A Methodology for Modeling Model-Inversion Attacks

***Abstract*—Model inversion (MI) is a type of attack on the confidentiality of training data induced by releasing machine-learning models, and has recently received increasing attention. Motivated by existing MI attacks and other previous attacks that turn out to be MI attacks "in disguise," this paper initiates a formal study of MI attacks by presenting a game-based methodology. Our methodology uncovers a lot of subtle issues, and devising a rigorous game-based definition, analogous to those in cryptography, is an interesting avenue for future work. We describe methodologies for two types of attacks: the methodology for *black-box*, considers an adversary who infers sensitive values with only oracle access to a model. The second methodology targets the *white-box* scenario where an adversary has some additional knowledge about the structure of a model. For the restricted class of Boolean models and black-box attacks, we characterize model invertibility using the concept of *influence* from Boolean analysis in the noiseless case, and connect model invertibility with *stable influence* in the noisy case. Interestingly, we also discovered an intriguing phenomenon, which we call "invertibility interference," where a highly invertible model quickly becomes highly non-invertible by adding little noise. For the white-box case, we consider a common phenomenon in machine-learning models where the model is a *sequential composition of several sub-models*. We show, quantitatively, that even very restricted communication between layers could leak a significant amount of information. Perhaps more importantly, our study also unveils unexpected computational power of these restricted communication channels, which, to the best of our knowledge, were not previously known.**

I. Introduction

Privacy concerns surrounding the release of statistical information have received considerable attention in the past decade. The goal of statistical data privacy research is to enable accurate extraction of valuable data patterns, while preserving an individual's privacy in the underlying dataset against *data privacy attacks*. In general, there have been two flavors of data privacy attacks in the literature. The first is against a *specific* privacy notion, such as *differential privacy* [1]. Investigations of such attacks have led to lower bounds (e.g. [2], [3]). The second kind of attacks is against *attribute privacy*, which is a general concept where one studies how much distortion is needed in order to prevent an adversary from inferring sensitive attributes from non-sensitive ones (for concreteness, see for example *reconstruction attacks*, e.g. [4], [5]). In particular, attribute privacy attacks are widely considered in the applied data privacy literature, such as releasing medical information.

This paper focuses on a specific type of attacks of the second type. The class of attacks we consider relate to inferring sensitive attributes from a released model (e.g. a machine-learning model), or *model inversion* (MI) attacks. Several of these types of attacks have appeared in the literature. Recently, Fredrikson et al. [6] explored MIT attacks in the context of personalized medicine. Specifically, Fredrikson et al. [6] "invert" the publicly released linear regression model in order to infer a sensitive genetic marker, based on the model *output* (Warfarin dosage) plus several other non-sensitive attributes (e.g., height, age, weight). Interestingly, they demonstrate that additionally knowing the model output (Warfarin dosage here), or even its reasonable approximation, leads to a statistically significant increased leakage of sensitive attribute. This leads to natural questions of how widely such effective and efficient inversion attacks exist for statistical models, as well as how to quantify the *additional* leakage due to accessing the model.

Recently, more "examples" of effective MI attacks were discovered, which further stimulated interest in these attacks. For example, [7] considers a "white-box" MI attack in the setting of inferring image features. They demonstrated that by exploiting the additional confidence information provided by common image processing libraries, one can significantly improve both the effectiveness and efficiency of an MI attack. Interestingly, we note that these attacks are reminiscent of privacy attacks discussed in the context of inverting highly compressed image features [8], [9], [10], which were explored before. It is our belief, however, that if we are to develop countermeasures against all these attacks, or even a precise explanation of the dangers they pose, we will need to go beyond example-based definitions and require a methodology to capture this phenomenon. We consider this paper to be the first step and lot of work remains to be done.

In this paper we take a first step toward providing a formal treatment of MI attacks. Our contributions are summarized as follows:

- We present two methodologies, which are both inspired by the "two world" games common in cryptographic definitions. A methodology for *black-box* attacks, where the adversary has oracle access to the model, and a methodology for *white-box* attacks, where the adversary has information about the model structure. Our methodology provides a "blue print" for making these definitions precise for specific cases. Extending this to a precise general definition (such as the real and the ideal world definition used in the SMC literature) will be interesting to pursue. Our methodology considers machine-learning (ML) models because those were the target of existing MI attacks. One shortcoming of our methodology is that we do not take into account the specific structure of the ML model or the learning task. Again, connecting our methodology to various notions in the ML literature (such as stability) provides an attractive avenue for future work.
- We then specialize our methodology to important special cases, in order to isolate *important factors that affect model invertibility* (i.e. how successfully one can invert the model). Identifying these factors is important for at least two applications: First, as a decision procedure during publishing a model, estimating invertibility can let us gauge the leakage of sensitive attributes, and thus help in deciding which part of a model is publishable. The second use is to help in preventing MI attacks. If the invertibility is low, then little noise may be used to effectively prevent MI attacks without sacrificing too much utility.
- For the case of models that are Boolean functions (e.g., decision trees with attributes having finite domains), we have some concrete results. In this case, we can leverage powerful tools from Boolean analysis. Specifically for black-box MI attacks where the adversary knows the model output and precisely all other features, and there is no noise, we show that model invertibility is characterized by *influence* from Boolean analysis. Unfortunately, it becomes significantly more complicated if there is noise in the prior knowledge of the adversary. Nevertheless, we show that the invertibility is related to *stable influence* in Boolean analysis. Interestingly, our exploration in the noisy situation also unveils a phenomenon where a highly invertible model quickly becomes highly non-invertible by adding a little noise. We study such phenomenon under the name "invertibility interference."
- For white-box MI attacks, we study a common phenomenon where the computation of a machine learning model is a sequential composition of several layers or models. Exploiting the intermediate information communicated between these layers, even when it is highly compressed, can give a significant advantage to the adversary. In fact, the white-box attack described in [7] exploits exactly such information where the confidence information is the likelihood probabilities computed at an intermediate layer of the model. We thus study how these restricted communication channels could leak information. Interestingly, our results show, quantitatively, that even with 1 *bit of communication there could be a significant leakage*. Our results also unveil unexpected computational power of these restricted channels, which, to the best of our knowledge, were previously unknown.

The rest of the paper is organized as follows: Section II describes our methodologies for black-box and white-box MI attacks. Then in Section III we give some technical background that is necessary for our development later. Section IV and Section V, specializes our general formulation to important special cases. Finally, we conclude the paper in Section VI by discussing connections of our formulation with other cryptographic notions.

II. A Methodology for Formalizing MI Attacks

An essential goal of studying MI attacks is to *quantify* the strength of the correlation between sensitive attributes and the output of the model. While this goal is very intuitive, formalizing these attacks poses a challenge due to the diversity of such attacks. Moreover, as we mentioned earlier, many different attacks can be viewed as "MI attacks." This suggests that it can be difficult to give a "unified" definition of MI attacks without risking over generalization (i.e., even a lot of benign cases with "weak correlation" will be classified as attacks). As a first attempt, our goal is thus to abstract out important factors from existing attacks, and present a methodology. Guided by these methodologies, later in this paper we identify special cases of MI attacks that lead to theoretical insights.

This section is organized as follows: We start by discussing concepts from machine learning, which provides the background for our methodology. Then we discuss MI attacks in an intuitive manner. In Section II-A and II-B we present methodologies for black-box MI and white-box MI attacks, respectively. Along the way, we discuss how our methodology captures existing attacks and can be used to model other interesting scenarios that have not been addressed before.

Background. We formalize MI attacks in the *generalized learning setting*. In the generalized learning setting, a machine learning task is represented as a triple (Z, H, ℓ), where Z is a sample space, H is a hypothesis space, and $\ell : H \times Z \mapsto \mathbb{R}$ is a loss function. Given a data generating distribution $\mathscr{D}$, the goal of learning is to solve the following stochastic optimization problem:

$$\min_{h \in H} \mathop{\mathbb{E}}_{x \sim \mathscr{D}}[\ell(h, z)].$$

In machine learning however, $\mathscr{D}$ is unknown and one must find an approximate solution using a dataset S of i.i.d. samples from $\mathscr{D}$.

Recall that in the supervised learning setting, Z is of the form $X \times Y$ where X is called a feature space and Y an output space. Further, a hypothesis $h \in H$ has to take the form $X \mapsto Y$. On the other hand, the generalized learning setting, as formulated above, also incorporates unsupervised learning. For example in clustering, one maps $z \in Z$, a collection of points, to a set of clusters.

MI Attacks: Scenarios and Observations. Intuitively, MI attacks are designed to capture privacy concerns about *participants in a training set*, which arise from the following scenario: An organization trains a model over some dataset collected from a large set of individuals. After restricted access (say under some strict access control) to the model *within the organization*, now they want to release the model to the *public* for general use (e.g. say by a medical-clinic that specializes in providing personalized medicine.) We envision two mechanisms for releasing a model: release the model as a black box so public can use it freely, or release the model as a white box with some information about its architecture and parameters published. The concern is that certain correlation encoded in the model may be too strong such that a potential adversary can leverage the publicly published model, plus additional knowledge about individuals in the training set, to recover *participants'* sensitive information. The essential goal of studying MI attacks is to *quantify the strength of such correlations* so that one can have a better understanding to what degree such concerns matter.

Towards this goal, one thus needs to formulate a reasonable adversary model to capture how an adversary may exploit the model. We have the following simple observations: (1) We are interested in MI attacks in the *test phase* of machine learning, where a model h has already been trained. (2) It is necessary to have some objective for an attack, which can be captured by some function τ that maps a sample $z \in Z$ to some range. (3) The quantification is carried over the training dataset, since the main concern is for participants in the dataset. (4) The quantification is supposed to compare "two worlds", one where the adversary has access to the model, and the other where the adversary does not. This is to capture the fact that we want to quantify the *additional risk* of releasing a model.

Limitations: Next we discuss some limitations of our methodology, and addressing these limitations provides interesting avenue for future work. Our methodology focusses on one organization, so for example, our model does not cover the following scenario: a different organization can collect data $S^\star$ similar to S and build a model $h^\star$ (may be using the same learning algorithm), which can then be used to infer sensitive information about participants. Moreover, the results need to be interpreted on a case-by-case basis. For example, assume that our definition with parameterization for a specific context yields advantage of $\frac{1}{N}$, where N is the size of the training set S. Should we consider this an attack? This depends on the context. We admit that our methodology does not exploit structure of the ML task and model (e.g., perhaps looking at the loss function l). In general, we believe that what is considered a privacy breach is highly dependent on the context.

A. Black-Box MI

We now present a methodology for formalizing black-box MI attacks where the adversary has oracle access to a model. Along the way, we introduce notation that will be used later.

Measuring Effectiveness of An Attack. It is attractive to consider the success of an attack on a single sample point. While this might be sensible in some specific scenarios (for example, the adversary wants to get genetic information about a specific individual), it does not seem to be a good formal measure. This is because a machine learning model, in contrast to an encryption, is supposed to communicate some information about the sample, so in the worst case it is always possible to extract some information about a specific individual.

On the other hand, one may attempt to measure an attack over the *data-generating distribution* $\mathscr{D}$, which is in the definition of a machine-learning task. However, this leads to a complication as $\mathscr{D}$ is unknown in general and so one has to impose assumptions on its structure. We choose to measure an attack over the dataset used to train the model. This thus provides a privacy loss measure for participants in the dataset. Moreover, this allows us to carry out the quantification without an additional parameter $\mathscr{D}$.

Adversaries and Their Power. We first note that a model at the test phase is fixed, so there is no asymptotic

behavior since there is no infinite family of models. We thus model an adversary as a probabilistic algorithm without limiting its computational complexity. In other words, the adversary is *all powerful*. We note that other data privacy formulations, such as differential privacy [1], also make such an assumption on the adversarial power.

We now present a *methodology* for formulating black-box MI attacks with the goal of measuring the effectiveness of these attacks. To use this methodology as a template to generate precise definitions for specific scenarios, one has to instantiate auxiliary information generators gen and sgen in the methodology for attacks and simulated attacks, respectively. Having two different generators in the two worlds allows us additional flexibility (e.g., for example in the Warfarin attack the attacker in the MI-Attack world knows some "approximation" of the Warfarin dosage.) It is true that in some cases, gen and sgen will be the same.

Methodology 1. The starting point of an MI attack is a machine learning problem, specified as a triple (Z, H, ℓ). We use the following notations:

1) Γ: A training algorithm of the learning problem, which outputs a hypothesis $\Gamma(S) \in H$ on an input training set S.
2) $\mathscr{D}_S$: A distribution over the training set S.
3) τ: The objective function computed by the adversary. For now, one can view it simply as some function that maps Z to $\{0,1\}^*$.
4) $\mathsf{gen}, \mathsf{sgen}$: Auxiliary information generators. They map a pair (S, z) to an advice string in $\{0,1\}^*$.

As we noted before there are two worlds in our methodology (the MI-attack world) and (the simulated attack world).

The MI-attack world: This world is described by a tuple $(A, \mathsf{gen}, \tau, S, \mathscr{D}_S, \Gamma)$, where the adversary (A) is a probabilistic *oracle machine* (recall that gen generates an advice string for the adversary from (S, z)). Now the following game is played between the Nature and the adversary A.

(1) Nature draws a sample z from $\mathscr{D}_S$.
(2) Nature presents $\nu = \mathsf{gen}(S, z)$ to the adversary.
(3) Adversary outputs $A^{\Gamma(S)}(\nu)$.

The gain of the game is evaluated as

$$\mathsf{gain}(A, \mathsf{gen}, \tau, S, \mathscr{D}_S, \Gamma) = \Pr[A^{\Gamma(S)}(\mathsf{gen}(S, z)) = \tau(z)]$$

where the probability is taken over the randomness of $z \sim \mathscr{D}_S$, the randomness of gen, and the randomness of A. In other words, the gain is the probability that the adversary A with oracle access to the model $\Gamma(S)$ and given the advice string generated by sgen is able to "guess" $\tau(z)$.

The simulated world: is described by a tuple $(A^*, \mathsf{sgen}, \tau, S, \mathscr{D}_S)$, where the adversary $(A^\star)$ is a non-oracle machine and sgen is the second auxiliary information generator. The game between the Nature and $A^\star$ is exactly the same as in the MI-attack world, but $A^\star$ *does not* have oracle access to the learned model $\Gamma(S)$. Similarly, the gain is defined as:

$$\mathsf{sgain}(A^*, \mathsf{sgen}, \tau, S, \mathscr{D}_S) = \Pr[A^*(\mathsf{sgen}(S, z)) = \tau(z)]$$

where the probability is taken over the randomness of $z \sim \mathscr{D}_S$, the randomness of sgen, and the randomness of A^*.

Advantage: For (τ, S, Γ), the *advantage* of (gen, A) over (sgen, A^*) is computed as

$$\mathsf{adv}^{(\mathsf{gen},A)}_{(\mathsf{sgen},A^*)} = |\,\mathsf{gain}(A, \mathsf{gen}, \tau, S, \mathscr{D}_S, \Gamma) - \mathsf{sgain}(A^*, \mathsf{sgen}, \tau, S, \mathscr{D}_S)|.$$

Leakage: We say that $\Gamma(S)$ has ε-leakage for $(\tau, \mathscr{D}_S)$ with respect to $(\mathsf{gen}, \mathsf{sgen}, A)$ if there exists an adversary A^* such that $\mathsf{adv}^{(\mathsf{gen},A)}_{(\mathsf{sgen},A^*)} \leq \varepsilon$. Finally, $\Gamma(S)$ has ε-leakage for $(\tau, \mathscr{D}_S)$ with respect to $(\mathsf{gen}, \mathsf{sgen})$ if for any probabilistic adversary A, there exists an adversary A^* such that $\mathsf{adv}^{(\mathsf{gen},A)}_{(\mathsf{sgen},A^*)} \leq \varepsilon$.

We remark that an interesting special case is to evaluate the gain against a *uniform distribution* over the training set. This case is interesting because a uniform distribution over the training set gives an approximation of the underlying data generating distribution $\mathscr{D}$, as S is i.i.d. drawn from $\mathscr{D}$. As a result, the gain against the uniform distribution over the training set also approximately measures the strength of the correlation for the data generating distribution.

Modeling Examples. We now discuss several examples of applying our methodology.

Example 1 (Warfarin Attack [6])**.** *Our first example is the Warfarin-dosage attack in the original work of Fredrikson et al. [6]. The Warfarin-dosage attack is a black-box MI attack in the supervised learning setting. Thus $Z = X \times Y$ and $X = \prod_{i=1}^{n} X_i$ where X_i's are binary encoding of features, such as genotypes, race, etc. The attack, put in our formalization, is summarized in Table I.*

Note that in this formulation z takes the form of (x, y) and we feed y to the adversary (not $h(x)$). This is

The MI-Attack World	The Simulated World
Oracle access to (Γ: a linear-regression model).	No access to the oracle.
τ: $\tau(z) = x_i$, the VKORC1 genetic marker.	τ: $\tau(z) = x_i$.
gen: $\mathsf{gen}(S, z) = (x_{-i}, \mathbf{y}, \text{marginals of } S)$.	sgen: $\mathsf{sgen}(S, z) = (x_{-i}, \text{marginals of } S)$.
A: An estimator w.r.t. $h, x_{-i}, \mathbf{y}$, marginals of S.	A^*: An estimator w.r.t. h, x_{-i}, marginals of S.

TABLE I: Warfarin-dosage Attack of Fredrikson et al. [6]. We describe how to set up various parameters in order to put the attack of [6] in our methodology. Note that in this formulation z takes the form of (x, y) and we feed y to the adversary (not $h(x)$). This is important because in Fredrikson et al.'s case, for the patients participating in the dataset, y does not come from model output, but rather is determined by medical doctors. Thus $h(x)$ is only an approximation of y.

important because in Fredrikson et al.'s case, for the patients participating in the dataset, y does not come from model output, but rather is determined by medical doctors. Thus $h(x)$ is only an approximation of y. □

Example 2 (Inferring Participation)**.** *A common privacy attack is to infer whether an individual is in a dataset. For example, differential privacy addresses such attacks, and uses noise to hide the participation of any* individual *(so, with/without a specific individual, the adversary draws the same conclusion with high probability.)*

We note that participation attacks fit naturally into our methodology. In particular, consider the following goal function τ, for $z \in Z$,

$$\tau(z) = \begin{cases} 1 & z \in S, \\ 0 & \textit{otherwise.} \end{cases}$$

That is, given $z \in Z$, the goal of the adversary is to decide whether z is in the training set or not.

One may think that differential privacy is precisely the countermeasure for this attack. However, in principle it is not, although applying differential privacy may have certainly effect the outcome. This is because the design of differential privacy allows *learning correlations, subject only to that any* individual participation *will* not *be able to change the correlation significantly. Therefore, once the correlation is found, one may still be able to use this correlation to infer participation of a population with certain accuracy. Nonetheless differential privacy ensures that* localized *to any particular individual, his or her participation will not significantly change the results of such inferences (so the guarantee here is a form of "plausible deniability" for that particular individual). We remark that it would be interesting to carry out this attack empirically in a real-world setting.* □

B. White-Box MI

We now move on to consider white-box MI attacks. We will now assume that the adversary has some additional knowledge about the model structure. The question is, however, how to model this knowledge about the structure?

We observe that machine learning models typically adopt a sequential composition of computations. For example, in the simplest case of linear models, one first computes a linear representation of the features, and then applies, for example, a logistic function to make a prediction (representing a probability in this case). As another example, in "one-vs-all" multiclass logistic regression, one trains multiple binary logistic regression models, each encoding the "likelihood" of a particular class, and then makes a final prediction based on these confidence information. As observed in [7], being able to observe such intermediate information, even though they might be highly compressed compared to the original information, can give the adversary a significant advantage in deducing sensitive values.

We are thus motivated to consider white-box MI attacks in the particular case of sequential composition. We note that this is in sharp contrast with attacks in cryptographic settings where the protocols typically have a significantly more complicated composition structure (compared to sequential composition). We start by defining machine learning models with k layers.

Definition 1 (k-Layer Model)**.** *Let X be a feature space and $Y_1, \ldots, Y_k$ be k output spaces. A k-layer model M is a model where its computation can be represented as a composition of k functions $h_1, \ldots, h_k$, where $h_1 : X \mapsto Y_1$ and $h_i : Y_{i-1} \mapsto Y_i$ ($2 \leq i \leq k$). The output of h_k is the output of the entire model.*

We can now define white-box MI attacks. Compared to the black-box case, the only thing changes now is that: (i) the adversary is aware of the composition structure, and (ii) he might be able to observe intermediate information passing between the layers. We have the following definition.

Methodology 2 (k-layer White-Box MI Attack)**.** The methodology of white-box MI attack is the same as the black-box one, except for two differences:

- Instead of letting the adversary A have oracle access to the model, we feed *the k-layer representation* of the machine-learning model as input to the adversary.
- The auxiliary information generators gen and sgen take an additional parameter Γ (the learning algorithm). This allows the auxiliary information generators to generate information that might depend on the learning algorithm Γ (see Example 4). An important point to note is that in the simulated world the adversary still *does not* have access to the model $\Gamma(S)$.

Modeling Examples. As in the black-box case, we now express existing attacks using our methodology.

Example 3 (Decision Tree [7])**.** *In [7] the authors studied the following attack against decision trees. Not only does the adversary know the model structure, but also he knows, for each path of the tree, how many instances in the training set correspond to that path. In other words, the adversary knows both the exact decision tree, as well as the confidence information of each path (intuitively, one wants to follow the path that more training instances follow). They show that with such confidence information one can significantly improve attack accuracy.*

Such a scenario can be captured by our methodology. The decision tree model is directly fed as input to the adversary. For the confidence information, the adversary can compute on its own by simulating the model on every instance of S. The adversary can do so because he is all-powerful and has white-box access to the model.

Example 4 (Neural Network [7])**.** *As we mentioned before, [7] also studied another attack where for a neural network with a softmax layer (this layer encodes the probabilities corresponding to each class), the adversary can query for probability in that layer. Again, accessing this piece of information significantly improves attack accuracy.*

This attack can also be easily captured by our methodology. One potential subtlety is the softmax probabilities, which cannot be computed by the adversary directly, though he has white-box access to the model. This is because he only knows partially the original input. Nevertheless, this can be generated by the auxiliary information generator by simulating $\Gamma(S)$ on z and encode the output of the softmax layer in the auxiliary information.

Interestingly, we observe that several privacy attacks [8], [9], [10] for recovering image features, which appeared before the work of Fredrikson et al. [6], can also be captured using white-box MI attacks in a similar way. Due to lack of space, we defer a detailed description to the full version of this paper.

III. Preliminaries

We now present technical preliminaries to facilitate our later development. In the first part of this section we give some background on Boolean analysis, which will be needed for our development of black-box MI attacks. In the second part we present some assumptions we put on the machine learning models.

Boolean Analysis. We need some elementary concepts from Boolean analysis. More details regarding these concepts can be found in O'Donnell [11]. In Boolean analysis, a Boolean function $f : \{-1,1\}^n \mapsto \{-1,1\}$ is viewed as a 2^n-dimensional real vector, and we consider an inner product space of these vectors, where the inner product is defined as $\langle f, g\rangle = \mathbb{E}_{x\sim\{-1,1\}^n}[f(x)g(x)]$. A central concept of Boolean analysis is its Fourier expansion, where the Fourier basis is the set of all parity functions $\Omega = \{\chi_S : S \subseteq [n]\}$ where $\chi_S(x) = \prod_{i\in S} x_i$ is the parity function of bits in S. Any function f can be represented as $f = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S$ where $\widehat{f}(S)$ is called the Fourier coefficient of f at S.

Definition 2 (Influence)**.** *Let $f : \{-1,1\}^n \mapsto \{-1,1\}$ and $i \in [n]$. The influence of i-th coordinate of f is $\mathbf{Inf}_i(f) = \Pr_{x\sim\{-1,1\}^n}[f(x) \neq f(x^{\oplus i})]$ where $x^{\oplus i}$ means to flip the i-th bit of x.*

Influence is related to the *difference operator* D_i.

Definition 3. *D_i is a linear operator applied to a Boolean function such that $(\mathrm{D}_i f)(x) = \frac{f(x^{i\to 1})-f(x^{i\to -1})}{2}$. Here $x^{i\to 1}$ means we set the i-th bit of x to 1.*

Definition 4. *Let $b \in \{-1,1\}$ and $-1 \leq \rho \leq 1$. A random bit b' is ρ-correlated with b if*

$$b' = \begin{cases} b & \text{w.p. } \frac{1}{2} + \frac{\rho}{2} \\ -b & \text{w.p. } \frac{1}{2} - \frac{\rho}{2} \end{cases}$$

We write it as $b' \sim N_\rho(b)$. As ρ tends to 1, b' is more likely to be b.

We say that z and x are ρ-correlated if each z_i is drawn independently from $N_\rho(x_i)$, for $i \in [n]$. In such a case, we write it as $z \sim N_\rho(x)$.

Definition 5 (Noise Stability)**.** *Let $-1 \leq \rho \leq 1$. The ρ-noise stability of f, denoted as $\mathbf{Stab}_\rho[f]$, is defined to be $\mathbf{Stab}_\rho[f] = \mathbb{E}_{\substack{x\sim\{-1,1\}^n \\ y\sim N_\rho(x)}}[f(x)f(y)]$.*

Definition 6 (Stable Influence)**.** *Let* $0 \leq \rho \leq 1$*. The* ρ*-stable influence of* f *at* i*, denoted as* $\mathbf{Inf}_i^{(\rho)}[f]$*, is defined to be* $\mathbf{Inf}_i^{(\rho)}[f] = \mathbf{Stab}_\rho[\mathrm{D}_i f] = \mathbb{E}_{\substack{x\sim\{-1,1\}^n \\ y\sim N_\rho(x)}}[\mathrm{D}_i f(x)\,\mathrm{D}_i f(y)]$*. Note that when* $\rho = 1$*, this reduces to* $\mathbf{Inf}_i[f]$*.*

Definition 7 (Noise Operator)**.** *Let* $-1 \leq \rho \leq 1$*. The noise operator* T_ρ *is defined as* $\mathrm{T}_\rho f(x) = \mathbb{E}_{y\sim N_\rho(x)}[f(y)]$*.*

The following lemma gives some elementary properties of the noise operator and stable influence.

Lemma 1 (O'Donnell[11])**.** *We have the following*

- T_ρ *is a linear operator.*
- $\mathrm{T}_\rho f = \sum_{S\subseteq[n]} \rho^{|S|}\widehat{f}(S)\chi_S$*.*
- $\mathbf{Stab}_\rho[f] = \langle f, \mathrm{T}_\rho f\rangle = \sum_{S\subseteq[n]} \rho^{|S|}\widehat{f}(S)^2$*.*

Model Inversion. Because our functions are models learned from collected data, we will make the following assumption on the models:

Definition 8 (No Trivial Feature Assumption)**.** *Let* $f : \{-1,1\}^n \mapsto \mathbb{R}$ *be a model learned from data. The* no trivial feature assumption *states that every feature has nontrivial influence. That is, for any* $i \in [n]$*,* $\mathbf{Inf}_i[f] > 0$*.*

The following simple proposition shows that if $\mathbf{Inf}_i[f] = 0$, then one can obtain an "equivalent" function over the Boolean cube without x_i.

Lemma 2. *Consider any* $f : \{-1,1\}^n \mapsto \{-1,1\}$*. Suppose that* $\mathbf{Inf}_i[f] = 0$*, then there exists another function* g *which maps* $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$ *to* $\{-1,1\}$*, such that* $\mathbf{Inf}_j[g] = \mathbf{Inf}_j[f]$ *for any* $j \neq i$*.*

IV. Black-Box MI Attacks

In this section we study black-box MI attacks. Due to lack of space, we will focus on the simplest possible models – binary classification where all the features are binary as well. In the full version of this paper, we also extend the results to binary classification over generalized but finite domains.

Recall that our main technical goal is to isolate important factors that can affect model invertibility. Unfortunately, our formulation in Section II is quite complex so many factors may play a role. For example, intuitively the training process may have an impact – if we know that a model is trained using linear regression, would this give us an advantage? On the other hand, if one thinks about MI attacks at the application phase, where a model is fixed anyway, then it suggests that invertibility should be *independent* of the training.

To gain more understanding, we choose to start with simple scenarios where we can characterize model invertibility exactly. Interestingly, even these very abstract and seemingly oversimplified scenarios provide insights to our main question. Perhaps more importantly, they also give rise to intriguing and natural questions that provide ample scope for future.

Specifically, in this section we specialize Methodology 1 in the following ways:

- We consider a Boolean model $h : \{-1,1\}^n \mapsto \{-1,1\}$ in the test phase.
- We assume that the model invertibility is evaluated over the uniform distribution. That is, we assume that a feature vector is drawn uniformly from $U_{\{-1,1\}^n}$.
- We consider two simple auxiliary information generators. In the first, *noiseless generator* gen_1,

$$\mathsf{gen}_1(S,(x,y)) = (x_{-i},y).$$

In the second *independent perturbation generator*,

$$\mathsf{gen}_\rho(S,(x,y)) = (z_{-i},y)$$

where each bit of z_{-i} equals that of x_{-i} with probability $\frac{1}{2}+\frac{\rho}{2}$, and is flipped otherwise. Note that for $\rho = 1$ it degenerates to our noiseless generator.

Under these specializations our main results are summarized as follows[1].

1) In the noiseless case, we characterize model invertibility using the influence of a Boolean function. Interestingly, it turns out in this case, model invertibility is *independent* of the training. These results are presented in Section IV-A.
2) In the noisy case, we show that model invertibility is related to the stable influence of a Boolean function, though stable influence does not exactly capture the invertibility. These results are presented in Section IV-B.
3) Interestingly, we find that under noise, there is an interesting phenomenon where a *highly invertible model quickly becomes highly non-invertible with only a little noise*. We study this phenomenon under the name "invertibility interference." The results are presented in Section IV-C.

A. Model Invertibility with No Noise

We now specialize our definition of black-box MI attacks (Definition 1) to the noiseless scenario. Note that this definition is a direct abstraction of the MI attack mentioned in Example 1.

[1] Due to lack of space all the proofs are put in the appendix.

Definition 9 (Noiseless Uniform Black-Box attack)**.** *Let (Z, H, ℓ) be a learning problem where H consists of hypotheses of the form $\{-1,1\}^n \mapsto \{-1,1\}$. Let Γ be a learning algorithm and S be a training set. For simplicity we denote $\Gamma(S)$ as h.* Noiseless Uniform Black-Box MI attack for coordinate i *is the following game. Let A be a probabilistic algorithm with binary output, then*

i. *Nature draws (x, y) from $\mathscr{D}_U$. That is, nature draws $x \sim \{-1,1\}^n$, and set $y = h(x)$.*
ii. *Nature presents*

$$\mathsf{gen}_1(S,(x,y)) = (x_{-i}, y)$$

to the adversary.
iii. *Adversary outputs $A^{\Gamma(S)}(x_{-i}, h(x))$.*

The gain *of this game is $\Pr[A^{\Gamma(S)}(x_{-i}, h(x)) = x_i]$, where the probability is over samples $x_1, \ldots, x_n$, and the randomness (if any) of the adversary. For simulation, sgen_1 is defined as $\mathsf{sgen}_1(S,(x,y)) = x_{-i}$. Because x_i is independently and uniformly drawn from $\{-1,1\}$, so for any simulated attack A^*, $\Pr[A^*(x_{-i}) = x_i] = 1/2$. Therefore, the* advantage *of the game is defined to be $\Pr[A^{\Gamma(S)}(x_{-i}, h(x)) = x_i] - 1/2$.*

For this type of MI attack, we consider the following *deterministic* algorithm,

Input: $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$. $y \in \{-1,1\}$. Oracle access to f.
Output: $b \in \{-1,1\}$.
1 Compute $y_1 = f(x_1, \ldots, x_{i-1}, -1, x_{i+1}, \ldots, x_n)$, and
2 $y_2 = f(x_1, \ldots, x_{i-1}, +1, x_{i+1}, \ldots, x_n)$.
3 If $y_1 \neq y_2$, then if $y_1 = y$, output -1, otherwise output $+1$.
4 Otherwise, output the constant 1.

Algorithm 1: A Deterministic Algorithm for Noiseless Uniform Flat MI Attack.

We have the following simple lemma.

Lemma 3. *Let A_1 denote Algorithm 1. Then $\mathsf{gain}(A_1, \mathsf{gen}_1, x_i, \mathscr{D}_U, S, \Gamma) = \frac{1}{2} + \frac{\mathbf{Inf}_i[\Gamma(S)]}{2}$, and so the advantage is $\frac{\mathbf{Inf}_i[\Gamma(S)]}{2}$. Further this gain is optimal.*

While this lemma is trivial to prove, an interesting observation regarding it is that the invertibility is *independent of the training*. That is, no matter what Γ and S are (and what distribution S is drawn from), the invertibility is *characterized* by influence, which is an intrinsic property of the model itself.

For noiseless MI attacks, it is also easy to characterize the most and least invertible functions. In the following recall that we assume the nontrivial feature assumption.

Most Invertible Functions. With the no trivial feature assumption, the most invertible function is $\chi_{[n]}(x) = \prod_{i=1}^n x_i$, where every coordinate has influence 1, and so the advantage is $1/2$.

Least Invertible Functions. What functions are least invertible if we measure the invertibility by the maximum influence $\mathbf{MaxInf}_i[h]$? In this direction, a natural candidate is the majority function. Indeed, using Stirling's formula (see Exercise 2.22 [11].), one can estimate that $\mathbf{Inf}_i[\mathrm{MAJ}_n] \approx O(1/\sqrt{n})$ for every $i \in [n]$. There are functions with much smaller influence. For example,

$$\mathrm{OR}_n(x) = \begin{cases} 1 & x_1 = x_2 = \cdots = x_n = 1. \\ -1 & \text{otherwise.} \end{cases}$$

Then it is easy to check that $\mathbf{Inf}_i[\mathrm{OR}_n] = 2^{1-n}$ for every $i \in [n]$. We can also characterize the structure of the least invertible functions. Under the no trivial feature assumption, these functions are those that are "constant except at one point."

Lemma 4. *Consider any $h : \{-1,1\}^n \mapsto \{-1,1\}$. If $\mathbf{Inf}_i[h] > 0$, then $\mathbf{Inf}_i[h] \geq 2^{1-n}$.*

Lemma 5. *Let $h : \{-1,1\}^n \mapsto \{-1,1\}$ be a Boolean function. If $\mathbf{Inf}_i[h] = 2^{1-n}$ for some $i \in [n]$, then $\mathbf{Inf}_i[h] > 0$ for every $i \in [n]$.*

Theorem 1. *Let $h : \{-1,1\}^n \mapsto \{-1,1\}$ be a Boolean function. Then h satisfies the property that for every $i \in [n]$, $\mathbf{Inf}_i[h] = 2^{1-n}$ if and only if h is constant except at a unique point x_0. In other words, there exist $x_0 \in \{-1,1\}^n$ and $b \in \{-1,1\}$ such that*

$$h(x) = \begin{cases} b & \text{if } x = x_0, \\ -b & \text{otherwise.} \end{cases}$$

Recall that a Boolean-valued function is *unanimous* if $h(1, \ldots, 1) = 1$ and $h(-1, \ldots, -1) = -1$. Therefore we have the following two corollaries,

Corollary 1. *OR_n and AND_n are the only unanimous Boolean functions where maximum influence is 2^{1-n}.*

Corollary 2. *OR_n and AND_n are the only monotone Boolean functions where maximum influence is 2^{1-n}.*

B. Model Invertibility with Independent Noise

We now move on to the independent perturbation case.

Definition 10 (ρ-Independent Perturbation Uniform Black-Box MI Attack)**.** *Let (Z, H, ℓ) be a learning problem where H consists of hypotheses of the form $\{-1,1\}^n \mapsto \{-1,1\}$. Let Γ be a learning algorithm and S be a training set. For simplicity we denote $\Gamma(S)$ as*

h. ρ-Independent Perturbation Uniform Black-Box MI attack *for coordinate* i *is the following game. Let* A *be a probabilistic algorithm with binary output, then*

i. *Nature draws* (x, y) *from* $\mathscr{D}_U$*. That is, nature draws* $x \sim \{-1,1\}^n$*, and set* $y = h(x)$*.*

ii. *Nature presents*

$$\mathsf{gen}_\rho(S,(x,y)) = (z_{-i}, y)$$

to the adversary.

iii. *Adversary outputs* $A^{\Gamma(S)}(z_{-i}, y)$*.*

The gain *of this game is* $\Pr[A^{\Gamma(S)}(z_{-i}, y) = x_i]$*, where the probability is over samples* $x_1, \ldots, x_n$*, the randomness of* gen_ρ*, and the randomness (if any) of the adversary. For simulation,* sgen_ρ *is defined as* $\mathsf{sgen}_\rho(S, x_{-i}, y) = z_{-i}$*. Because* x_i *is independently and uniformly drawn from* $\{-1,1\}$*, so for any simulated attack* A^**,* $\Pr[A^*(z_{-i}) = x_i] = 1/2$*. Therefore, the* advantage *is defined as* $\Pr[A^{\Gamma(S)}(x_{-i}, y) = x_i] - 1/2$*.*

We now consider the following algorithm for performing MI attack. The algorithm is the same as Algorithm 1, we repeat it here and note that now the input to the algorithm is z_{-i}, instead of x_{-i}.

Input: $z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n$. $y \in \{-1,1\}$. Oracle access to f.
Output: $b \in \{-1,1\}$.
1 Compute $y_1 = f(z_1, \ldots, z_{i-1}, -1, z_{i+1}, \ldots, z_n)$, and
2 $y_2 = f(z_1, \ldots, z_{i-1}, +1, z_{i+1}, \ldots, z_n)$.
3 If $y_1 \neq y_2$, then if $y_1 = y$, output -1, otherwise output $+1$.
4 Otherwise, output the constant 1.

Algorithm 2: A Deterministic Algorithm for ρ-Perturbation Uniform Singleton Flat MI Attack.

Nicely, the gain of this algorithm is exactly the so called *stable influence*. Intuitively this is clear: Recall from Definition 6 that stable influence is defined as $\mathbb{E}_{x\sim\{-1,1\}^n, z\sim N_\rho(x)}[\mathrm{D}_i f(x)\,\mathrm{D}_i f(z)]$. Thus if $\mathrm{D}_i f(x)$ and $\mathrm{D}_i f(z)$ are of the same sign then Algorithm 2 guessed correctly. If the signs are different, then the guess is incorrect. Otherwise, one can show that the gain is $1/2$. Formally, we have the following theorem.

Theorem 2. *Let* $\rho \in [0,1]$*. Let* A_ρ *denote Algorithm 2. Then*

$$\mathsf{gain}(A_\rho, \mathsf{gen}_\rho, x_i, \mathscr{D}_U, S, \Gamma) = \frac{1}{2} + \frac{\mathbf{Inf}_i^{(\rho)}[h]}{2}$$

where $\mathbf{Inf}_i^{(\rho)}[h]$ *is the* ρ*-stable influence of* h *(See Definition 6) at* i*-th coordinate.*

For $\rho = 1$, $\mathbf{Inf}_i^1[h] = \mathbb{E}[\mathrm{D}_i f(x)\,\mathrm{D}_i f(z)] = \mathbb{E}[\mathrm{D}_i f(x)^2] = \mathbf{Inf}_i[h]$. We thus get back the influence in the noiseless model of Lemma 3.

Remark 1 (On Optimality)**.** *Unfortunately, we note that for the* ρ*-independent perturbation model, Algorithm 2 no longer achieves the maximum possible gain. That is, there exists some model* $h : \{-1,1\}^n \mapsto \{-1,1\}$ *and some inversion algorithm* A'*, such that the gain of* A' *is larger than* $(1 + \mathbf{Inf}_i^{(\rho)}[h])/2$*. The intuition is that, since the adversary knows* $h(x)$ *exactly, so it can leverage the function table of* h *to "de-noise", For example, consider* OR_n*. As long as we see that the model output is* 1*, we know that all input bits are* 1*. Therefore, the advantage we can achieve, even in the independent perturbation model, is* $\mathbf{Inf}_i[\mathrm{OR}_n]/2 = 2^{-n}$ *for any* $0 \leq \rho \leq 1$*. On the other hand, the advantage of Algorithm 2 is* $\rho^{n-1}2^{-n}$*.*

Question 1. *Consider* ρ*-independent perturbation model. Let* A_ρ *denote Algorithm 2. Is it the case that for any* $h : \{-1,1\}^n \mapsto \{-1,1\}$*, and any probabilistic algorithm* A'*,*

$$\begin{aligned} &\mathsf{gain}(A', \mathsf{gen}_\rho, x_i, \mathscr{D}_U, S, \Gamma) \\ \leq\ &\mathsf{gain}(A_\rho, \mathsf{gen}_\rho, x_i, \mathscr{D}_U, S, \Gamma) + o_n(1)\ ? \end{aligned}$$

C. Invertibility Interference

Intuitively it is clear that noise will incur a decay on the gain of inversion. Theorem 2 quantifies this intuition using stable influence. For example, as we saw in the above, the gain of a natural algorithm (Algorithm 2) on OR_n goes from $\mathbf{Inf}_i[\mathrm{OR}_n]$ to $\rho^{n-1}\mathbf{Inf}_i[\mathrm{OR}_n]$, which is exponentially small in the influence. However, this example is not very interesting in the sense that the influence of OR_n is already very small (2^{1-n}) in the noiseless case.

A more interesting phenomenon regarding noise is that highly invertible models in the noiseless case quickly becomes highly non-invertible due to a little noise. The reason behind is that multiple influential coordinates interfere with each other under noise. Let us see an example. In the noiseless model the most invertible function is the parity function $\chi_{[n]} = \prod_{i\in[n]} x_i$. In this case, $\mathbf{Inf}_i[f] = 1$ for every i. On the other hand, under independent noise, the invertibility of χ_n becomes $\mathbf{Inf}_n^{(\rho)}[\chi_{[n]}] = \mathbf{Stab}_\rho[\mathrm{D}_n \chi_{[n]}] = \langle \chi_{[n-1]}, \mathrm{T}_\rho \chi_{[n-1]} \rangle = \rho^{n-1}$. Therefore the invertibility decays exponentially fast in n.

We term this phenomenon "invertibility interference:" When noise presents, if one does not know one of these influential coordinates exactly, then he cannot effectively invert the model to deduce the target feature. What is the

stable influence if we have t influential coordinates? In this direction, we have the following simple result:

Theorem 3. *Suppose that $h : \{-1,1\}^n \mapsto \{-1,1\}$ has t coordinates with influence 1. Let $0 < \rho \leq 1$, then for any $i \in [n]$, $\mathbf{Inf}_i^{(\rho)}[h] \leq \rho^{t-1}\mathbf{Inf}_i[h]$.*

Question 2. *If, instead of having coordinates of influence 1, we are only guaranteed that individual influence is lower bounded by $1-\delta$ for some $\delta > 0$, how fast will the stable influence decay with respect to δ?*

V. White-Box MI Attacks

We now move on to study white-box MI attacks. As discussed before, we assume that the computation of the models follows a sequential composition. This thus gives a natural view of MI attacks as *communication games*: One can think of *each layer of the model* as a player who sends a message to the next player, and the adversary as *another player* who observes the model output and has some additional information. Together, the goal of this game is to compute some function τ. This view gives a natural question:

"How would knowing the communication structure and (possibly) observing some intermediate information in the communication help MI attacks?"

Empirically, the answer is that it helps a lot. As mentioned earlier, it is essential for white-box attacks as studied in [7] to have access to the auxiliary confidence information, which makes the inversion algorithm much more effective.

The main purpose of this section is to give *theoretical justifications* for these empirical observations. At a high level, our results are summarized as follows:

1) Similar to our study of black-box MI attacks, we choose to specialize our methodology so as to obtain theoretical insights. To do so, we focus on white-box MI attacks on *decision trees*. An advantage of studying attacks on decision trees is its simplicity: The communication channel is very restricted, not only it is a sequential composition, but also in each iteration a player only reads a single bit of the input, and decides a binary output.
2) We show how to interpret white-box MI attacks on decision trees as alternating (communication) games. Specifically, these are communication games where the communication channel is one-way, unicast (following the sequential composition), and players *alternatively* hold two inputs. We give examples showing that these communication games are very restricted.
3) We show that, however, even when restricting these communication games to have 1 bit of communication between two neighboring players, it is still the case that for *any* goal function τ, there exists a game with enough players (corresponding to a machine learning model with enough many layers), such that there is an adversary who can compute τ correctly *everywhere*. This result illustrates the unexpected computational power of a restricted communication game, and in particular, that the leakage can be significant even in a very restricted white-box case.

We now give more details in the rest of this section.

A. Decision Trees, MI Attacks, and Alternating Games

From now on we consider *oblivious decision trees*, which are decision trees in which the same feature is examined at each level (of the tree). This restricts the machine learning models we consider (it is even a subclass of decision trees). Note that, however, the more we restrict the model (and its communication), the stronger our conclusion (regarding leakage in the white-box case) is if we can show significant information leakage.

We have mentioned that for a model with sequential composition, the communication channel is restricted: it is *one-way* and *unicast*. That is, each player only sends one message to the next player, in a fixed order. This is in sharp contrast with communication games studied in typical communication complexity literature [12], [13], [14], [15], [16], [17], [18], where the channel is either bidirectional or the messages are broadcasted.

We note that for oblivious decision trees, such communication games are further restricted. Consider an adversary who knows part of the input to the decision tree, then the communication game *alternates* between input he knows and input he does not know. Specifically, suppose that the input to the decision tree is $z \in \{0,1\}^n$, and assume that the adversary can see the bits at positions $K \subseteq [n]$ (K stands for "known" positions), then, without loss of generality, the communication game can be viewed as: the first player examines several variables at positions in K, then sends a bit to the next player, who then examines several variables in $[n] \setminus K$, and so on. The final player, which is the adversary (who knows bits in K), determines an output.

The following definition captures our discussion so far mathematically:

Definition 11 (Alternating MI Attacks (AMI Attacks))**.** *Let n, ℓ be natural numbers. Let $k \geq 3$ be also a natural number. In Alternating MI Attack there are $(k-1) \geq 2$ functions: $h_1, \ldots, h_{k-1}$ in the form of $h_1 : \{0,1\}^n \mapsto$*

$\{0,1\}^\ell$ *and* $h_i : \{0,1\}^n \times \{0,1\}^\ell \mapsto \{0,1\}^\ell \quad (i = 1,\ldots,k-1)$. *let* $h^{(1)},\ldots,h^{(k-1)} : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}^\ell$ *be the following sequence:*

$$h^{(1)}(x,y) = h_1(x)$$
$$h^{(i)}(x,y) = h_i(y, h^{(i-1)}(x,y)) \quad i = 2,4,\ldots$$
$$h^{(i)}(x,y) = h_i(x, h^{(i-1)}(x,y)) \quad i = 3,5,\ldots$$

Let A be a probabilistic algorithm that is an "adversary." Let $\tau : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ be a Boolean function on $2n$ bits. The alternating MI attack proceeds as follows:

i. *Nature samples x, y uniformly random from $\{0,1\}^n$.*
ii. *If k is odd, then nature presents x to A, but not y. Otherwise, nature presents y to A, but not x.*
iii. *Nature also presents the output of $h^{(k-1)}$, that is the output of the "outermost model", to A.*

For odd k, the gain of the alternating MI attack is measured by

$$\Pr[A(x, h^{(k-1)}(x,y)) = \tau(x,y)].$$

Similarly for even k, the gain is defined as $\Pr[A\left(y, h^{(k-1)}(x,y)\right) = \tau(x,y)\big]$. Both probabilities are taken over all the randomness: the randomness of sampling x, y (uniformly), the private randomness of $h_1,\ldots,h_k$, and the randomness of A.

Note that the adversary in this formulation can only see the output of the outer model ($h^{(k-1)}$), beyond knowing part of the input. However, we also want to capture the intuition that the adversary may "inspect" some messages passed between layers in a machine learning model. We thus consider the following modification of the definition:

Definition 12 (Alternating MI Attacks with Early Inspection)**.** *In alternating MI attacks with early inspection, the only difference is that instead of feeding $h^{(k-1)}$ to the adversary, the adversary can* choose once *to inspect the output of $h^{(i)}$ ($1 \le i \le k-1$), and based on that to compute the output.*

Note that we restrict the adversary to be only able to inspect once — this is, again, to pose restriction on the communication, which gives stronger implication on the risk of leakage.

In the above definition of alternating MI attacks, we still need to distinguish between "layers" in a machine model, and the adversary. Towards our main result, which states that for any goal function τ there exists a model (with enough layers) that can allow an adversary to compute τ everywhere, we find that it is more convenient to work with a definition where we do not distinguish between functions inside a model and the function computed by the adversary. This leads to the following definition.

Definition 13 (Alternating One-Way Unicast Communication Games (AOWU))**.** *Let n, k, ℓ be natural numbers. In Alternating One-Way Unicast Communication Games we have:*

(1) *The goal of the communication is to compute some function $\tau : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$.*
(2) *There are k players, $P_1,\ldots,P_k$. These players are allowed to use private randomness.*
(3) *The players communicate in the way of* one-way unicast*. Namely, they play in the fixed order of $P_1,\ldots,P_k$, and player P_i is only allowed to send one message, a bit string of length ℓ, to player P_{i+1}, for $i = 1,\ldots,k-1$.*
(4) *P_k is required to output a single bit, which is viewed as the output of the protocol.*

Similar to alternating MI attack, one can define the sequence of composed function $P^{(1)},\ldots,P^{(k)}$, where $P^{(i)}$ is the function computed by the first i players on $\{0,1\}^n \times \{0,1\}^n$:

$$P^{(1)}(x,y) = P_1(x)$$
$$P^{(i)}(x,y) = P_i(y, P^{(i-1)}(x,y)) \quad i = 2,4,\ldots$$
$$P^{(i)}(x,y) = P_i(x, P^{(i-1)}(x,y)) \quad i = 3,5,\ldots$$

For AMI Attacks with Early Inspection, we have the following definition,

Definition 14 (AOWU*)**.** *An AOWU game with early stopping, or called an AOWU* game, is an AOWU game where any player P_i, $i \in [k]$, can stop the protocol, and claim his or her output as the output of the protocol.*

From our discussion so far it follows that

Lemma 6. *(n,k,ℓ)-alternating MI attack and (n,k,ℓ)-AOWU games are equivalent. Further, $(n,k,\ell$-alternating MI attack with early inspection and (n,k,ℓ)-AOWU* games are equivalent.*

B. On the Power of AMI Attacks with Early Inspection

We now study the power of alternating MI attacks with early inspection. Clearly, we can equivalently study AOWU games with early stop. We show that, even when restricting to 1 bit of communication, it is still surprisingly powerful.

To motivate this result, let us fist give an example, which illustrates "how restricted" these games are.

Example 5. *Let $\mathrm{IP}(x,y)$ be the inner product of x and y, that is for $x, y \in \{0,1\}^n$, $\mathrm{IP}(x,y) = \bigoplus_{i=1}^n (x_i \wedge$*

y_i). *Consider one-way unicast alternating games that try to compute* $\mathrm{IP}(x, y)$. *The communicated messages are restricted to be of* 1*-bit long (that is* $\ell = 1$*).*

Let us consider the simple case that $n = 2$*. That is, we want to compute the inner product of two length-*2 *bit strings. Note that in the traditional two-player communication model where Alice holds* x *and Bob holds* y*, then there is a trivial protocol where Alice sends to Bob* 2 *bit messages, one* x_1 *and one* x_2*, so that Bob then has complete knowledge of* x *and can compute* any function *on* x, y*.*

However, with AOWU games, there is a now a difficulty. Suppose that P_1 *sends* x_1 *to* P_2*, and* P_2 *computes* x_1y_1*. Then what will* P_2 *send to* P_3*? If* P_2 *sends* y_2 *to* P_3*, then the progress that* P_2 *has made is essentially lost. However, if he sends* x_1y_1*,* P_3 *still does not know any information about* y_2*. Therefore, at least with* 2 *bits communication they cannot solve the problem. What is the "right" lower bound on the number of players that are needed in order to compute inner product with* 1 *bit of communication?* □

We now construct a "universal" protocol that can compute any $\tau : \{0,1\}^n \times \{0,1\}^n$ using an AOWU* protocol with 1 *bit of communication*.

Construction 1 (Universal-1 Protocol)**.** *Let* $\tau : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$ *be any function. Consider the following protocol:*

- *There are* $2 \cdot 2^n = 2^{n+1}$ *players, split into pairs,* $(1,2)$*,* $(3,4)$*,* $(2i-1, 2i), \ldots, (2^{n+1}-1, 2^{n+1})$*. Note that odd-numbered players hold* x*, and even-numbered players hold* y*.*
- *Player* $2i-1$ *sends the following bit to player* $2i$*: He sends* 1 *if* x *is the lexicographically the* i*-th smallest string of all binary strings of length* n*. For example, player* 1 *sends* 1 *to player* 2 *if* $x = 0$*, and sends a bit* 0 *otherwise.*
- *For player* $2i$*, if she receives a bit* 1*, then she can be certain about the value of* x*. Because she also knows* y*, she can compute* $\tau(x, y)$*, stops the protocol early by asserting the special stopping bit, and claims the output. Otherwise, she keeps the special stop bit as* 0 *to indicate player* $2i+1$ *to continue the protocol.*

Note that early stopping is essential here, otherwise player $2i+1$ cannot distinguish between a "value" that is for computing τ and a "signal" which indicates that the computation is already done. Following the construction we immediately have the following theorem,

Theorem 4. *Universal-*1 *protocols compute any* τ *with* 1*-bit message and* 2^{n+1} *players.*

Thus we have obtained the claimed main result regarding alternating MI attacks with early inspection.

Theorem 5. *For any goal function* $\tau : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$*, there exists a machine learning model with* $O(2^n)$ *layers, and an alternating MI attack with* 1*-bit communication, that computes* τ *correctly everywhere.*

We close this section with two open questions.

Question 3. *For* 1*-bit communication, is universal-*1 *protocol essentially optimal in the sense that there is a function* $\tau : \{0,1\}^n \times \{0,1\}^n$ *where any protocol computing* τ *requires* $\Omega(2^n)$ *rounds of communications?*

Question 4. *For* 1*-bit communication, is there a universal AOWU protocol (instead of AOWU***) that computes every function* $\tau : \{0,1\}^n \times \{0,1\}^n \mapsto \{0,1\}$*?*

VI. Connections with Other Cryptographic Notions

In this section we compare MI attack with two classic cryptographic primitives: Hard-Core Predicate and Secure Multiparty Computation. We assume that the readers have some basic familiarity with cryptographic terminologies.

Connection with Hard-Core Predicate. Let us first recall the definition of hard-core predicate

Definition 15 (Hard-Core Predicate [19])**.** *Let* U_n *be a uniform distribution over* $\{0,1\}^n$*, and* $f : \{0,1\}^* \mapsto \{0,1\}^*$*. A polynomial time computable predicate* $b : \{0,1\}^* \mapsto \{0,1\}$ *is called a* hard-core *of* f *if for every probabilistic polynomial time algorithm* A'*, there exists a negligible function* $\mu(\cdot)$ *such that for all sufficiently large* n*'s*

$$\Pr_{x \sim U_n}[A'(f(x)) = b(x)] \leq \frac{1}{2} + \mu(n).$$

By viewing f as a "model", one can then simulate this definition by a black-box MI attack. Specifically, let $\mathsf{out}(x) = b(x)$, which is to compute a single bit. The joint distribution is $J_U = (U_n, f(U_n))$. In the real world, given $x \sim U_n$, the auxiliary information generator gen gives the advice string $\mathsf{gen}(d_f, x, f(x)) = f(x)$ to the adversary A'. One can then observe that the gain of A' in the real world, $\mathsf{gain}_{J_U,f,b}(\mathsf{gen}, A')$, is exactly $\Pr_{(x,y)\sim J_U}[A'(y) = b(x)] = \Pr[A'(f(U_n)) = b(U_n)]$. Therefore the goal of hard-core predicate is to find a "hard" predicate b so that for any adversary A', any negligible function $\mu(\cdot)$, and all sufficiently large n's, the gain $\mathsf{gain}_{J_U,f,b}(\mathsf{gen}, A') \leq 1/2 + \mu(n)$.

Following this simulation one can also observe two notable differences: First, in an MI attack an adversary typically has more auxiliary information than the case of hard-core predicates (which only sees the function output). For example, in a black-box MI attack as defined in Definition 9, an adversary has information about all except one feature. Second, we note that the goal of MI attacks is not to construct hard predicates. Rather, its goal is somewhat its "dual": The purpose is to study the leakage of certain model with respect to computing some output function. For statistical output, understanding this leakage may help us decide which information is publishable.

Connection with Secure Multiparty Computation (SMC). A more interesting notion to compare with is the Secure Multiparty Computation (SMC). To this end, we recall first the definition of m-party secure protocols

Definition 16 (m-party secure protocols – sketch [20])**.** *Let f be an m-ary functionality and Π be an m-party protocol operating in the real model.*

- *For a real-model adversary A, controlling some minority of the parties (and tapping all communication channels), and an m-sequence $\overline{x}$, we denote by $\mathrm{REAL}_{\Pi,A}(\overline{x})$ the sequence of m outputs resulting from the execution of Π on input $\overline{x}$ under attack of the adversary A.*
- *For an ideal-model adversary A', controlling some minority of the parties, and an m sequence $\overline{x}$, we denote by $\mathrm{IDEAL}_{\Pi,A}(\overline{x})$ the sequence of m outputs resulting from the ideal process (that they together send input to a trusted third party, which then gives back the output) on input $\overline{x}$ under attack of the adversary A'.*

We say that Π securely implements f with honest majority if for every feasible real model adversary A, controlling some minority of the parties, there exists a feasible ideal model, controlling the same parties, so that the probability ensembles $\{\mathrm{REAL}_{\Pi,A}(\overline{x})\}_{\overline{x}}$ and $\{\mathrm{IDEAL}_{\Pi,A}(\overline{x})\}_{\overline{x}}$ are computationally indistinguishable.

We observe that in this definition the privacy concerns of the *output* is *not* considered. Specifically, it can happen that in the ideal case, upon receiving the output from the trusted third party, one can infer partial information about the input of the other party. Yet such concerns will not factor in the distinguishability as they are contained in the ideal world. Put in another way, the privacy concerns considered in a secure protocol is whether the *communication* among parties in the real world leaks *additional* information compared to the ideal case.

Black-box MI Attacks and SMC. By contrast, for *black-box MI attack*, one considers precisely the privacy concerns of the *output*. That is how much sensitive information an adversary can recover from the output. To this end, one may argue that it is questionable why should one be bothered with the concern of the output, since this is the purpose of the computation.

On one hand, we feel that a fundamental difference here is what information constitutes the output. In the setting of SMC, the output is precisely defined (for example, whether two inputs are equal). However, in the setting of MI attack, the output is statistical and "noisy." For example, a model may carry too much information of some individuals if the learning procedure over-fits. Publishing such models may thus induce effective MI attacks and unwanted disclosure. Studying MI attacks can help us identify and quantify such leakage.

On the other hand, we note that, frequently, additional information is intentionally revealed by an SMC protocol due to performance considerations (for example, revealing the centroid in privacy-preserving clustering or revealing a bit of a honest party's input in the dual-execution SMC protocol [21]). However, the ramifications of leaking this additional information are unclear. Perhaps our framework can be used to address such problems.

More concretely, let us consider a simple example where our current results for MI attacks (though studied in the setting of machine learning) can be applied to SMC. Consider two parties Alice and Bob who jointly compute a Boolean function $f(x, y)$, where x is Alice's input and y is Bob's input. Suppose that the inputs are drawn uniformly from two sets X, Y. Now if Bob is malicious, and can see insensitive information x_{-i} of Alice for $x \sim X$, then with access to f how much better can he guess x_i over X, compared to random guessing?

Let $f_y(y) = f(x, y)$ (i.e. f_y is the specialization f where the second input is y). The answer to this question becomes exactly an MI problem against uniform distribution over X, where the adversary Bob has auxiliary information x_{-i}. Our results in Section IV tells that the advantage is the influence of coordinate i of $f_Y(x) = f(x, y)$.

Therefore, what remains is to estimate the influence of coordinate i of f_Y. In this direction, we note that recent years there has been interesting progress on the algorithmic side of the influence theory. For example, a recent work by Ron, Rubinfeld, Safra, Samorodnistsky and Weinstein [22] proves lower bounds in approximating influence and gives a better upper bound for

monotone Boolean functions. By invoking their influence estimation algorithms (for example, their Algorithm 1), one can thus obtain a quantitative understanding of the risk of outputting the functio f.

White-box MI Attacks and SMC. This situation changes when we consider *white-box MI attack*. Intuitively, in composed MI attack, if one view sub-models as "parties", then we can ask how much information do these compositions (or communication) leak. Therefore, even if one modulo the concerns of the output of the *outermost* model, one could still investigate the *additional* leakage caused by the composition (or communication). This is closer to the goal of SMC.

Unlike SMC, however, is that the communication pattern of the white-box MI attacks is usually much more restricted. For example, composition of models in the usual sense gives "one-way unicast communication", rather than broadcasts, or arbitrary point-to-point communication (so it is not one-way). Indeed, as we saw in the paper, this way of communication induces intriguing and somewhat unexpected connections with communication complexity that does not seem to have been studied before.

REFERENCES

[1] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006, pp. 265–284.

[2] C. Dwork, A. D. Smith, T. Steinke, J. Ullman, and S. P. Vadhan, "Robust traceability from trace amounts," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 650–669.

[3] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. P. Vadhan, "The limits of two-party differential privacy," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 18, p. 106, 2011.

[4] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, 2003, pp. 202–210.

[5] S. P. Kasiviswanathan, M. Rudelson, and A. Smith, "The power of linear reconstruction attacks," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, 2013, pp. 1415–1433.

[6] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proceedings of the 23rd USENIX Security Symposium, San Diego, CA, USA, August 20-22, 2014.*, 2014, pp. 17–32.

[7] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM conference on Computer and communications security*, 2015.

[8] M. Daneshi and J. Guo, "Image reconstruction based on local feature descriptors," *Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, Tech. Rep*, 2011.

[9] E. d'Angelo, L. Jacques, A. Alahi, and P. Vandergheynst, "From bits to images: Inversion of local binary descriptors," *CoRR*, vol. abs/1211.1265, 2012.

[10] H. Kato and T. Harada, "Image reconstruction from bag-of-visual-words," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 955–962.

[11] R. O'Donnell, *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[12] C. Damm, S. Jukna, and J. Sgall, "Some bounds on multiparty communication complexity of pointer jumping," *Computational Complexity*, vol. 7, no. 2, pp. 109–127, 1998.

[13] A. Chakrabarti, "Lower bounds for multi-player pointer jumping," in *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, 2007, pp. 33–45.

[14] M. Braverman, F. Ellen, R. Oshman, T. Pitassi, and V. Vaikuntanathan, "Tight bounds for set disjointness in the message passing model," *CoRR*, vol. abs/1305.4696, 2013.

[15] M. J. Fischer, N. A. Lynch, and M. Paterson, "Impossibility of distributed consensus with one faulty process," *J. ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[16] R. M. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking, "Randomized rumor spreading," in *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, 2000, pp. 565–574.

[17] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, 2003, pp. 482–491.

[18] A. Ganor and R. Raz, "Space pseudorandom generators by communication complexity lower bounds," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, 2014, pp. 692–703.

[19] O. Goldreich, *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.
[20] ——, *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.
[21] Y. Huang, J. Katz, and D. Evans, "Quid-pro-quo-tocols: Strengthening semi-honest protocols with dual execution," in *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*, 2012, pp. 272–284.
[22] D. Ron, R. Rubinfeld, M. Safra, A. Samorodnitsky, and O. Weinstein, "Approximating the influence of monotone boolean functions in o($\sqrt{n}$) query complexity," *TOCT*, vol. 4, no. 4, p. 11, 2012.

APPENDIX

Proof of Lemma 2

Proof. Because $\mathbf{Inf}_i[f] = \sum_{S:i\in S} \widehat{f}(S)^2$, so if $\mathbf{Inf}_i[f] = 0$, this means $\widehat{f}(S) = 0$ for any $i \in S$. In particular, we can set now g to be the following function:

$$g(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n) = \sum_{S\not\ni i} \widehat{f}(S)x^S.$$

$f(x) = g(x_{-i})$ and $\mathbf{Inf}_j[g] = \mathbf{Inf}_j[f]$ for any $j \in [n]$, $j \neq i$. □

Proof of Lemma 3

Proof. For any fixed $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$, if $f(x) \neq f(x^{\oplus i})$, then we guess x_i right with probability 1. Otherwise, conditioned on $f(x) = y$, x_i is uniformly and independently distributed over $\{-1,1\}$. In this case, by constantly guessing 1 the correct probability is $1/2$. This gives the desired gain of Algorithm 1, as well as its optimality. □

Proof of Lemma 4

Proof. If for any input x, $f(x) \neq f(x^{\oplus i})$, then there are at least two inputs, namely $y = x, x^{\oplus i}$ such that $f(y) \neq f(y^{\oplus i})$, this shows the probability $\Pr_{x\sim\{-1,1\}^n}[f(x) \neq f(x^{\oplus i})] \geq 2^{1-n}$. □

Proof of Lemma 5

Proof. Without loss of generality, assume for contradiction that x_1 has influence 0. Then by Lemma 2, there is a function $g(x_2,\ldots,x_n)$ such that $\mathbf{Inf}_j[g] = \mathbf{Inf}_j[f]$ for every $j \geq 2$. Now by Lemma 4, $\mathbf{Inf}_j[f] = \mathbf{Inf}_j[g] \geq 2^{2-n} > 2^{1-n}$ for every $j \geq 2$, contradiction. □

Proof of Theorem 1

Proof. **(⇐)** If f is constant except at a unique point x_0, then for any i, the only inputs x on which $f(x) \neq f(x^{\oplus i})$ are $x = x_0$ and $x = x_0^{\oplus i}$. This proves that $\mathbf{Inf}_i[f] = 2^{1-n}$.

(⇒) We induct on n. If $n = 1$, then $\mathbf{Inf}_1[f] = 1$, so $f(x_1) = x_1$ or $-x_1$ and the result is true. Fix any $n \geq 2$. The induction hypothesis is the following: Let g be a Boolean function on $k \leq (n-1)$ variables. If every coordinate of g has influence 2^{1-k}, then g is constant except at one point.

Now let f be a function on n variables that $\mathbf{Inf}_i[f] = 2^{1-n}$ for every $i \in [n]$. Consider two Boolean functions on $n-1$ variables, $g(x_2,\ldots,x_n) = f(1,x_2,\ldots,x_n)$ and $h(x_2,\ldots,x_n) = f(-1,x_2,\ldots,x_n)$. We claim that the influence of x_2 is 2^{2-n} in one of g and h, and 0 in the other. Indeed, by the assumption that $\mathbf{Inf}_2[f] = 2^{1-n}$, there must be a unique setting $z_{-2} = (z_1,z_3,\ldots,z_n)$ such that $f(z^{2\to 1}) \neq f(z^{2\to -1})$. Note that z_1 is fixed to be 1 or -1, thus the influence of x_2 is 2^{2-n} in g or h, and 0 in the other.

Without loss of generality, suppose $\mathbf{Inf}_2[g] = 2^{2-n}$. Note that g is defined over $n-1$ variables, so by Lemma 5, $\mathbf{Inf}_j[g] > 0$ for every $2 \leq j \leq n$. On the other hand, clearly $\mathbf{Inf}_j[g] \leq 2^{2-n}$ for every $j = 2,\ldots,n$. Thus we can apply the induction hypothesis to conclude that g is constant except one point. Moreover, h is constant. Finally, suppose $g(y) = b$ except $g(y_0) = -b$ where $y_0 \in \{-1,1\}^{n-1}$ and $b \in \{-1,1\}$ is some fixed point. Because $\mathbf{Inf}_1[f] = 2^{1-n}$, so it must be that $h \equiv b$. This shows that f is constant except at one point. □

Proof of Theorem 2

Proof. Let A denote Algorithm 2. Let $\alpha = f(x^{i\to 1}) - f(x^{i\to -1})$ and $\beta = f(z^{i\to 1}) - f(z^{i\to -1})$. Consider the following four disjoint events:

(E1) $\alpha\beta > 0$. Conditioned on E1 happens, A always outputs the correct bit. Thus in this case $\Pr[b = x_i \mid E_1] = 1$.

(E2) $\alpha\beta < 0$. Conditioned on E2 happens, A always output $-x_i$ upon input z_{-i}. Thus in this case $\Pr[b = x_i \mid E_2] = 0$.

(E3) $\beta = 0$. In this case A outputs a uniform random bit. Thus $\Pr[b = x_i \mid E_3] = \frac{1}{2}$.

(E4) $\alpha = 0$ but $\beta \neq 0$. We claim that $\Pr[b = x_i \mid E_4] = 1/2$. Indeed, noting that A is deterministic, writing out all the randomness we have

$$\begin{aligned}&\Pr[b = x_i \mid E_4]\\ =\ &\Pr\Big[A(z_{-i},f(x)) = x_i \;\Big|\\ &f(x^{i\to 1}) = f(x^{i\to -1}),\\ &f(z^{i\to 1}) \neq f(z^{i\to -1})\Big]\end{aligned}$$

Note that the event $E_4 = \{f(x^{i\to 1}) = f(x^{i\to -1}), f(z^{i\to 1}) \neq f(z^{i\to -1})\}$

only depends on x_{-i}, and is independent of x_i, so the above is

$$\begin{aligned}&\Pr[b = x_i \mid E_4]\\ =\ &\frac{1}{2}\Big(\Pr[A(z_{-i}, f(x^{i\to 1})) = 1 \mid E_4]\\ &+\Pr[A(z_{-i}, f(x^{i\to -1})) = -1 \mid E_4]\Big)\end{aligned}$$

Note that conditioned on E_4,

$$\begin{aligned}&\Pr_{\substack{x_{-i}\sim\{-1,1\}^{n-1}\\ z_{-i}\sim N_\rho(x_{-i})}}[A(z_{-i}, f(x^{i\to -1})) = -1 \mid E_4]\\ =\ &\Pr_{\substack{x_{-i}\sim\{-1,1\}^{n-1}\\ z_{-i}\sim N_\rho(x_{-i})}}[A(z_{-i}, f(x^{i\to 1})) = -1 \mid E_4]\end{aligned}$$

This gives that

$$\begin{aligned}&\Pr_{\substack{x_{-i}\sim\{-1,1\}^{n-1}\\ z_{-i}\sim N_\rho(x_{-i})}}[A(z_{-i}, f(x^{i\to 1})) = 1 \mid E_4]\\ +\ &\Pr_{\substack{x_{-i}\sim\{-1,1\}^{n-1}\\ z_{-i}\sim N_\rho(x_{-i})}}[A(z_{-i}, f(x^{i\to -1})) = -1 \mid E_4]\\ =\ &1\end{aligned}$$

Therefore, the probability of guessing correctly is $\Pr[E_1] + \frac{\Pr[E_3]+\Pr[E_4]}{2}$. Observe that $\Pr[E_1] - \Pr[E_2]$ is exactly the ρ-stable influence ($\rho \in [0, 1]$),

$$\begin{aligned}\Pr[E_1] - \Pr[E_2] &= \mathbb{E}[\mathrm{D}_i f(x)\, \mathrm{D}_i f(z)]\\ &= \mathbf{Stab}_\rho[\mathrm{D}_i f]\\ &= \mathbf{Inf}_i^{(\rho)}[f].\end{aligned}$$

Combining with $\Pr[E_1]+\Pr[E_2]+\Pr[E_3]+\Pr[E_4] = 1$, we have that

$$2\Pr[E_1] + \Pr[E_3] + \Pr[E_4] = 1 + \mathbf{Inf}_i^{(\rho)}[f]$$

Dividing by 2 on both sides gives the desired gain. □

Proof of Theorem 3

Proof. We know that $\mathbf{Inf}_i^{(\rho)}[f] = \sum_{S\ni i} \rho^{|S|-1}\widehat{f}(S)^2$. Consider any $S \subseteq [n]$ such that $|S| < t$. We show that $\widehat{f}(S) = 0$. For this let $A = \{x : f(x) = \chi_S(x)\}$. We show that $|A| = |A^c|$ where A^c is the complement of A. Indeed, consider any position $i^* \notin S$ such that $\mathbf{Inf}_{i^*}[f] = 1$. Such i^* must exist by our assumption. Thus the mapping $x \mapsto x^{\oplus i^*}$ is a mapping from A to A^c that is one-to-one and onto. Therefore,

$$\begin{aligned}\mathbf{Inf}_i^{(\rho)}[f] &= \sum_{S\ni i:|S|\geq t} \rho^{|S|-1}\widehat{f}(S)^2\\ &\leq \rho^{t-1}\sum_{S\ni i}\widehat{f}(S)^2\\ &\leq \rho^{t-1}\,\mathbf{Inf}_i[f].\end{aligned}$$

The proof is complete. □

Proof of Lemma 6

Proof. By viewing the first k players $P_1, \ldots, P_k$ as models, the last player P_{k+1} as the adversary $\mathscr{A}$, and message strings in $\{0,1\}^\ell$ as the model output in the model composition, the proof is complete. □