using the clean data and $\hat{\theta}' = \mathsf{FtrSelector}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)$ using the poisoned data.*
4. *Adversary wins if $\mathrm{Supp}(\hat{\theta}) \neq \mathrm{Supp}(\hat{\theta}')$, and we use the following notation to denote the winning:*

$$\mathbf{OblFtrSel}(A, k, \mathcal{D}, \mathsf{FtrSelector}, n) = 1.$$

In the security game for data-aware *attackers, all the steps are the same as above, except that the order of steps 1 and 2 are different. Namely, challenger first samples and sends the dataset to adversary.*

$\mathbf{AwrFtrSel}(k, \mathcal{D}, \mathsf{FtrSelector}, n)$.

1. *C samples $[X \,|\, Y] \leftarrow \mathcal{D}^n$ and sends it A.*
2. *Knowing the algorithm* $\mathsf{FtrSelector}$ *and distribution $\mathcal{D}$ supported on $\mathbb{R}^{d+1}$, the dataset $[X \,|\, Y]$, and given k as input, the adversary A generates a poisoning dataset $[X' \,|\, Y'] \in [-1, 1]^{k\times(d+1)}$ of size k such that each row $[X' \,|\, Y']$ has ℓ_2 norm at most 1 and sends it to C.*
3. *C recovers models $\hat{\theta} = \mathsf{FtrSelector}([X \,|\, Y])$ using the clean data and $\hat{\theta}' = \mathsf{FtrSelector}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)$ using the poisoned data.*
4. *Adversary wins if $\mathrm{Supp}(\hat{\theta}) \neq \mathrm{Supp}(\hat{\theta}')$, and we use the following notation to denote the winning:*

$$\mathbf{AwrFtrSel}(A, k, \mathcal{D}, \mathsf{FtrSelector}, n) = 1.$$

Variations of security games for Definition 2.1. Definition 2.1 is written only for the case of feature-flipping attacks by only injecting poison data. One can, however, envision variants by changing the adversary's goal and how it is doing the poisoning attack. In particular, one can define more specific goals for the attacker to violate the feature selection, by aiming to add or remove non-zero coordinates to the recovered model compared to the ground truth.[2] In addition, it is also possible to change the method of the adversary to employ data elimination or substitution attacks.

[2] In fact, one can even define *targeted* variants in which the adversary even picks the feature that it wants to add/remove or flip.

One can also imagine *partial-information* attackers who are exposed to a fraction of the data set $\mathcal{S}$ (e.g., by being offered the knowledge of a randomly selected p fraction of the rows of $[X|Y]$. Our experiments deal with this very setting.

Why bounding the ℓ_2 norm of the poison points? When bounding the number of poison points, it is important to bound the norm of the poisoning points according to some threshold (e.g. through a clipping operation) otherwise a single poison point can have infinitely large effect on the trained model. By bounding the ℓ_2 norm of the poison data, we make sure that a single poison point has a bounded effect on the objective function and cannot play the role of a large dataset. We could remove this constraint from the security game and enforce it in the algorithm through a clipping operation but we keep it as a part of definition to emphasize on this aspect of the security game. Note that in this work we always assume that the data is centered around zero and normalized. That is why we only use a constraint on the norm of the poison data points. However, the security game could be generalized by replacing the ℓ_2 norm constraint with an arbitrary filter F for different scenarios.

Why using $\hat{\theta}$ instead of θ. Note that in security games of Definition 2.1 we do not use the *real* model θ (or more accurately its set of features $\mathrm{Supp}(\theta)$), but rather we work with $\mathrm{Supp}(\hat{\theta})$. That is because, we will work with promised data sets for which FtrSelector provably recovers the true set of features $\mathrm{Supp}(\hat{\theta}) = \mathrm{Supp}(\theta)$. This could be guaranteed, e.g., by putting conditions on the data.

Why injecting the poison data to the end? Note that in security games of Definition 2.1, we are simply injecting the poison examples to the *end* of the training sequence defined by X, Y, instead of asking the adversary to pick their locations. That is only for simplicity, and the definition is implicitly assuming that the feature selection algorithm is symmetric with respect to the order of the elements int the data set (e.g., this is so for Lasso estimator). However, one can generalize the definition directly to allow the adversary to pick the specific location of the added elements.

3 Separating the Power of Data-oblivious and Data-aware Attacks

In this section, we provably demonstrate that the power of data-oblivious and data-aware adversaries could significantly differ. Specifically, we study the power of poisoning attacks on feature selection.

Feature selection by the Lasso estimator. We work in the feature selection setting, and the exact format of our problem is as follows. There is a target parameter vector $\theta^* \in (0,1)^d$. We have a $n \times d$ matrix X (n vectors, each of d features) and we have $Y = X \times \theta^* + W$ where W itself is a small noise, and Y is the vector of noisy observations about θ^*, where the number of non-zero elements (denoting the actual relevant features) in θ^* is bounded by s namely, $|\mathrm{Supp}(\theta^*)| \leq s$. The goal of the feature selection is to find a model $\hat{\theta}$, given $[X \,|\, Y]$, such that $\mathrm{Supp}(\hat{\theta}) = \mathrm{Supp}(\theta^*)$.

The Lasso Estimator tries to learn θ^* by optimizing the regularized loss with regularization parameter λ and obtain the solution $\hat{\theta}_\lambda$ as

$$\hat{\theta}_\lambda = \underset{\theta \in (0,1)^d}{\mathrm{argmin}} \frac{1}{n} \cdot \|Y - X \times \theta\|_2^2 + \frac{2\lambda}{n} \cdot \|\theta\|_1.$$

We use $\mathsf{Lasso}([X \,|\, Y], \lambda)$ to denote $\hat{\theta}_\lambda$, as learned by the Lasso optimization described above. When we λ is clear from the context, we use $\mathsf{Lasso}([X \,|\, Y])$ and $\hat{\theta}$. We also use $\mathrm{Risk}(\hat{\theta}, [X \,|\, Y], \lambda)$ (and $\mathrm{Risk}(\hat{\theta}, [X \,|\, Y], \lambda)$ when λ is clear from the context) to denote the "scaled up" value of the Lasso's objective function

$$\mathrm{Risk}(\hat{\theta}, [X \,|\, Y]) = \left\| Y - X \times \hat{\theta} \right\|_2^2 + 2 \cdot \lambda \cdot \left\| \hat{\theta} \right\|_1.$$

It is known by a work of Wainwright [72] that under proper conditions Lasso estimator can recover the correct feature vector (See Theorems C.2 and C.4 in Appendix C for more details.) The robust version of this result, where part of the training data is chosen by an adversary, is also studied in Thakurta et al. [66]. (See Theorems C.5 and C.3 in Appendix C for more details.) However, the robust version considers robustness against data-aware adversaries that can see the dataset and select the poisoning points based on the rest of training data. In the following theorem, we show that the robustness against data-oblivious adversaries could be much higher than robustness against data-aware adversaries.

Separation for feature selection. We prove the existence of a feature selection problem such that, with high probability, it stays secure in the data-oblivious attack model of Definition 2.1, while the same problem's setting is highly vulnerable to poisoning adversaries as defined in the data-aware threat model of Definition 2.1. We use Lasso estimator for proving our separation result.

Theorem 3.1. *For any $k \in \mathbb{N}$ and $\varepsilon_1 < \varepsilon_2 \in (0,1)$, there exist an $n, d \in \mathbb{N}$, $\sigma \in \mathbb{R}$ and $\theta^* \in \mathbb{R}^d$ such that the distribution $\mathcal{D} \equiv (\mathcal{X}, \mathcal{Y})$ for $\mathcal{X} \equiv \mathcal{N}(0, \sigma^2)^{n \times d}$ and $\mathcal{Y} \equiv X \times \theta^* + \mathcal{N}(0, 1/4)$ is recoverable using Lasso estimator, meaning that with high probability over the randomness of sampling a dataset $[X \,|\, Y] \leftarrow \mathcal{D}^n$ we have*

$$\mathrm{Supp}(\mathsf{Lasso}([X \,|\, Y]) = \mathrm{Supp}(\theta^*),$$

while the advantage of any data-oblivious adversary in changing the support set is at most ε_1. Namely for any data-oblivious adversary A we have

$$\mathop{\mathbb{E}}_{\mathcal{S} \leftarrow D} \left[\mathbf{OblFtrSel}(A, k, \mathcal{D}, \mathsf{Lasso}, n) \right] \leq \varepsilon_1$$

On the other hand, there is an adversary that can win the data-aware security game with probability at least ε_2. Namely, there is an data-aware adversary A such that

$$\mathop{\mathbb{E}}_{\mathcal{S} \leftarrow D} \left[\mathbf{AwrFtrSel}(A, k, \mathcal{D}, \mathsf{Lasso}, n) \right] \geq \varepsilon_2.$$

The main idea behind the proof. To prove the separation, we use the fact that data-oblivious adversaries cannot discriminate between the coordinates that are not in the support set of θ^*. Imagine the distribution of data has a property that with high probability there exists a unique feature that is not in the support set, but it is possible to add that feature to the support set with a few number of poisoning examples. We call such a feature an "unstable" feature. Suppose the distribution also has an additional property that each coordinate has the same probability of being the unstable feature. Then, the only way that adversary can find the unstable feature is by looking into the dataset. Otherwise, if the adversary is data-oblivious, it does not have any information about the unstable feature and should attack blindly and pick one of the coordinates at random. On the other hand, the data-aware adversary can investigate the dataset and find the unstable feature. In the rest of this section we formalize this idea by constructing a distribution D that has the properties mentioned above.

Below we first define the notion of stable and unstable features and then formally define two properties for a distribution $\mathcal{D}$ that if satisfied, we derive Theorem 3.1 for it.

Definition 3.2 (Stable and unstable coordinates)**.** *Consider a dataset $[X \,|\, Y] \in \mathbb{R}^{n \times (d+1)}$ with a unique solution $\hat{\theta}_\lambda$ for the Lasso minimization. $[X \,|\, Y]$ is k-unstable on coordinate $i \in [d]$ if the ith coordinate of the feature vector obtained by running Lasso on $[X \,|\, Y]$ is 0, namely $\mathsf{Lasso}\left([X \,|\, Y]\right)_i = 0$, and there exist a data set $[X' \,|\, Y']$ with size k and ℓ_∞ norm at most 1 on each row such that $i \in \mathrm{Supp}\left(\mathsf{Lasso}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)\right)$. On the other hand, $[X \,|\, Y]$ is k-stable on a coordinate i, if for all datasets $[X' \,|\, Y']$ with k rows and ℓ_∞ norm at most 1 on each row we have*

$$\mathrm{Sign}(\mathsf{Lasso}\left([X \,|\, Y]\right)_i) = \mathrm{Sign}\left(\mathsf{Lasso}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)_i\right).$$

The following definitions capture two properties of a distribution D. The first property states that with high probability over the randomness of D, a dataset sampled from D has at least one unstable feature.

Definition 3.3 ((k, δ)-unstable distributions)**.** *A distribution D is (k, ε_2)-unstable if it is k-unstable on at least one coordinate with probability at least (ε_2). Namely*

$$\Pr_{S \leftarrow D}[\exists\, i \in [d] : \text{ The } i^{\text{th}} \text{ feature is } k\text{-unstable on } S] \geq \varepsilon_2.$$

The following notion defines the resilience of a distribution against a single poison dataset. In a nutshell, a distribution is resilient if there does not exist a universal poisoning set that can be effective against all the datasets coming from that distribution.

Definition 3.4. *[(k, ε)-resilience] A distribution D over $\mathbb{R}^{n\times(d+1)}$ is (k, ε)-resilient if for any poisoning dataset $\mathcal{S}'$ of size k and ℓ_∞ norm bounded by 1 we have*

$$\Pr_{\mathcal{S}\leftarrow D}[\text{Supp}\left(\mathsf{Lasso}\left(\begin{bmatrix}\mathcal{S}\\ \mathcal{S}'\end{bmatrix}\right)\right) \neq \text{Supp}(\mathsf{Lasso}(S))] \leq \varepsilon.$$

Remark 3.5. *Note that Definitions 3.3 and 3.4 have an implicit dependence on n, the size of the dataset sampled from the distribution that we omit from the notation for simplicity.*

Before constructing a distribution D we first prove the following Proposition about (k, δ)-unstable and (k, ε)-resilient distributions. The proof can be found in Appendix D

Proposition 3.6 (Separation for unstable yet resilient distributions)**.** *If a data distribution is (k, ε_1)-resilient and (k, ε_2)-unstable, then there is an adversary that wins the data-aware game of definition 2.1 with probability ε_2, while no adversary can win the data-oblivious game with probability more than ε_1.*

3.1 (Un)Stability and Resilience of Gaussian

The only thing that remains to prove Theorem 3.1 is to show that Gaussian distributions with proper parameters are (k, ε_2)-unstable and (k, ε_1)-resilient at the same time. Here we sketch the two steps we take to prove this.

Gaussian is Unstable. We first show that each feature in the Gaussian sampling process has a probability of being k-unstable that is proportional to $e^{\lambda-k}$. Note that the unstability of i-th feature is independent from all other features and also note that the probability is independent of d. This shows that, if d is chosen large enough, with high probability, there will be at least one coordinate that is k-unstable. However, note that the probability of a particular feature being unstable is still low and we are only leveraging the large dimensionality to increase the chance of having an unstable feature. Roughly, if we select $d = \omega(\varepsilon_2/\varepsilon_1)$, we can make sure that the ratio of the success rate between data-aware and data-oblivious adversary is what we need. The only thing that remains is to select n, λ and σ in a way that the data oblivious adversary has success rate of at most ε_1 and at least $\Omega(\varepsilon_1)$.

This result actually shows the tightness of the robustness theorem in [66] (See Theorem C.3 for the full description of this result). The authors in [66] show that running Lasso on Gaussian distribution can recover the correct support set, and is even robust to a certain number of adversarial entries. Our result complements theirs and shows that their theorem is indeed tight. Note that the robustness result of [66] is against dataset-aware attacks. In the next step, we show a stronger robustness guarantee for data-oblivious attacks in order to prove our separation result. See Appendix D for a formalization of this argument.

Gaussian is Resilient. We show the LASSO is resilient when applied on Gaussian of any dimension. In particular, we show that if the adversary aims at adding a feature to the support set of the model, it should "invest" in that feature meaning that the l_2 weight on that feature should be high across all the poison entries. The bound on the l_2 norm of each entry will prevent the adversary to invest on all features and therefore, the adversary has to predict which features will be unstable and invest in them. On the other hand, since Gaussian is symmetric, each feature has the same probability of being unstable and the adversary will have a small chance of succeeding. In a nutshell, by selecting $\lambda = \Omega(k + \sigma\sqrt{(n+k)\ln(1/\varepsilon_1)})$ we can make sure that the success probability of the oblivious adversary is bounded by ε_1. This argument is formalized in Appendix D.

4 Experiments

In this section, we highlight our experimental findings on both synthetic and real data to compare the power of data-oblivious and data-aware poisoning attacks in the context of feature selection. Our experiments empirically support our separation result in Theorem 3.1.

Our partial-knowledge attack: The attack first explores through the part of data that it has access to and identifies which feature is the most unstable feature. The key here is that the data-aware adversary can search for the most vulnerable feature in the available data. Then, the attack will use that feature to craft poison points that create maximum correlation between that feature and the response variable.

This attack is optimal for both data-aware and data-oblivious adversaries. See Appendix D.2 for how this is exactly done.

We provide experiments on Gaussian synthetic data and experiments with real datasets. In both sets of experiments, we simulate the power of an adversary that is completely oblivious to the dataset all the way up to a full-information adversary. We change the power of partial information adversaries that only have $p\%$ information about the dataset at regular intervals in between 0% and 100%.

Experiments with Gaussian distribution. For the synthetic experiment, we demonstrate the separation result occurs for a large dataset sampled from a Gaussian distribution. For $n = 300$ rows and $d = 5 \times 10^5$ features, we demonstrate that unstable features occur for a dataset drawn from $\mathcal{N}(0,1)^{n \times d}$. For the LASSO algorithm, we use the hyperparameter of $\lambda = 2\sigma\sqrt{n \log p}$. We vary the "knowledge" the adversary has of the dataset from $p = 0, 5, 10, \ldots 95, 100\%$ by only showing the adversary a random sample of $p\%$ (for $p = 0$, the adversary is completely oblivious and so must choose a feature uniformly at random). The adversary then chooses the most unstable feature out of their $p\%$ of the data and perform the attack on that feature to add it to the $\mathrm{Supp}(\hat{\theta})$. We observe a clear separation between data-oblivious, data-aware, and partially-aware adversaries in Figure 1.

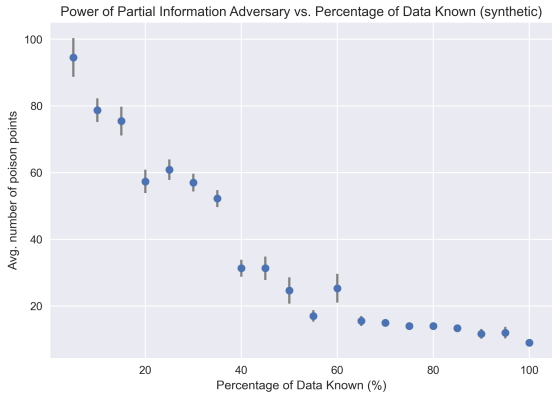

Figure 1: *Synthetic experiment.* The y-axis is the average (over 30 random $p\%$ splits of the dataset given to the adversary) number of poison points needed to add the feature to $\hat{\theta}$.The leftmost point shows the power of an oblivious adversary while the rightmost point shows the power of a full-information adversary. The oblivious adversary needs significantly more poison points, on average, to add their uniformly chosen feature to $\mathrm{Supp}(\hat{\theta})$.

Experiments with real data. We also consider MNIST and four other datasets used widely in the feature selection literature to explore this separation in real world data: Boston, TOX, Protate_GE, and SMK. [3] For full descriptions of each dataset, we refer the reader to Appendix E.2.

We first preprocess the data by standardizing to zero mean and unit variance. Then, we chose λ such that the resulting parameter vector $\hat{\theta}$ has a reasonable support size (at least 10 features in the support); this was done by searching over the space of $\lambda/n \in [0, 1.0]$, and resulted in $\lambda = 50.1$ for Boston, $\lambda = 9.35$ for SMK, $\lambda = 17$ for TOX, $\lambda = 5.1$ for Prostate, and $\lambda = 1000$ for MNIST. Just as in the synthetic experiments, we allow the adversary to have the knowledge of $p = 0, 5, 10, \ldots, 95, 100\%$ fraction of the data. Denote the features *not* in $\mathrm{Supp}(\hat{\theta})$ as $\mathcal{G}$. We attack each feature $i \in \mathcal{G}$ with the same attack as our synthetic experiment, where $X' \in \mathbb{R}^{k \times d}$ and $Y' \in \mathbb{R}^{k \times 1}$. We plot the average best value of k needed by the adversary to add a feature to $\mathrm{Supp}(\hat{\theta})$ against how much knowledge ($p\%$) of the dataset they have. We show the results for SMK and TOX in Figure 2 and the result for MNIST in Figure 3. The rest are in Appendix E.2.

[3] TOX, SMK, and Prostate_GE can be found here: `http://featureselection.asu.edu/datasets.php`. Boston can be found with scikit-learn's built-in datasets: `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html`

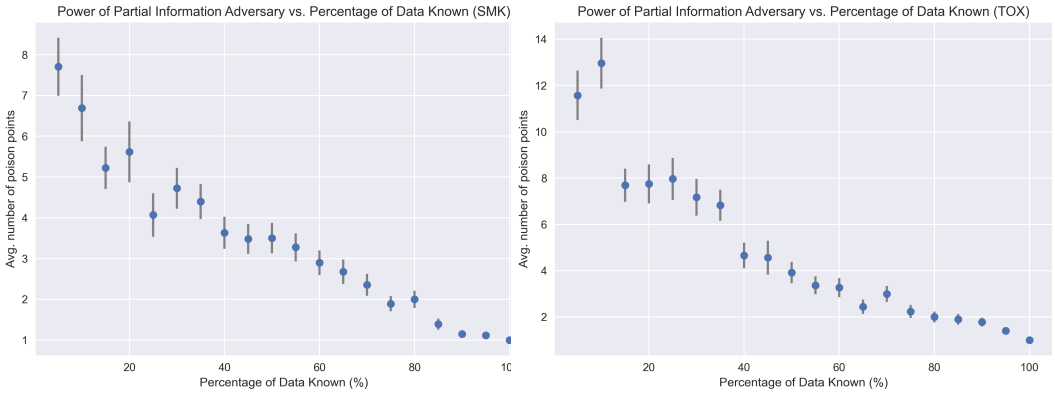

Figure 2: *SMK and TOX Experiments.* The behavior of attack on these two datasets is very similar to synthetic experiments. We believe this is because of the noisy nature of these feature selection datasets which causes them to be similar to the Gaussian distribution. Since the noise is large, even given the half of the dataset, the attacker cannot identify the most unstable feature.

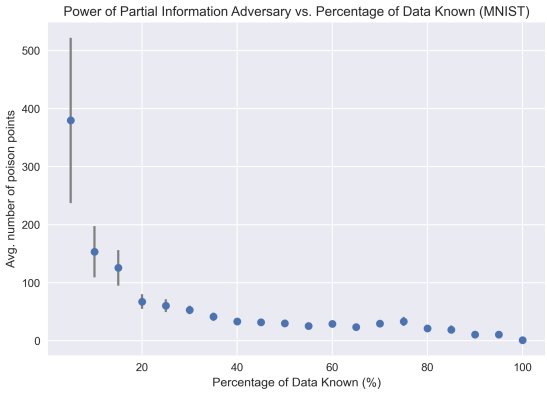

Figure 3: *MNIST experiments.* Compared to other experiments, the number of poison points drops faster as the percentage of data-awareness grows. This can be explained by separability (less noisy nature) of MNIST dataset.

5 Conclusion

In this paper we initiated a formal study of the power of data-oblivious adversaries who do not have the knowledge of the training set in comparison with data-aware adversaries who know the training data completely before adding poison points to it.

Our main result proved a separation between the two threat models by constructing a sparse linear regression problem. We show that in this natural problem, Lasso estimator is robust against data-oblivious adversaries that aim to add a non-relevant features to the model with a certain poisoning budget. On the other hand, for the same problem, we prove that data-aware adversaries, with the same budget, can find specific poisoning examples based on the rest of the training data in such a way that they can successfully add non-relevant features to the model. We also experimentally explored the partial-information adversaries who only observe a fraction of the training set and showed that even in this setting, the adversary could be much weaker than full-information adversary.

As a result, our work sheds light on an important and yet subtle aspect of modeling the threat posed by poisoning adversaries. We, leave open the question of separating different aspects of poisoning threat model including computational power of adversaries, computational power of learners, clean-label nature of adversaries and etc.

Acknowledgments. Mohammad Mahmoody was supported by NSF grants CCF-1910681 and CNS-1936799. Sanjam Garg is supported in part by DARPA under Agreement No. HR00112020026, AFOSR Award FA9550-19-1-0200, NSF CNS Award 1936826, and research grants by the Sloan Foundation, and Visa Inc. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

References

[1] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. *arXiv preprint arXiv:2005.00191*, 2020.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

[3] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25. ACM, 2006.

[4] Alex Beatson, Zhaoran Wang, and Han Liu. Blind attacks on machine learners. In *Advances In Neural Information Processing Systems*, pages 2397–2405, 2016.

[5] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643, 2019.

[6] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion Attacks against Machine Learning at Test Time. In *ECML/PKDD*, pages 387–402, 2013.

[7] Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Asian conference on machine learning*, pages 97–112, 2011.

[8] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1467–1474. Omnipress, 2012.

[9] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

[10] Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from cryptographic pseudo-random generators. *arXiv preprint arXiv:1811.06418*, 2018.

[11] Sébastien Bubeck, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. *arXiv preprint arXiv:1805.10204*, 2018.

[12] Sébastien Bubeck, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. *arXiv preprint arXiv:1805.10204*, 2018.

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

[14] Di Cao, Shan Chang, Zhijian Lin, Guohua Liu, and Donghong Sun. Understanding distributed poisoning attack in federated learning. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 233–239. IEEE, 2019.

[15] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017.

[16] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60. ACM, 2017.

[17] Akshay Degwekar and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. *arXiv preprint arXiv:1902.01086*, 2019.

[18] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 655–664. IEEE, 2016.

[19] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. *arXiv preprint arXiv:1803.02815*, 2018.

[20] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional Gaussians and Gaussian mixtures. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 73–84. IEEE, 2017.

[21] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical Gaussians. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1047–1060. ACM, 2018.

[22] Ilias Diakonikolas, Weihao Kong, and Alistair Stewart. Efficient algorithms and lower bounds for robust linear regression. *arXiv preprint arXiv:1806.00040*, 2018.

[23] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

[24] Dimitrios I Diochnos, Saeed Mahloujifar, and Mohammad Mahmoody. Lower bounds for adversarially robust pac learning. *arXiv preprint arXiv:1906.05815*, 2019.

[25] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

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

[27] Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference 2020*, pages 3019–3025, 2020.

[28] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

[29] Sanjam Garg, Somesh Jha, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarially robust learning could leverage computational hardness. *arXiv preprint arXiv:1905.11564*, 2019.

[30] Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276*, 2020.

[31] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Data security for machine learning: Data poisoning, backdoor attacks, and defenses. *arXiv preprint arXiv:2012.10544*, 2020.

[32] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[33] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[34] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR*, 2015.

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

[36] Junfeng Guo and Cong Liu. Practical poisoning attacks on neural networks. *Proceedings of the European Conference on Computer Vision*, 2020.

[37] David Harrison and Daniel Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5:81–102, 03 1978.

[38] Rui Hu, Yuanxiong Guo, Miao Pan, and Yanmin Gong. Targeted poisoning attacks on social recommender systems. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2019.

[39] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.

[40] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894, 2017.

[41] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

[42] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 69–75. IEEE, 2020.

[43] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 665–674. IEEE, 2016.

[44] Daniel Lowd and Christopher Meek. Adversarial learning. In *KDD*, pages 641–647, 2005.

[45] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: attacks and defenses. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4732–4738. AAAI Press, 2019.

[46] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

[47] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4536–4543, 2019.

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

[49] Saeed Mahloujifar and Mohammad Mahmoody. Can adversarially robust learning leverage computational hardness? *arXiv preprint arXiv:1810.01407*, 2018.

[50] Saeed Mahloujifar and Mohammad Mahmoody. Can adversarially robust learning leverage computational hardness? In *Algorithmic Learning Theory*, pages 581–609, 2019.

[51] Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal multi-party poisoning attacks. In *International Conference on Machine Learning*, pages 4274–4283, 2019.

[52] Blaine Nelson, Benjamin I. P. Rubinstein, Ling Huang, Anthony D. Joseph, and J. D. Tygar. Classifier Evasion: Models and Open Problems. In *PSDM*, pages 92–98, 2010.

[53] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[54] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael Wellman. Towards the science of security and privacy in machine learning. *arXiv preprint arXiv:1611.03814*, 2016.

[55] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[56] Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

[57] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially Robust Generalization Requires More Data. *arXiv preprint arXiv:1804.11285*, 2018.

[58] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018.

[59] Shiqi Shen, Shruti Tople, and Prateek Saxena. A uror: defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519. ACM, 2016.

[60] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[61] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529, 2017.

[62] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. When does machine learning {FAIL}? generalized transferability for evasion and poisoning attacks. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1299–1316, 2018.

[63] Lichao Sun. Natural backdoor attack on text data. *arXiv preprint arXiv:2006.16176*, 2020.

[64] Fnu Suya, Saeed Mahloujifar, David Evans, and Yuan Tian. Model-targeted poisoning attacks: Provable convergence and certified bounds. *arXiv preprint arXiv:2006.16469*, 2020.

[65] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

[66] Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory*, pages 819–850, 2013.

[67] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pages 480–501. Springer, 2020.

[68] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

[69] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[70] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[71] Sameer Wagh, Divya Gupta, and Nishanth Chandran. Securenn: 3-party secure computation for neural network training. *Proceedings on Privacy Enhancing Technologies*, 2019(3):26–49, 2019.

[72] Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202, 2009.

[73] Yizhen Wang and Kamalika Chaudhuri. Data poisoning attacks against online learning. *arXiv preprint arXiv:1808.08994*, 2018.

[74] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *ICML*, pages 1689–1698, 2015.

[75] Mengchen Zhao, Bo An, Wei Gao, and Teng Zhang. Efficient label contamination attacks against black-box learning models. In *IJCAI*, pages 3945–3951, 2017.

[76] Chen Zhu, W Ronny Huang, Ali Shafahi, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. *arXiv preprint arXiv:1905.05897*, 2019.

Checklist

1. All authors ...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Theorem 3.1
   (b) Did you describe the limitations of your work? [Yes] Yes, we elaborated on that in the intro.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Yes, see Introduction where we discuss the implication of our results.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Yes.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Yes, See Theorem 3.1
   (b) Did you include complete proofs of all theoretical results? [Yes] Yes, See Appendix.
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)? [Yes] We provide our code in supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Yes see Section E for the full details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report error bars in Figures 1, 2, and 3.
   (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)? [Yes] See Section E
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? [Yes] We cite each dataset we use.
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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] Not applicable.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Not applicable.
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Not applicable.

A Separation Result Between Data-oblivious and Data-aware Poisoning Attacks: Supplemental Material

In the supplemental material, we first provide a more detailed related work section. Then, In Section B we discuss some subtle aspects of defining poisoning attacks for data-oblivious and data-aware attacks. In Section C we provide the details on the borrowed results about the robustness of LASSO that we use to prove our results. Then, in Section D we provide the full proofs of the Theorems stated in the main body. In Section E we provide more details and additional experimental results for our feature selection experiments. Finally, in Sections F and G, we provide some additional theoretical and experimental results (These results are not stated in the main body) about separation of data-oblivious and data-aware adversaries for the case of classification.

A More Details on Related Work

As opposed to the data poisoning setting, the question of adversary's (adaptive) knowledge was indeed previously studied in the line of work on adversarial examples [44, 52, 65]. In a test time evasion attack the adversary's goal is to find an adversarial example, the adversary knows the input x *entirely* before trying to find a close input x' that is misclassified. So, this adaptivity aspect already differentiates adversarial examples from random noise. Moreover, the question of whether adversary knows the θ completely or it only has a black-box access to it [53] also adds another dimension of adaptivity to the story.

Some previous work have studied poisoning attacks in the setting of federated/distributed learning [5, 51]. Their attacks, however, either (implicitly) assume a full information attacker, or aim to increase the population risk (as opposed to injecting features in a feature selection task). Thus, our work is novel in both formally studying the differences between data-aware vs. data-oblivious attacks, and *provably* separating the power of these two attack models in the contexts of feature selection. Xiao et al. [74] also empirically examine the robustness of feature selection in the context of poisoning attacks, but their measure of stability is across sets of features. We are distinct in that our paper studies the effect of data-oblivious attacks on *individual* features and with provable guarantees.

Our work's motivation for data secrecy might seem similar to other works that leverage privacy-preserving learning (and in particular differential privacy [23, 26, 25]) to limit the power of poisoning attacks by making the learning process less sensitive to poison data [45]. However, despite seeming similarity, what we pursue here is fundamentally different. In this work, we try to understand the effect of keeping the data secret from adversaries. Whereas the robustness guarantees that come from differential privacy has nothing to do with secrecy and hold even if the adversary gets to see the full training set (or even select the whole training set in an adversarial way.).

We also distinguish our work with another line of work that studies the computational complexity of the attacker [49, 29]. Here, we study the "information complexity" of the attack; namely, what information the attacker needs to succeed in a poisoning attack, while those works study the *computational resources* that a poisoning attacker needs to successfully degrade the quality of the learned model. Another recent exciting line of work that studies the computational aspect of robust learning in poisoning contexts, focuses on the computational complexity of the *learning* process itself [18, 43, 16, 20, 21, 19, 56, 22], and other works have studied the same question about the complexity of the learning process for evasion attacks [11, 10, 17]. Furthermore, our work deals with information complexity and is distinct from works that study the impact of the training set (e.g., using clean labels) on the success of poisoning [58, 76, 62, 70].

Finally, we try to categorize the existing poisoning attacks in literature into data-oblivious and data-aware categories. The recent survey of [31] and classifies existing poisoning attacks based on their techniques and goals. We use the same classes to categorize the attacks.

- Feature Collision Attacks: [data-oblivious] Feature Collision is a technique used in targeted poisoning attacks where the adversary tries to inject poison points around a target point x so that the classification of x is different than the correct label [1, 36, 58, 76]. There is usually a "clean label" constraint for targeted attacks that prevents the adversary from using the same point as the target point. These attacks will be mostly categorized as data-oblivious as the attacker does not usually need to see the training set.

- Bi-level Optimization Attacks: [data-aware] Bi-level optimization is generic technique used for optimizing the poisoning points to achieve attacker's objective [8, 13, 30, 38]. This optimization heavily relies on knowledge of training set.
- Label-Flipping Attacks: [both] The idea of label-flipping is very simple yet effective. The random label-flipping attacks are data-oblivious as the only thing that the adversary does is to sample data from (conditional) distribution and flip the label. However, some variants of label-flipping [75, 28, 7] are relying on the training set to optimize the examples which makes them data-aware.
- Influence Function Attacks: [data-aware] Attacks based on influence function look at the effect of training examples on the final loss of the model[40, 27]. This technique require the knowledge of the training set.
- Online Learning Attacks: [data-aware] *Online* poisoning adversaries studied in [48, 73, 50]is a form of attack that lies somewhere between data-oblivious and data-aware attacks. In their model, an online adversary needs to choose its decision about the ith example (i.e., to tamper or not tamper it) based only on the history of the first $i-1$ examples, and without the knowledge of the future examples. So, their knowledge about the training data is limited, in a partial way. Since we separate the power of data-aware vs. data-oblivious attacks, a corollary of our results is that at least one of these models is different from the online variant for recovering sparse linear regression. In other words, we are in one of the following worlds: (i) online adversaries are provably stronger than data-oblivious adversaries or (ii) data-aware adversaries are provably stronger than online adversaries.
- Federated Learning Attacks: [both] The attack against federated learning [5, 67, 63, 14], use a range of ideas that covers all the previous techniques and hence have both data-aware and data-oblivious variants. In general, since in federated learning the adversary sees the model updates at each round, they are more aware of the randomness of training process compared to typical poisoning attacks hence they can be more effective.

B Further Details on Defining Oblivious Attacks

In this section, we discuss other definitional aspects of oblivious and full-information poisoning attacks.

B.1 Oblivious Variants of (Full-Information) Data Poisoning Attacks

In this section, we explain how to formalize oblivious poisoning attackers in general, and in the next subsection we will describe how to instantiate this general approach for the case of feature selection.

A poisoning adversary of "budget" k, can tamper with a training sequence $\mathcal{S} = \{e_1, \ldots, e_n\}$, by "modifying" $\mathcal{S}$ by at most k changes. Such changes can be in three forms

- **Injection.** Adversary can inject k new examples $e'_1, \ldots, e'_k$ to $\mathcal{S}$. This is without loss of generality when the learner is symmetric and is not sensitive to the order in the training examples. More generally, when the training set is treated like a sequence $\mathcal{S} = (e_1, \ldots, e_n)$, the adversary can even choose the *location* of these planted examples $e'_1, \ldots, e'_k$. More formally, the adversary picks k numbers $1 \leq i_1 < \cdots < i_k \leq n+k$, and constructs the new data sequence $\mathcal{S}' = (e''_1, \ldots, e''_{n+k})$ by letting $e''_j = e'_{i_j}$ and letting $\mathcal{S}$ fill the remaining coordinates of $\mathcal{S}'$ in their original order from $\mathcal{S}$.
  Oblivious injection. In the full-information setting, the adversary can choose the poison examples and their locations based on $\mathcal{S}$. In the oblivious variant, the adversary chooses the poison examples $e'_1, \ldots, e'_k$ and their locations $1 \leq i_1 < \cdots < i_k \leq n+k$ without knowing the original set $\mathcal{S}$.
- **Elimination.** Adversary can eliminate k of the examples in $\mathcal{S}$. When $\mathcal{S}$ is a sequence, the adversary only needs to state the indexes $1 \leq i_1 < \ldots, i_k \leq n$ of the removed examples.
  Oblivious elimination. In the full-information setting, the adversary can choose the locations of the deleted examples based on $\mathcal{S}$. In the oblivious variant, the adversary chooses the locations without knowing the original set $\mathcal{S}$.

- **Substitution and it oblivious variant.** These two settings are similar to data elimination, with the difference that the adversary, in addition to the sequence of locations, chooses k poison examples $e'_1, \ldots, e'_k$ to substitute e_{i_j} by e'_j for all $j \in [k]$.

More general attack strategies. One can think of more fine-grained variants of the substitution attacks above by having different "budgets" for injection and elimination processes (and even allowing different locations for eliminations and injections), but we keep the setting simple by default.

B.2 Taxonomy for Attacks on Feature Selection

Sometimes the goal of a learning process is to recover a model $\hat{\theta}$, perhaps from noisy data, that has the same set of features $\mathrm{Supp}(\hat{\theta})$ as the true model θ. For example, those features could be the relevant factors determining a decease. Such process is called feature selection (or model recovery). A poisoning attacker attacking a feature selection task would directly try to counter this goal. Now, regardless of *how* an attacker is transforming a data set $\mathcal{S}$ into $\mathcal{S}'$, let $\hat{\theta}'$ be the model that is learned from $\mathcal{S}'$. Below we give a taxonomy of various attack scenarios.

- **Feature adding.** In this case, the adversary's goal is to achieve $\mathrm{Supp}(\hat{\theta}') \not\subseteq \mathrm{Supp}(\theta)$. Namely, adding a feature that is not present in the true model θ.
- **Feature removal.** In this case, the adversary's goal is to achieve $\mathrm{Supp}(\theta) \not\subseteq \mathrm{Supp}(\hat{\theta}')$. Namely, removing a feature that is present in the true model θ.
- **Feature flipping.** In this case, the adversary's goal is to do either of the above. Namely, $\mathrm{Supp}(\theta) \neq \mathrm{Supp}(\hat{\theta}')$, which means that at least one of the features' existence is flipped.

Targeted variants of the attacks above. For each of the three attack goals above (in the context of feature selection), one can envision a *targeted* variant in which the adversary aims to add/remove or flip a specific feature $i \in [d]$ where d is the data dimension.

C Borrowed Results

In this section, we provide some preliminary results about the LASSO estimator. We first specify the sufficient conditions for a dataset that makes it a good dataset for robust recover using Lasso estimator. We borrow these specifications from the work of [66]. We use these results in proving Theorem 3.1.

Definition C.1 (Typical systems)**.** *Suppose $\theta^* \in [0,1]^d$ be a model such that $|\mathrm{Supp}(\theta^*)| = s$. Let $X \in \mathbb{R}^{n\times d}$ and $Y \in \mathbb{R}^{n\times 1}$ and $W = Y - X \times \theta^*$. Also let $X_I \in \mathbb{R}^{n\times s}$ be a matrix formed by columns of X whose indices are in $\mathrm{Supp}(\theta^*)$ and $X_O \in \mathbb{R}^{n\times(d-s)}$ be a matrix formed by columns of X whose indices are not in $\mathrm{Supp}(\theta^*)$. The pair $(\theta^*, [X \,|\, Y])$ is called an (n, d, s, ψ, σ)-*typical system, if the following hold:*

- ***Column normalization:*** *Each column of X has ℓ_2 norm bounded by $\sqrt{n}$.*
- ***Incoherence:*** $\left\|((X_O^T X_I)(X_I^T X_I)^{-1}\mathsf{sign}(\theta^*))\right\|_\infty \leq 1/4$.
- ***Restricted strong Convexity:*** *The minimum eigenvalue of $X_I X_I^T$ is at least ψ.*
- ***Bounded noise*** $\left\|X_O^T(I_{n\times n} - X_I(X_I^T X_I)^{-1}X_I^T)W\right\|_\infty \leq 2\sigma\sqrt{n\log(d)}$.

The following theorem is a modified version of result of [72] borrowed from [66].

Theorem C.2 (Model recovery with Lasso [72])**.** *Let $(\theta^*, [X \,|\, Y])$ be a (n, d, s, σ, ψ)-typical system. Let $\alpha = \mathrm{argmin}_{i\in[d]} \max(\theta_i^*, 1-\theta_i^*)$. If $n \geq 16 \cdot \frac{\sigma}{\psi\cdot\alpha}\sqrt{s\cdot\log(d)}$ and then $\hat{\theta} = \mathsf{Lasso}([X \,|\, Y])$ would have the same support as θ^* when $\lambda = 4\sigma\sqrt{n\cdot\log(d)}$.*

The following theorem is about robust model recovery with Lasso in [66].

Theorem C.3 (Robust model recovery with Lasso [66])**.** *Let $(\theta^*, [X \,|\, Y])$ be a (n, d, s, σ, ψ)-typical system. Let $\alpha = \mathrm{argmin}_{i\in[d]} \max(\theta_i^*, 1-\theta_i^*)$. If*

$$n \geq \max\left(\frac{16\sigma}{\psi\cdot\alpha}\sqrt{s\cdot\log(d)}, \frac{4s^4k^2(1/\psi+1)^2}{\log(d)\sigma^2}\right)$$

then $\hat{\theta} = \mathsf{Lasso}([X \,|\, Y])$ *would have the same support as* θ^* *when* $\lambda = 4\sigma\sqrt{n \cdot \log(d)}$.

In addition, adding any *set of* k *labeled vectors* $[X' \,|\, Y']$ *with* ℓ_∞ *norm at most 1 to* $[X \,|\, Y]$ *would not change the support set of the model recovered by Lasso estimator. Namely,*

$$\mathrm{Supp}\left(\mathsf{Lasso}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)\right) = \mathrm{Supp}(\mathsf{Lasso}([X \,|\, Y]) = \mathrm{Supp}(\theta^*).$$

Two theorems above are sufficient conditions for (robust) model recovery using lasso estimator. Bellow, we show two simple instantiating of the theorems on Normal distribution. Theorem bellow from the seminal work of Wainwright [72] shows that the Lasso estimator with proper parameters provably finds the correct set of features, if the dataset and noise vectors are sampled from normal distributions.

Theorem C.4 ([72])**.** *Let* X *be a dataset sampled from* $\mathcal{N}(0, 1/4)^{n\times d}$ *and* W *be a noise vector sampled from* $\mathcal{N}(0, \sigma^2)^n$*. For any* $\theta^* \in (0, 1)^d$ *with at most* s *number of non-zero coordinates, for* $\lambda = 4\sigma\sqrt{n \times \log(d)}$ *and* $n = \omega(s \cdot \log(d))$*, with probability at least* $3/4$ *over the choice of* X *and* W *(that determine* Y *as well) we have* $\mathrm{Supp}(\hat{\theta}) = \mathrm{Supp}(\theta^*)$ *where* $\hat{\theta} = \mathsf{Lasso}([X \,|\, Y])$*. Moreover,* $\hat{\theta}$ *is a unique minimizer for* $\mathrm{Risk}(\cdot, [X \,|\, Y])$.

The above theorem requires the dataset to be sampled from a certain distribution and does not take into account the possibilities of outliers in the data. The robust version of this theorem, where part of the training data is chosen by an adversary, can be instantiated using Theorem C.2 as follows:

Theorem C.5 ([66])**.** *Let* X *be a dataset sampled from* $\mathcal{N}(0, 1/4)^{n\times d}$ *and* W *be a noise vector sampled from* $\mathcal{N}(0, \sigma^2)^n$*. For any* $\theta^* \in (0, 1)^d$*, if* $\lambda = 4\sigma\sqrt{n \times \log(d)}$ *and* $n = \omega(s\log(d) + s^4 \cdot k^2)$*, with probability at least* $3/4$ *over the choice of* X, W *(determining* Y*), and* $Y = X \times \theta^* + W$ *it holds that, adding* any *set of* k *labeled vectors* $[X' \,|\, Y']$*, such that rows of* X' *has* ℓ_∞ *norm at most 1 and* Y *has* ℓ_∞ *norm at most* s*, to* $[X \,|\, Y]$ *would not change the support set of the model recovered by Lasso estimator. Namely,*

$$\mathrm{Supp}\left(\mathsf{Lasso}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)\right) = \mathrm{Supp}(\mathsf{Lasso}([X \,|\, Y]) = \mathrm{Supp}(\theta^*).$$

Note that Theorems C.4 and C.5 are instantiation of Theorems C.2 and C.3 for normal distribution and are proved by showing that the sufficient conditions of those theorems will happen with high probability over the choice of dataset.

D Proof of Proposition 3.6 and Theorem 3.1

In this section, we prove Proposition 3.6 and Theorem 3.1.

D.1 Proof of Proposition 3.6

Proof. We first argue that winning the data-aware game of Definition 2.1 is always possible. This is because, after getting the dataset $[X \,|\, Y]$ the adversary inspects the dataset to find out which coordinate is unstable and find a poisoning dataset that would add that unstable coordinate to the support set of the model.

Now, we prove the other part of the proposition. That is, we show that no adversary can win the oblivious security game of Definition 2.1 with probability more than ϵ. The reason behind this claim is the (k, ϵ)-resiliency of the dataset. For any fixed poisoning dataset S' selected by adversary, the probability of S' being successful in changing the support set is at most ϵ. Therefore, the best strategy for an adversary that does not see the dataset is to pick the best possible poison dataset that maximizes the average success over all training data sampled from D, which we know is smaller than ϵ because of the resiliency. Note that, by averaging argument, randomness does not help the oblivious attack.

Therefore, the proof of Proposition 3.6 is complete. □

D.2 Proof of Theorem 3.1

In this section, we prove Theorem 3.1. We first some intermediate theorems that will be used to prove the main result. The following theorem shows an upper bound on the number of examples that a data-aware adversary need to add a non-relevant feature to the support set of resulting model. Before stating the Theorem, we define two useful notions.

Definition D.1. *We define*

$$\alpha_i([X \,|\, Y]) = X^T[i](Y - X \cdot \hat{\theta})$$

where $\hat{\theta} = \mathsf{Lasso}([X \,|\, Y])$. *We also define* β_i *similarly with difference that the minimization of Lasso is done in the subspace of vectors with the correct support. Namely,*

$$\beta_i(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]) = X^T[i](Y - X \cdot \hat{\theta}')$$

where $\hat{\theta}' = \operatorname{argmin}_{\theta \in C} \frac{1}{n} \cdot \left\| \begin{bmatrix} Y \\ Y' \end{bmatrix} - \begin{bmatrix} X \\ X' \end{bmatrix} \times \theta \right\|_2^2 + \frac{2\lambda}{n} \cdot \|\theta\|_1$. *and* C *is the subspace of models that their* i*th feature is 0 for all* $i \notin \operatorname{Supp}(\theta)$.

Theorem D.2 (Unstability of Gaussian)**.** *Let* $X \in \mathbb{R}^{n \times d}$ *be an arbitrary matrix,* $\theta^* \in [0,1]^d$ *be an arbitrary vector,* W *be a noise vector sampled from* $\mathcal{N}(0, \sigma^2)^{n \times 1}$*, and let* $Y = X \times \theta^* + W$*. Also let* λ *be the penalty parameter that is used for Lasso. Then for any* i *there is a dataset* $[X' \,|\, Y']$ *with at most* $\lambda - |\alpha_i([[X \,|\, Y])|$ *examples of* ℓ_2 *norm at most* 1*, such that*

$$i \in \operatorname{Supp}\left(\mathsf{Lasso}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)\right).$$

Theorem above proves the existence of an attack that can add any feature to the training set. Below, we first provide the description of the attack. For completeness, the attack description is also provided in the proof of Theorem D.2.

The Attack: To attack a feature i with k examples, The attack first calculates $b = \operatorname{Sign}(\alpha_i([X \,|\, Y]))$ use a dataset $\mathcal{S}' = [X' \,|\, Y']$ as follows:

$$X' = \begin{bmatrix} 0 & \dots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{k \times d}, Y' = \begin{bmatrix} b \\ \vdots \\ b \end{bmatrix} \in \mathbb{R}^{k \times 1}.$$

The attack then adds S' to the training set. Note that this attack is oblivious as it does not use the knowledge of the clean training set. This is the attack that we use in our experiments in Section 4.

Now we state another theorem that shows a lower bound on the number of poisoning points required.

Theorem D.3. *Let* $[X' \,|\, Y']$ *be such that*

$$i \in \operatorname{Supp}\left(\mathsf{Lasso}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)\right)$$

and

$$i \notin \operatorname{Supp}\left(\mathsf{Lasso}\left([X \,|\, Y]\right)\right)$$

then for some $j \notin \operatorname{Supp}\left(\mathsf{Lasso}\left([X \,|\, Y]\right)\right)$ *we have*

$$2\left\|X'[j]\right\|_1 \geq \lambda - \beta_j(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]).$$

Definition D.4 (Re-sampling Operator)**.** *We define* $R(X, I, \sigma)$ *to be an operator that removes the* i*th column of* X *and replace it with a fresh sample from* $\mathcal{N}(0, \sigma^2)$ *for all* $i \in I$.

Lemma D.5. *Consider* $\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]$*, for any* $i \in [d]$ *and set* I *such that* $i \in I$*, we have*

$$\beta_i(\left[\begin{array}{c|c} R(X, I, \sigma) & Y \\ X' & Y' \end{array}\right]) \equiv \mathcal{N}(0, \sigma_2^2)$$

where $\sigma_2^2 = \left\|(Y - \hat{\theta}' X)\right\|_2^2 \cdot \sigma^2 \leq (n+k)\sigma^2$ *for* $\hat{\theta}'$ *of Definition D.1.*

Theorem D.6 (Resilience of Gaussian)**.** *Let $[X', Y']$ be a dataset such that $|X'|_1 \leq k$ and let $S = \mathrm{Supp}(\mathsf{Lasso}([X \,|\, Y])$ then we have*

$$\Pr[\mathrm{Supp}(\mathsf{Lasso}(\begin{bmatrix} R(X, [d] \setminus S, \sigma) & \Big| & Y \\ X' & \Big| & Y' \end{bmatrix})) \neq S] \leq 2e^{-\frac{(\lambda-2k)^2}{2\sigma_2^2}}$$

Theorem above states that if we re-sample the ith coordinate of X, then the probability of $[X', Y']$ being successful in adding ith feature to support set is limited.

Putting things together Now we put things together to complete the proof of Theorem 3.1. For the oblivious adversary, by Theorem D.6, the probability of the oblivious attacker succeeding according to Theorem D.3 is bounded by probability $2e^{-\frac{(\lambda-2k)^2}{2(n+k)\sigma^2}}$. This means, setting $\lambda = 2k + \sigma\sqrt{2(n+k)\log(2/\epsilon_2)}$ will guarantee that the oblivious attacker will succeed with probability at most ϵ_2. For the data-aware adversary, consider the distribution $\mathbb{R}(X, I, \sigma)(Y - X\hat{\theta})$. We know that this distribution is a Gaussian distribution with standard deviation $w\sigma$ for $w = \left\|Y - \hat{\theta}X\right\|_2$. Therefore, by Theorem D.2, and Gaussian tail bound, we know that with probability at least $p_1 \geq 1 - (1 - 2e^{-2\frac{(\lambda-k)^2}{w\sigma^2}})^{d-s}$ the data-aware adversary will succeed in adding a feature to the support set. Note that w is independent of d and is only a function of the features in the support set. Now, let w^* be a probabilistic lower bound on w so that $\Pr[w^2 > n \cdot {w^*}^2] \geq 1 - \varepsilon_1/2$. This is possible by selecting $w^* = \Omega(\sigma\sqrt{\log(-\epsilon_1)})$ because the expected value of error on each point is positive and proportionall to the variance of the noise vector (see [66]). Now, we can set $d = s + \frac{\log(1-\varepsilon_1/2)}{\log(1-2e^{-2\frac{(\lambda-k)^2}{nw^{*2}\sigma^2}})}$ so that the oblivious adversary succeeds with probability at least ε_1.

D.3 Proofs of Theorems D.2, D.3 and D.6 and Lemma D.5,

We first state and prove the following useful lemma.

Lemma D.7. *Let $X \in \mathbb{R}^{n\times d}$ and $Y \in \mathbb{R}^n$. Let $\hat{\theta}$ be a vector that minimizes $\mathrm{Risk}(\cdot, [X \,|\, Y])$. Then, for all non-zero coordinates $j \in [d]$, where $\hat{\theta}_j \neq 0$ we have*

$$\sum_{i=1}^{n} X_{(i,j)} \cdot (Y_i - \langle \hat{\theta}, X_i \rangle) = -\lambda \cdot \mathrm{Sign}(\hat{\theta}_j),$$

and for all 0 coordinates $j \in [d]$, where $\theta_j = 0$, we have

$$\left| \sum_{i=1}^{n} X_{(i,j)} \cdot (Y_i - \langle \hat{\theta}, X_i \rangle) \right| < \lambda.$$

Proof of Lemma D.7. Since $\hat{\theta}$ is a minimizer of $f(\cdot)$, the derivative of f should be 0 or undefined on all coordinates at $\hat{\theta}$. Note that, for all non-zero coordinates i the derivative of the second term $2\lambda \|\theta\|_1$ is equal to $2\lambda\,\mathrm{Sign}(\theta_i)$. Therefore, for non-zero coordinates the derivative of the first term should be equal to $-2\lambda \cdot \mathrm{Sign}(\theta_i)$. That is,

$$2(X^T \times (Y - X \times \hat{\theta}))_i = 2\lambda \cdot \mathrm{Sign}(\theta_i)$$

which proves the first part of the lemma. For the second part, note that the derivative of f does not exist, but the left-hand and right-hand derivatives exist and $\hat{\theta}$ minimizes f. Therefore, the left-derivative should be negative and the right hand derivative should be positive. Thus, we have

$$2(X^T \times (Y - X \times \hat{\theta}))_i + 2\lambda > 0,$$

and

$$2(X^T \times (Y - X \times \hat{\theta}))_i - 2\lambda < 0,$$

which implies that

$$-\lambda < (X^T \times (Y - X \times \hat{\theta}))_i < \lambda,$$

finishing the proof of the lemma. □

Lemma D.8. *Define* $f_{L,\sigma}(x) = \frac{erf(\frac{L+x}{\sigma})+erf(\frac{L-x}{\sigma})}{2erf(\frac{L}{\sigma})}$. *For any* $a \in R$ *and* $b \in R$ *we have* $f(a)f(b) > f(|a|+|b|)$.

Proof. Define $g(x) = \log(f_{L,\sigma}(x))$. It is easy to check that g is a concave function with the property that $|x|g'(|x|) \leq g(x)$. Assume $|b| < |a|$, we have

$$g(|a|+|b|) \leq g(|a|) + |b|g'(|a|) \leq g(a) + |b|g'(|b|) \leq g(a) + g(b)$$

□

Corollary D.9. *Let* $a = R^d$ *be a vector such that* $|a|_1 = l$ *and let* $b \equiv \mathbb{N}(0, \sigma^2)^d$. *We have,* $\Pr[|b+a|_\infty > r] \leq e^{\frac{(r-l)^2}{2\sigma^2}}$.

Proof. This follows from Lemma D.8 by writing the exact probability using the CDF of Gaussian and then applying a Gaussian tail bound. □

Now, we prove Theorem D.2 and show how to construct the poisoning dataset by using the Lemma above.

Proof of Theorem D.2. Let $k \geq \lambda - |\alpha_i([X \,|\, Y])|$ and consider X' which is a $k \times d$ matrix that is 0 everywhere except on the ith column that is 1 and Y' is a $k \times 1$ vector that is equal to $b = \text{Sign}(\alpha_i([X \,|\, Y])$ everywhere. We show that by adding this matrix the adversary is able to add i^{th} coordinate to the support set of the $\hat{\theta}' = \textsf{Lasso}\left(\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right)$. To prove this, suppose the ith coordinate of $\hat{\theta}'$ is 0. Thus, we have

$$\left(\begin{bmatrix} X \\ X' \end{bmatrix}^T \times \left(\begin{bmatrix} Y \\ Y' \end{bmatrix} - \begin{bmatrix} X \\ X' \end{bmatrix} \times \hat{\theta}'\right)\right)_i = kb + \left(X^T \times (Y - X \times \hat{\theta}')\right)_i. \tag{1}$$

Now we prove that $\hat{\theta}'$ also minimizes the Lasso loss over $[X \,|\, Y]$. This is because for any vector θ with i^{th} coordinate 0, we have

$$\text{Risk}\left(\theta, \left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right) = kb + \text{Risk}(\theta, [X \,|\, Y]).$$

Now, let $\hat{\theta}$ be the minimizer of $\text{Risk}(\cdot, [X \,|\, Y])$. We know that $\hat{\theta}$ is 0 on the ith coordinate. Therefore we have,

$$\begin{aligned} \text{Risk}\left(\hat{\theta}, \left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right) &= kb + \text{Risk}\left(\hat{\theta}, [X \,|\, Y]\right) \\ &\geq \text{Risk}\left(\hat{\theta}', \left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]\right) = kb + \text{Risk}(\hat{\theta}', [X \,|\, Y]). \end{aligned} \tag{2}$$

where the last inequality comes from the fact that $\hat{\theta}'$ minimizes the loss over $\left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right]$. On the other hand, we know that

$$\text{Risk}(\hat{\theta}', [X \,|\, Y]) \geq \text{Risk}(\hat{\theta}, [X \,|\, Y]) \tag{3}$$

because $\hat{\theta}$ minimizes $\text{Risk}(\cdot, [X \,|\, Y])$. Inequalities 2 and 3 imply that

$$\text{Risk}(\hat{\theta}, [X \,|\, Y]) = \text{Risk}(\hat{\theta}', [X \,|\, Y])$$

and that $\hat{\theta}$ minimizes $\text{Risk}(\cdot, \left[\begin{array}{c|c} X & Y \\ X' & Y' \end{array}\right])$. Therefore, based on Lemma D.7, since the ith coordinate of $\hat{\theta}$ is zero we have

$$\left|(\begin{bmatrix} X \\ X' \end{bmatrix}^T \times (\begin{bmatrix} Y \\ Y' \end{bmatrix} - \begin{bmatrix} X \\ X' \end{bmatrix} \times \hat{\theta}))_i\right| < \lambda. \tag{4}$$

However, by definition of α we have

$$\left| \begin{bmatrix} X \\ X' \end{bmatrix}^T \left(\begin{bmatrix} Y \\ Y' \end{bmatrix} - \begin{bmatrix} X \\ X' \end{bmatrix} \times \hat{\theta} \right)_i \right| = |\alpha_i([X \,|\, Y]) + \text{Sign}(\alpha_i([X \,|\, Y])) \cdot k| \geq \lambda.$$

This is a contradiction. Hence, the ith coordinate could not be 0 and the proof is complete. □

Now we prove Theorem D.3.

Proof of Theorem D.3. Consider $\hat{\theta}'$ to be the optimal model on the subspace defined by the support of $\hat{\theta}$. If $[X' \,|\, Y']$ adds feature i to the support set, then by uniqueness, $\hat{\theta}'$ cannot be a solution. This means that the sub-gradients of $\hat{\theta}'$ should not satisfy the properties of Lemma D.7. The only thing the adversary can do is to violate the condition on of the coordinates that are not in support. In particular, for some j, the jth coordinate must have

$$\left| \sum_{i=1}^{n+k} \begin{bmatrix} X \\ X' \end{bmatrix}_{(i,j)} \cdot \left(\begin{bmatrix} Y \\ Y' \end{bmatrix}_i - \langle \hat{\theta}', \begin{bmatrix} X \\ X' \end{bmatrix}_i \rangle \right) \right| \geq \lambda.$$

Therefore, by the norm constraint of the last k columns we have

$$\left| \sum_{i=1}^{n} X_{(i,j)} \cdot (Y_i - \langle \hat{\theta}', X_i \rangle) \right| \geq \lambda - 2 \left\| X'[j] \right\|_1 .$$

□

Now we prove Lemma D.5.

Proof of Lemma D.5. First, note that the re-sampling operation will either add the ith coordinate to support set or will not affect anything else in the obtained model. By definition, we have

$$\alpha_i([R(X, i, \sigma) \,|\, Y]) \equiv \sum_{i=1}^{n} (Y - \hat{\theta} X)[i] \cdot \mathcal{N}(0, \sigma^2) \equiv \mathcal{N}(0, \left\| (Y - \hat{\theta} X) \right\|_2^2 \sigma^2).$$

Similarly, for β_i we have

$$\beta_i(\left[\begin{array}{c|c} R(X, i, \sigma) & Y \\ X' & Y' \end{array} \right]) \equiv \sum_{i=1}^{n} (Y - \hat{\theta}' X)[i] \cdot \mathcal{N}(0, \sigma^2) \equiv \mathcal{N}(0, \left\| (Y - \hat{\theta}' X) \right\|_2^2 \sigma^2).$$

We know that

$$\left\| (Y - \hat{\theta}' X) \right\|_2^2 \leq (n + k) s^2$$

because θ' minimizes the criterion and should lead to a smaller loss than a model with 0 everywhere. Also,

$$\left\| (Y - \hat{\theta} X) \right\|_2^2 \geq \Omega(n)$$

as the expected error on each row is positive. □

Now we prove Theorem D.6.

Proof of Theorem D.6. Let $r_j = |X'[j]|$ and vector $r = (2r_1, \ldots, 2r_d)$. also define vector $\beta = (\beta_1, \ldots, \beta_d)$. According to Theorem D.3, we know that $|(r + \beta)|_\infty \geq \lambda$ must hold. On the other hand, by Lemma D.5 we know that β is distributed according to a Gaussian distribution with standard deviation σ_2. Therefore, by Corollary $D.9$ we can bound the probability of success of the adversary by $2e^{-\frac{(\lambda - 2k)^2}{2\sigma_2^2}}$.

□

E More Details about Feature Selection Experiments

Attacking a specific feature: For our experiments on feature selection we use the following attack similar to the attack described in Section D.2. To attack a feature i with k examples, we use a dataset $\mathcal{S}' = [X' \,|\, Y']$ as follows:

$$X' = \begin{bmatrix} 0 & \dots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{k \times d}, Y' = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{k \times 1}.$$

The attack then adds S' to the training set. Note that this attack is oblivious as it does not use the knowledge of the clean training set.

How to select the feature to attack? In total, the attack aims to find the feature that requires the minimum number of rows to add to $\mathrm{Supp}(\hat{\theta})$, where $\hat{\theta}$ is the learned parameter vector from Lasso. From Section D.2 we know that the attack above is almost optimal for adding a specific feature. However, the attack still has to decide which feature to attack as some features are much more unstable than others. A question that might arise here is why the power of data-oblivious and data-aware adversaries would change if the same underlying attack is used by both of them. The key here is that the data-aware adversary can search for the most vulnerable feature and target that feature specifically. However, the data-oblivious adversary has to choose the features that will *probably* be unstable. Namely, the attack should sample multiple dataset from the distribution and identify which features get unstable more frequently. However, there could be a lot of entropy in the instability of different features which makes the job of oblivious adversary hard.

Using this logic, we can even run experiment on partially-aware adversaries by only allowing them to see a random fraction of the training set. In this setting the adversary first tries to identify the unstable features by leveraging the part of the training set that it has access to. Specifically, the adversary first completes the dataset by randomly sampling from the distribution and combining it with the fraction of the data that it has access to. Then, it checks all the coordinates of this constructed dataset and identifies the one that requires the lowest number of attack samples, based on the attack described above. As the fraction of the known data increases, the approximation of stability of features decreases and the attacker can identify the unstable features with higher probability. This is the key in the decrease in the success of the attack when less data is known to the attacker.

Datasets: Here we give further details about the particular construction for our model recovery experiments in Section 4 and the real-world experiments for larger datasets.

E.1 Synthetic Experiments: Natural Gaussian Dataset

We construct the natural Gaussian dataset by sampling from $\mathcal{N}(0,1)^{n \times d}$, which gives us:

$$X = \begin{bmatrix} \mathcal{N}(0,1) & \dots & \mathcal{N}(0,1) \\ \vdots & \ddots & \vdots \\ \mathcal{N}(0,1) & \dots & \mathcal{N}(0,1) \end{bmatrix} \in \mathbb{R}^{n \times d}$$

We let $n = 300$ and $d = 5 \times 10^5$ and sample our noise vector W from $\mathcal{N}(0, \sigma^2)^{n \times 1}$ with $\sigma = 1/2$. Let $\theta^* \in [0,1]^d$ be the optimal parameter vector with the first $s = 5$ coordinates with non-zero values and the last $d - s$ coordinates equal to 0. With these parameters, we set the target variables to be $Y = X\theta^* + W$. For the Lasso estimator itself, we use $\lambda = 4\sigma\sqrt{n \log(d)}$ as in Theorem C.5.

E.2 Experiment with Real-world Data

For our real-world datasets, we use four datasets typically used in the model recovery setting.

- **Boston.** [37] (506 examples, 13 features) The task in this dataset is to predict the median value of a house in the Boston, Mass. area, given attributes that describe its location, features, and surrounding area. The outcome variable is continuous in the range $[0, 50]$.

- **TOX.** [32] (171 examples, 5,748 features) The task in this dataset is to predict whether a patient is a myocarditis and dilated cardiomyopathy (DCM) infected male, a DCM infected female, an uninfected male, or an uninfected female. Each feature is a gene, and each example is a patient. The outcome variable is discrete in $\{1, 2, 3, 4\}$, for each of the four possibilities.
- **Prostate_GE.** [33] (102 examples, 5,966 features) The task in this dataset is to predict whether a patient has prostate cancer. Each feature is a gene, and each example is a patient. The outcome variable is binary in $\{0, 1\}$, for cancer or no cancer.
- **SMK.** [32] (187 examples, 19,993 features) The task in this dataset is to predict whether a smoker has lung cancer or not. Each example is a smoker, and each faeture is a gene. The outcome variable is binary in $\{0, 1\}$ for cancer or no cancer.
- **MNIST** We use MNIST data for feature selection. Each pixel number constitutes a feature and we want to recover a subsst of them that are more relevant.

In Figure, we show the effect of knowledge of the data on Boston and Prostate datasets. We did not report these two experiments in the main body.

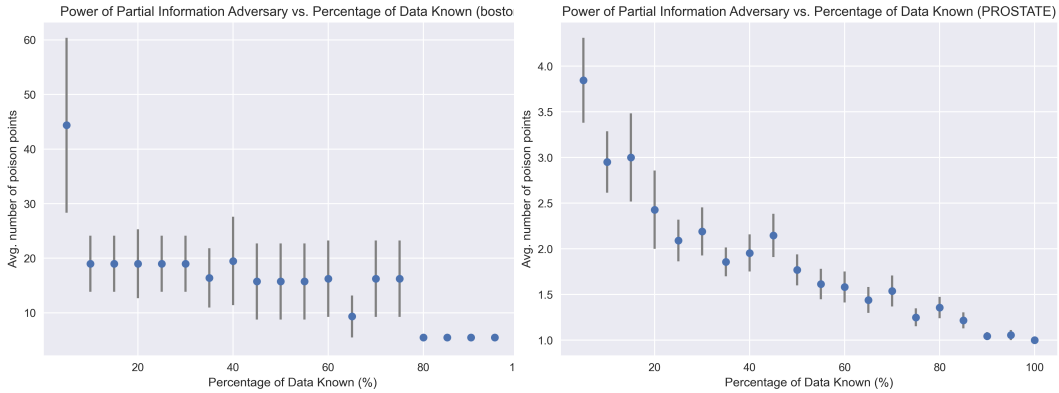

Figure 4: *SMK and TOX Experiments.*

F Separating Oblivious and Full-information Poisoning on Classification

In this section, we show a separation on the power of oblivious and full-information poisoning attacks on classification. In particular we show that empirical risk minimization (ERM) algorithm could be much more susceptible to full-information poisoning adversaries, compared to oblivious adversaries.

Before stating our results, we shall clarify that the attack on classification can also focus on different goals. One goal could be to increase the population risk of the resulting model θ' that the learner generates from the (poisoned) data $\mathcal{S}'$, compared to the model θ that would have been learned from $\mathcal{S}$ [61]. A different goal could be to make θ' fail on a particular test set of adversary's interest, making it a *targeted poisoning* [3, 59] or increase the probability of a general "bad predicate" of θ [47]. Our focus here is on attacks that aim to increase the population risk.

We begin by giving a formal definition of the threat model.

Definition F.1 (Oblivious and full-information data injection poisoning for population risk)**.** *We first describe the* data oblivious *security game between a challenger C and an adversary A, and then will describe how to modify it into a full-information variant. Such game is parameterized by adversary's budget k, a data set $\mathcal{S}$ a learning algorithm L, and a distribution D over $\mathcal{X} \times \mathcal{Y}$ (where $\mathcal{X}$ is the space of inputs and $\mathcal{Y}$ is the space of outputs).*[4]

$\mathbf{OblRisk}(k, \mathcal{S}, L, D)$.

1. *Adversary A generates k new examples $(e'_1, \ldots, e'_k)$ and them to C.*
2. *C produces the new data set $\mathcal{S}'$ by adding the injected examples to $\mathcal{S}$.*

[4]Since we deal with risk, we need to add D as a new parameter compared to the games of Definition 2.1.

3. *C runs L over $\mathcal{S}'$ to obtain (poisoned) model $\theta' \leftarrow L(\mathcal{S}')$.*
4. *A's advantage (in winning the game) will be* $\mathsf{Risk}(\theta', D) = \Pr_{(x,y)\leftarrow D}[\theta'(x) \neq y]$.[5]

In the full information *security game, all the steps are the same as above, except that in the first step the following is done.*

$\mathbf{AwrRisk}(k, \mathcal{S}, L, D)$.

- *Step 0: C sends $\mathcal{S}$ to A.*
- *The rest of the steps are the same as those of the game* $\mathbf{OblRisk}(k, \mathcal{S}, L, D)$.

One can also envision variations of Definition F.1 in which the goal of the attacker is to increase the error on a particular instance (i.e., a *targeted* poisoning [3, 59]) or use other poisoning methods that eliminate or substitute poison data rather than just adding some.

We now state and prove our separation on the power of oblivious and full-information poisoning attacks on classification. In particular we show that empirical risk minimization (ERM) algorithm could be much more susceptible to full-information poisoning adversaries, compared to oblivious adversaries.

Theorem F.2. *There is a distribution of distributions $\mathfrak{D}$ such that there is a data injecting adversary with budget $\varepsilon \cdot n$ that wins the full-information security game for classification by advantage ε, namely*

$$\exists A : \mathop{\mathbb{E}}_{\substack{D \leftarrow \mathfrak{D} \\ \mathcal{S} \leftarrow D^n}} \Big[\text{Advantage of } A \text{ in } \mathbf{AwrRisk}(\varepsilon \cdot n, \mathcal{S}, \mathsf{ERM}, D))\Big] \geq \Omega(\varepsilon).$$

On the other hand, any adversary will have much smaller advantage in the oblivious game. Namely, the following holds.

$$\forall A : \mathop{\mathbb{E}}_{\substack{D \leftarrow \mathfrak{D} \\ \mathcal{S} \leftarrow D^n}} \Big[\text{Advantage of } A \text{ in } \mathbf{OblRisk}(\varepsilon \cdot n, \mathcal{S}, \mathsf{ERM}, D))\Big] \leq O(\varepsilon^2).$$

Proof. Here we only sketch the proof. To prove this we use the problem of learning concentric halfspaces in Gaussian space $\mathcal{N}(0,1)^2$. We assume that the prior distribution is uniform over all concentric halfspaces. We first show that there is a full-information attack with success (ε). The way this attack works is as follows, attacker first uses ERM to learn a halfspace w_1 on the clean data. Assume this halfspace has risk δ. Then the attacker selects another halfspace w_2 that disagrees with w_1 on $\varepsilon \cdot n - 1$ number of points in the training data. Note that this is possible because the attacker can keep rotating the half-space until it has exactly $n \cdot \varepsilon - 1$ points disagreeing with w_1. Now if the adversary puts all the poison points on the separating line for w_1 and with the opposite label of what w_1 predicts, then ERM would prefer w_2 over w_1. Therefore the empirical error of w_2 on clean dataset would be at least equal to $\varepsilon - \delta$. Now if we increase n, the generalization error would go to zero which means the population error of w_2 would be close to $\varepsilon - \delta$. Also, since we are assuming the problem is realizable by half-spaces, it means δ would also converge to 0. Therefore, the final population risk could be bounded to be at least $\varepsilon/2$ for n larger than some reasonable values. Which means our proof for the full-information attack is complete.

Now, we show that no oblivious adversary cannot increase the error by more than ε^2, on average. The reason behind this boils down to the fact that each poison point added can affect at most ϵ-fraction of the choices of ground truth. To be more specific, we can fix the poison data to a fixed set D_p with size $\epsilon \cdot n$, as we can assume that the oblivious adversary is deterministic. Now if we fix the ground truth to some w^g, and define the epsilon neighborhood of a model w to be all the points that have angle at most $\epsilon \cdot \pi$ with w and denote it by w_ϵ. Then we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{X_c \leftarrow \mathcal{N}(0,1)^n \\ y_c = w^g(X_c) \\ D_c = (X_c, y_c) \\ w^p = \mathsf{ERM}(D_c \cup D_p), w^c = \mathsf{ERM}(D_c)}} [\mathsf{Risk}(w^p) - \mathsf{Risk}(w^c)] &\leq \mathop{\mathbb{E}}_{\substack{X_c \leftarrow \mathcal{N}(0,1)^n \\ y_c = w^g(X_c) \\ D_c = (X_c, y_c) \\ w^p = \mathsf{ERM}(D_c \cup D_p)}} [\mathsf{Risk}(\mathsf{ERM}(w^p))] \\
&\leq \mathop{\mathbb{E}}_{\substack{X_c \leftarrow \mathcal{N}(0,1)^n \\ y_c = w^g(X_c) \\ D_c = (X_c, y_c) \\ w^p = \mathsf{ERM}(D_c \cup D_p)}} [\mathsf{Risk}_{D_c}(w^p)] + \delta \qquad (5)
\end{aligned}$$

[5]Note that this is a real number, and more generally we can use any loss function, which allows covering the case ore regression as well.

where δ is the generalization parameter that relates to n and goes to 0 with rate $1/n$. Now consider an event E where the angle between w^c and w^g is at most $\epsilon \cdot \pi$ and $w^g_{2\epsilon} \cap X_c$ has at least ϵ points on each side of w^g. We denote the probability of this event by $1 - \delta'$ and we know that δ' goes down to 0 as n grows, by rate $1/\sqrt{n}$ (Using Chernoff Bound). Now we can observe that conditioned on E, we have $\mathsf{Risk}_{D_c}(w_p) \leq |w^g_{2\epsilon} \cap X_c|$. This is because the poison points cannot increase the errorn by more than ϵ so w^p would disagree with w^c on at most $\epsilon \cdot n$ points in D_c. On the other hand, we know that in 2ϵ neighborhood of w_g there are at least $\epsilon \cdot n$ points on each side of w_g, which means there are at least $\epsilon \cdot n$ points on each side of w^c (because w^c and w^g would fall between the same two points in D_c). Therefore, the poisoned model, would definitely be in the $2 \cdot \epsilon$ neighborhood of the w_g. At the same time, we know that the maximum number of points in D_c that w^g and w^p disagree on are at most equal to the number of poison points that fall in their disagreement region. And since the disagreement region is a subset of $w^g_{2\epsilon}$, we have the maximum number of points in D_c that w^g and w^p disagree on are at most equal to $|w^g_{2\epsilon} \cap X_c|$. Now having this, using Equation (12) we can write

$$\mathop{\mathbb{E}}_{\substack{X_c \leftarrow \mathcal{N}(0,1)^n \\ y_c = w^g(X_c) \\ D_c = (X_c, y_c) \\ w^p = \mathsf{ERM}(D_c \cup D_p), w^c = \mathsf{ERM}(D_c)}} [\mathsf{Risk}(w^p) - \mathsf{Risk}(w^c)] \leq \frac{|D_p \cap w^g_{2\epsilon}|}{n} + \delta + \delta'$$

Now by also taking the average over w^g we get

$$\mathop{\mathbb{E}}_{\substack{w^g \leftarrow \mathfrak{D} \\ X_c \leftarrow \mathcal{N}(0,1)^n \\ y_c = w^g(X_c) \\ D_c = (X_c, y_c) \\ w^p = \mathsf{ERM}(D_c \cup D_p), w^c = \mathsf{ERM}(D_c)}} [\mathsf{Risk}(w^p) - \mathsf{Risk}(w^c)] \leq \mathop{\mathbb{E}}_{w^g \leftarrow \mathfrak{D}} [\frac{|D_p \cap [w^g_{2\epsilon}|}{n}] + \delta + \delta' = 2\epsilon^2 + \delta + \delta'$$

As δ and δ' converge to 0 with rate $1/\sqrt{n}$, for $n \geq \omega(1/\epsilon^2)$ we have

$$\mathop{\mathbb{E}}_{\substack{w^g \leftarrow \mathfrak{D} \\ X_c \leftarrow \mathcal{N}(0,1)^n \\ y_c = w^g(X_c) \\ D_c = (X_c, y_c) \\ w^p = \mathsf{ERM}(D_c \cup D_p), w^c = \mathsf{ERM}(D_c)}} [\mathsf{Risk}(w^p) - \mathsf{Risk}(w^c)] \leq O(\epsilon^2).$$

□

We also state the theorem about separation of oblivious and full-information adversaries in the data elimination setting. This theorem has shows that the gap between oblivious and full-information adversaries could be wider in the data elimination settings. We use $\mathbf{AwrRisk}^{\mathsf{elim}}$ and $\mathbf{OblRisk}_{\mathsf{elim}}$ to denote the information risk in presence of oblivious and full-information data elimination attacks.

Theorem F.3. *There is a distribution of distributions $\mathfrak{D}$ such that there is a data elimination adversary with budget $\varepsilon \cdot n$ that wins the full-information security game for classification by advantage ε, namely*

$$\exists A : \mathop{\mathbb{E}}_{\substack{D \leftarrow \mathfrak{D} \\ \mathcal{S} \leftarrow D^n}} \Big[\text{Advantage of } A \text{ in } \mathbf{AwrRisk}^{\mathsf{elim}}(\varepsilon \cdot n, \mathcal{S}, \mathsf{ERM}, D))\Big] \geq \Omega(\varepsilon).$$

On the other hand, any adversary will have much smaller advantage in the oblivious game. Namely, the following holds.

$$\forall A : \mathop{\mathbb{E}}_{\substack{D \leftarrow \mathfrak{D} \\ \mathcal{S} \leftarrow D^n}} \Big[\text{Advantage of } A \text{ in } \mathbf{OblRisk}_{\mathsf{elim}}(\varepsilon \cdot n, \mathcal{S}, \mathsf{ERM}, D))\Big] \leq e^{-\omega((1-\varepsilon)n)}.$$

Proof. For the negative part on the power of full-information attacks, we observe that for a fixed w_g the attacker can find a half-space w_c that has angle $\pi\epsilon/2$ with the ground-truth w_g, and remove all the points where w_c and w_g disagree. Note that the number of points in the disagreement region would be at most ϵ with some large probability $1 - \delta$ where δ goes to 0 with rate $1/\sqrt{n}$. After the adversary removes all the points in disagreement region, the learner cannot distinguish them and will incur an error $\epsilon/2$ on average. We note that this attack is similar to the hybrid attack described in the work of Diochnos et al. [24]. For the positive result, we make a simple observation that oblivious poisoning adversary can only reduce the sample complexity for the learner. In other words, non-removed examples would remain i.i.d examples. This means that after removal, we can still use uniform convergence theorem to bound the error of resulting classifier. Since the error of learning realizable half-spcaces will go to zero with rate $\Omega(1/n)$, therefore the average error after the attack would be $\Omega(1/(1-\epsilon)n))$. □

G Separating Data-Oblivious and Data-Aware Classification: Experiments

In this section, we design an experiment to empirically validate the claim made in Theorem F.2, that there is a separation between oblivious and data-aware poisoning adversaries for classification. We setup the experiment just as in the proof of Theorem F.2, as follows.

First, we sample training points $X = x_1, x_2, \ldots x_m$ for $m = 1,000$ from the Gaussian space $\mathcal{N}(0,1)^2$, and pick a random ground-truth halfspace w^* from $\mathcal{N}(0,1)^2$. Using w^*, we find our labels $y_1, y_2, \ldots y_m$ by taking $(w^*)^T x_k$ for $k \in [m]$. This ensures the data is linearly separable by the homogeneous halfspace produced by w^*.

To attack this dataset simulating our data-aware adversary with budget ϵ, we construct $\epsilon \cdot m$ poison points d as follows:

$$d = \cos(\epsilon\pi) \cdot \frac{v}{\|v\|} + \sin(\epsilon\pi) \cdot \frac{w}{\|w\|}, \quad \text{where } v = \begin{bmatrix} 1, & -\frac{w_1}{w_2} \end{bmatrix}$$

and we add $\epsilon \cdot m$ of these d rows to our dataset. Note that this specific d corresponds to halfspace w_2 in our Proof of Theorem F.2, the halfspace obtained by rotating the original halfspace until it has exactly $\epsilon \cdot m$ points disagreeing with w^*. We label each of these d rows to be $y_d = -(w^*)^T d$, the opposite label from ground-truth. Then, we train our halfspace via ERM on this poisoned dataset of $m \cdot (1+\epsilon)$ points (from appending $\epsilon \cdot m$ rows of d). We evaluate our poisoned halfspace on another $X' = x'_1, x'_2, \ldots x'_m$ test points from the same Gaussian $\mathcal{N}(0,1)^n$ distribution.

To attack this dataset simulating the oblivious adversary, we try three oblivious strategies of attack that an adversary with no knowledge of the dataset might wage, each with ϵ budget:

1. Sample a single random point p from $\mathcal{N}(0,1)^n$ and repeat it $\epsilon \cdot m$ times. Choose the label p_y uniformly at random from $\{-1, 1\}$. Poison by adding these $\epsilon \cdot m$ rows to the dataset.
2. Sample $\epsilon \cdot m$ points IID from $\mathcal{N}(0,1)^n$ and choose the label p_y uniformly at random from $\{-1, 1\}$. Label all of the $\epsilon \cdot m$ points with p_y. Poison by adding these $\epsilon \cdot m$ rows to the dataset.
3. Sample $\epsilon \cdot m$ points IID from $\mathcal{N}(0,1)^n$ and choose the label p_y uniformly at random from $\{-1, 1\}$ for *each point.* That is, we flip a coin to label each poison example, rather than just choosing one label, as in 2. Poison by adding these $\epsilon \cdot m$ rows to the dataset.

We also use the same ERM algorithm, as in the data-aware case, to train the poisoned classifiers on these three oblivious poisoning strategies.

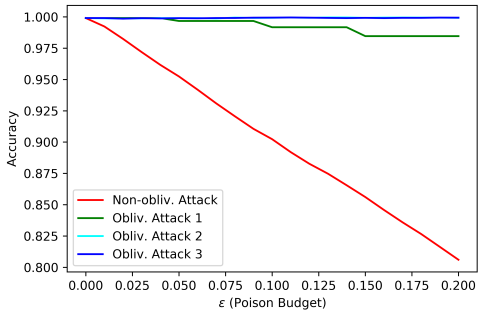

Figure 5: *Oblivious and data-aware poisoning separation in classification.* Over 20 trials, we vary the poisoning budget ϵ and construct poisoned datasets as discussed above for each adversary. We plot the effect of each adversary's attack on the accuracy of our resulting poisoned ERM halfspace.

We repeat this experiment 20 times for poison budget $\epsilon \in \{0, 0.01, 0.02, \ldots 0.19, 0.2\}$. We observe in Figure 5 that there indeed exists a separation between the power of our data-aware adversary and the oblivious adversaries. The data-aware adversary can increase the error linearly with ϵ using this strategy, while the oblivious adversaries fail to have any consistent impact on the resulting classifier's error with their strategies.