Theoretical Limits of Provable Security Against Model Extraction by Efficient Observational Defenses[*]

Ari Karchmer[†]
Boston University
arika@bu.edu

January 3, 2023

Abstract

Can we hope to provide *provable* security against model extraction attacks? As a step towards a theoretical study of this question, we unify and abstract a wide range of "observational" model extraction defenses (OMEDs) — roughly, those that attempt to detect model extraction by analyzing the distribution over the adversary's queries. To accompany the abstract OMED, we define the notion of *complete* OMEDs — when benign clients can freely interact with the model — and *sound* OMEDs — when adversarial clients are caught and prevented from reverse engineering the model. Our formalism facilitates a simple argument for obtaining provable security against model extraction by complete and sound OMEDs, using (average-case) hardness assumptions for PAC-learning, in a way that abstracts current techniques in the prior literature.

The main result of this work establishes a partial computational incompleteness theorem for the OMED: any *efficient* OMED for a machine learning model computable by a polynomial size decision tree that satisfies a basic form of completeness cannot satisfy soundness, unless the subexponential Learning Parity with Noise (LPN) assumption does not hold. To prove the incompleteness theorem, we introduce a class of model extraction attacks called *natural Covert Learning attacks* based on a connection to the Covert Learning model of Canetti and Karchmer (TCC '21), and show that such attacks circumvent *any* defense within our abstract mechanism in a black-box, nonadaptive way. As a further technical contribution, we extend the Covert Learning algorithm of Canetti and Karchmer to work over any "concise" product distribution (albeit for juntas of a logarithmic number of variables rather than polynomial size decision trees), by showing that the technique of learning with a distributional inverter of Binnendyk et al. (ALT '22) remains viable in the Covert Learning setting.

1 Introduction

In a *model extraction attack*, an adversary maliciously probes an interface to a machine learning model in an attempt to extract the machine learning model itself. In many cases, preventing model extraction helps increase security and privacy, especially with respect to model inversion and adversarial example attacks (see e.g. [7] and references therein). Additionally, in Machine Learning as a Service (MLaaS), the model is considered confidential as the server usually operates with a pay-per-query scheme. Therefore, maintaining the secrecy of ML models by finding effective model extraction defense mechanisms is paramount. Indeed, the problem of how to defend against model extraction has been considered from a practical perspective previously (e.g. [8, 16, 18, 22, 26]).

[*] A version of this work is (to appear) in the IEEE Conference on Secure and Trustworthy Machine Learning (SATML), Feb '23. This is an updated version. A previous version of this paper used the title "Limits on Provable Security against Model Extraction Attacks."

[†] Supported by the DARPA SIEVE program, Agreement Nos. HR00112020020 and HR00112020021.

Most proposed model extraction defenses (MEDs) in the literature belong to two types (except a few notable exceptions, see e.g. [10]). The first type aims to limit the amount of information revealed by each client query. One intuitive proposal for this type of defense is to add independent noise (i.e. respond an incorrect prediction independently with some probability) or even deliberately modify the underlying model. This type of solution is not a focus of this work, because it necessarily sacrifices predictive accuracy of the ML model, and is therefore not an option for many ML systems where accuracy is critical such as autonomous driving, medical diagnosis, or malware detection.

The second type of MED that has been proposed aims to separate "benign" clients — those that want to obtain predictions but will not attempt to extract the model — and "adverse" clients — clients that aim to extract the model. This type of "observational" defense is the focus of the present work (see Figure 1). A common implementation of the observational defense involves so-called "monitors" that, receive as input a batch of queries submitted by the client, and compute some statistic meant to measure the likelihood of adversarial behavior, with the goal of rejecting a client's requests when the queries pass a certain threshold on the statistic (e.g. [16, 18, 22]). Essentially, observational defenses aim to control the *distribution* of the client's queries, by classifying any clients that fail to conform to the appropriate distributions as adverse, and then prohibiting them from accessing the model. To date, the choice of such appropriate distributions has been made heuristically, for instance, in [16], an appropriate distribution is one with the property that the distribution over hamming distances between independent queries is normally distributed.

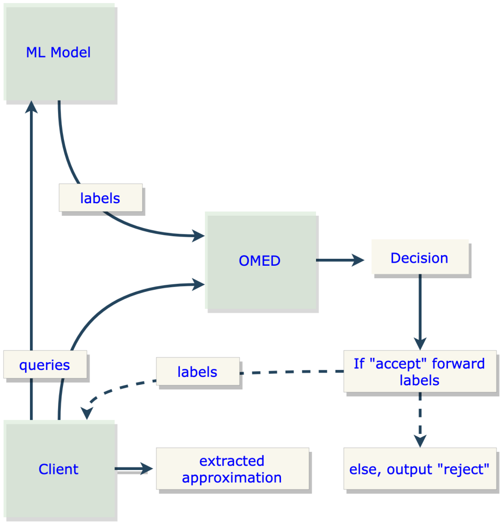

Figure 1: A depiction of the model extraction setting in the presence of an OMED. The adverse client queries the ML model, attempting to extract an approximation. The OMED watches over the interaction and outputs a decision to accept (and forward the labels) or reject the client based on whether or not it is deemed adverse or benign.

However, no formal definitions of security against model extraction have been suggested, and there has not been much formal work done in an effort to understand the theoretical underpinnings of the proposed observational defenses. This is highlighted by Vaikuntanathan as an open problem in [27]. As a result, a "cat-and-mouse" progression of attacks and defenses has developed, while no satisfying guarantees have been discovered (for neither cat nor mouse).

1.1 Towards Provable Security Against Model Extraction

In contrast, a lofty goal inspired by the theory of Cryptography would be to ultimately obtain *provable* security guarantees. For example, an initial attempt could try to leverage zero-knowledge style simulation-based security, to obtain the guarantee that a client learns nothing about the ML model that they could not have already learned prior to the interaction with the model. However, this is too strong of a goal, because at the very least the client will learn some queried examples.[1]

What kind of guarantees could we feasibly hope to obtain, then? One possible revised goal, could be to guarantee that a client learns only as much as they could learn from a set of *random queries* on the model. This privilege constitutes a middle ground between the too restrictive full zero-knowledge guarantee, and allowing a client total query access to the model.

We observe that this notion of security appears to be implicitly behind existing observational defenses. The literature on practical observational defenses tends to cite the goal of *detecting* model extraction, but the downstream effect is that the observational defenses seek to *exactly confine* the queries obtained by the client to some *specific distributions* (by enforcing a particular benign behavior). The benign behavior is enforced because the OMED will reject the client's queries if their distribution fails some chosen statistical test. Hence, the idea of only serving clients confined to these benign query distributions undoubtedly assumes that whatever can be deduced by a benign client about the model is indeed "secure."

To think about this deeper, let us focus on the case of observational defenses for binary classifiers. At first glance, the beautiful learning theory of Vapnik and Chervonenkis — which tells us that a number of samples proportional to the VC dimension of the hypothesis class suffices for PAC learning — seems to dash the hopes of using this model of security to obtain any meaningful protection. Indeed, an adversary with unbounded computational power could simply query the model according to one of the appropriate distributions of random examples for a sufficient number of times, and then apply a PAC-learning algorithm. The output of the algorithm would be a function which would be a strong approximation to the underlying ML model with high confidence.

However, this view does not account for the *complexity* of the implied model extraction attack: depending on the complexity of the model, this type of attack may have superpolynomial query and computational complexity. For instance, for many important families of classifiers (e.g. boolean decision trees), no efficient (i.e., polynomial time) PAC-learning algorithms are known despite intense effort from the learning theory community (though they exist given superpolynomial computational resources).[2] In fact, no efficient algorithms are known even when the queries are restricted to being uniformly distributed, and the classifier itself is drawn from some kinds of distributions (i.e., in an average-case way, see e.g. [5]).

Hence, this lends credence to the idea that the model of security implicitly considered by observational defenses might actually be effective in preventing unwanted model extraction by *computationally bounded adversaries*, by forcing the adversary to interact with the query interface in a way that mimics uniformly random examples, or some other hard example distribution. In this way, security against model extraction by computationally bounded adversaries could be *provable* in a complexity-theoretic way: one could hope to give a reduction from PAC-learning to model extraction in the presence of observational defenses. In other words, one could hope to prove a theorem that says "any efficient algorithm to learn an approximation of a proprietary ML model when constrained by an observational defense yields a distribution-specific PAC-learning algorithm (that is currently beyond all known techniques)."[3]

Yet, for classes of ML models that are efficiently PAC-learnable, the hope of any OMED is lost because of the VC theory argument described in the earlier in the section. *Thus, for the reasons outlined, this paper*

[1] For example, in the setting of Machine Learning as a Service (MLaaS), the client must be granted "in good faith" at least some ability to learn information, since otherwise the client may take business elsewhere.

[2] We note that there exist efficient learning algorithms for polynomial size decision trees that use *correlated queries* [13] [19]. Therefore, these families of classifiers are at least efficiently learnable by model owner who had this type of data access, so the setting is still nontrivial (i.e., the model is not completely unlearnable and therefore easy to defend against model extraction).

[3] One potential pitfall of the preceding discussion of provable security is that due to the worst-case guarantees for PAC-learning, the described reduction would not rule out the useless case that a single model is hard to extract in the presence of observational defenses, but all others are easy. However, even in an average-case or heuristic PAC-learning setting, where the concept itself is drawn from a distribution (see [5], [20]), there is still a conjectured cryptographic hardness of learning for sufficiently complex classes of concepts and concept distributions. Therefore, we can continue to envision a reduction from average-case learning to model extraction for *most* underlying ML models (provided they are sufficiently complex to begin with).

focuses on polynomial time adversaries who attempt to extract machine learning models with evidence of (polynomial time) hardness of learning. A concrete setting within focus that is handled in this paper is a polynomial time adversary trying to steal a polynomial size decision tree.

1.2 Our Contributions

Since we have established some faith behind the idea that observational defenses might be effective against computationally bounded adversaries (even in a provable way), a natural follow-up question to consider is if and when observational defenses can be *efficiently implemented* against those efficient adversaries. Thus, we seek an answer to the following two-part question:

> Can we provide *provable security* against model extraction attacks, for any ML model, using an observational defense? If so, can it be efficiently implemented?

In this work, we outline a framework for obtaining cryptographic-strength provable security via an observational defense. The framework is an abstraction of the prevailing heuristics used in the literature on practical observational defenses. Then, we provide a negative answer to the second part of the question. We do this via the following program:

- We formally define a class of abstract MEDs by unifying the common observational defense technique seen in the literature.

- We formalize the concepts of *complete* and *sound* observational defenses, namely, the provable guarantees that benign clients are accepted and may interact with the machine learning interface, and adverse clients are rejected. We show how our formalisms give a route to obtaining a (very) basic form of provable security against efficient model extraction attacks by relying on the computational hardness of PAC-learning. Throughout the paper, we argue that the proposed method is a useful abstraction of the methods that are (implicitly) employed by the literature on the construction of observational defenses.

- Via a connection to the Covert Learning model of [6], we give a method for generating provably good and efficient attacks on the abstract defense, granted that the defense is efficient and it satisfies a basic form of completeness. We then obtain an attack on decision tree models protected by the abstract class of MEDs by an existing algorithm of [6]. The attack relies on the subexponential Learning Parity with Noise (LPN) assumption, and to the best of our knowledge, constitutes the first provable and efficient attack on any large class of MEDs, for a large class of ML models.

- Using the existence of the attack, we prove our main result: informally, every efficient defense mechanism (within the abstract class of MEDs) for decision tree models which satisfies a basic notion of completeness does not satisfy soundness, even for efficient attackers. This result essentially prevents instantiating the described method for provable security, assuming the basic notion of completeness is satisfied.

- Finally, we extend the algorithm of [6] to work in the more general setting of learning with respect to "concise" product distributions, which gives a stronger impossibility result on the viability of efficient OMEDs. On the other hand, the new Covert Learning algorithm works for concepts computable by $O(\log n)$-juntas.[4]

[4] Though not as ubiquitous as larger decision trees, small junta functions still arise often in certain applications affected by model extraction. For example, consider a cloud-based ML model that accepts data corresponding to large strings of DNA, and labels them according to some genetic trait that is determined only by a small "active" part of the DNA string.

In the next two sections, we discuss related work, starting with a discussion on the nature of Covert Learning and the connection to model extraction, and an overview of the techniques used by [6] to obtain Covert Learning algorithms. We then highlight more related work including other approaches to security against model extraction and existing Covert Learning algorithms.

1.3 Covert Learning, and the Relationship to Model Extraction in the Presence of Observational Defenses

The Covert Learning model — a variant of Valiant's PAC model in the agnostic learning with membership queries setting — formalizes a new type of privacy in learning theory. Specifically, Covert Learning algorithms provide the guarantee that the membership queries leak very little information about the concept, with respect to a computationally bounded passive adversary. In other words, the learner can PAC-learn the concept in question (using knowledge of some internal randomness only known to itself),[5] while the adversary remains nearly completely "in the dark" with respect to the concept, even given a view of the entire transcript of membership queries and oracle responses. Crucially, the adversary is not privy to the secret randomness of the learning algorithm. At its heart, the Covert Learning model uses the foundational simulation paradigm of cryptography to achieve these goals.

Roughly, any membership query learning algorithm is a Covert Learning algorithm if it has an accompanying *simulator* which, when given access only to random examples (an "ideal" learner), emulates the distribution over the membership queries (the "real" learner) in a computational indistinguishable way. In other words, the simulator is able to produce a "believable" transcript of the interaction between the learner and the concept oracle, *without* query access to the underlying concept, but only (for instance) uniformly random examples. For a concept class where access only to uniformly random examples makes learning computationally hard, this proves that the learning transcript reveals very little information to a computationally bounded adversary. Hence, Covert Learning algorithms are reserved for concept classes that are not (known to be) efficiently learnable in the original PAC-model with respect to a certain "hard example distribution."

In this work, we focus on a special case of Covert Learning, which we term *natural* Covert Learning. In natural Covert Learning, rather than making necessary a simulator for the interaction, we strengthen the requirement by asking simply for the distribution over the membership queries to be computationally indistinguishable from some pre-defined "hard example distribution." For concreteness, let us consider the uniform distribution as the pre-defined "hard distribution." Also, we will consider a version that lives in the realizable learning setting (rather than agnostic).

For this concrete case, Figure 2 provides a graphical depiction. To explain a bit more formally, fix a concept $f : \{0,1\}^n \to \{-1,1\}$ contained in a concept class $\mathcal{C}$, and consider (for concreteness) the uniform distribution U_n over $\{0,1\}^n$. A learning algorithm under the Covert Learning model is tasked with finding an hypothesis $h : \{0,1\} \to \{-1,1\}$ that best approximates the concept D on unobserved examples $(x, f(x))$, for $x \sim U_n$. This notion is captured by a loss function, such as $L(h) = \Pr_{x\sim U}[h(x) \neq f(x)]$. A natural Covert Learning algorithm should then satisfy the PAC-learning guarantee: output h such that $\Pr_{x\sim U}[h(x) \neq f(x)] \leq \varepsilon$ with high probability (such an h is called ε-good). In order to achieve this goal, the learner is given access to a membership query oracle that labels a queried input $x \in \{0,1\}^n$ with a corresponding label $f(x)$. However, the important part of natural Covert Learning is that, essentially, the *joint* distribution over the membership queries made by the learner $[x_1, \cdots x_m]$ must be indistinguishable from $[x_1, \cdots x_m] \sim (U_n)^m$ by any computationally bounded adversary. The following is an informal definition:

Definition 1.1 (Natural Covert Learning — informal version of Definition 4.1). *A natural covert learning*

[5] Note that, the internal randomness is not even shared with the concept oracle. To use an analogy from Cryptography, Covert Learning is "public-key" in nature, as opposed to "symmetric-key," which might rely on shared randomness between the learner and the concept oracle.

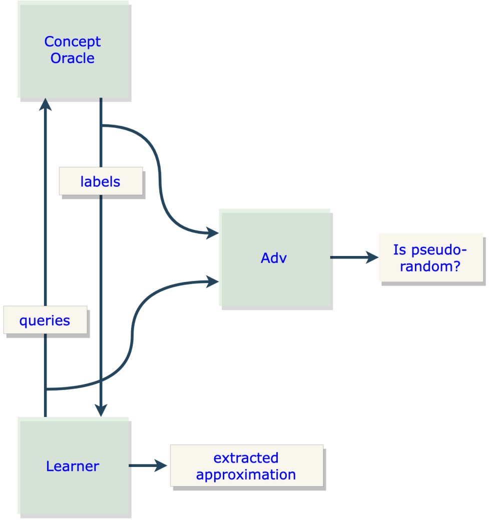

Figure 2: A depiction of natural Covert Learning. A learning algorithm queries an oracle for a concept at points of it's choice, with the goal of obtaining an approximation $\hat{f}$. Meanwhile, an eavesdropping adversary obtains a transcript of the interaction with the oracle and tries to distinguish the transcript from a set of random examples.

algorithm — for a class of concepts $\mathcal{C}$ and a distribution D over examples — is an algorithm that, for any $f \in \mathcal{C}$ and accuracy parameters ε, δ, interacts with an oracle that labels queries to the concept f such that the following are true:

- *Learning: The learning algorithm outputs an ε-good hypothesis for the concept with probability $1 - \delta$.*
- *Privacy: The joint distribution over all queries and responses to and from the oracle is computationally indistinguishable from the distribution $(x_1, f(x)), \cdots (x_m, f(x_m))$ for $x_1, \cdots x_m \sim (D_n)^m$.*

The conceptual crux of this work. There is a connection between the adversary in Covert Learning — a *distinguisher* that attempts to distinguish the membership queries made by the learner algorithm from random examples, — and the "adversary" in a model extraction attack — an OMED. At first glance this sounds confusing, because normally the adversary in model extraction is the malicious client that is trying to reverse engineer the model. However, consider that the job of the OMED is to somehow distinguish benign clients from adverse clients. Thus, thinking from the perspective of the actual adversary, i.e., the model extractor, the OMED is adversarial in nature and has essentially the same job as the natural Covert Learning adversary: to detect some property about the client's distribution of queries. This is even more apparent by considering the similarities between Figure 1 and Figure 2.

By exploiting this connection, we can show that natural Covert Learning algorithms form a recipe for an attack that can fool the OMED in the same way that they fool the Covert Learning adversary. Plus, the Learning guarantee of natural Covert Learning then allows us to still argue that the client can extract the underlying model. Hence, we can show that a single Covert Learning attack can achieve extraction while "fooling" *any* OMED. That is, any OMED will output "accept" with high probability even though an extraction attack is being performed. Since the attack is completely black-box with respect to the implementation of the OMED (it only requires that the OMED is *efficient* and satisfies the completeness condition for some basic distributions over queries), the existence of this attack can be used to demonstrate the *incompleteness* of the OMED.

On the need for Covert Learning instead of PAC-learning. It is important to understand exactly how Covert Learning algorithms add to this paper. In Section 1.1, we touched on the fact that *all* models are at risk of extraction by the very fact that they are exposed by some interface — to repeat, the VC argument outlined in Section 1.1 shows that, given enough random access to the model, and enough computational power, an adversary has a PAC-learning algorithm to obtain an approximation to the underlying model. Additionally, since the PAC-learning algorithm only uses random examples, then the adversary can choose to make queries according to a distribution that "passes" the OMED.

However, the VC argument in general relies on unbounded computational power (only very simple concept classes are known to be PAC-learnable in polynomial time from random examples, e.g. linear models). Hence, the natural question is whether there exists effective model extraction defenses against polynomial time adversaries, as is customary in Modern Cryptography. Again, this is because many important models (e.g. decision trees, low-depth circuits) are not known to be PAC-learnable in polynomial time. On the other hand, a class like decision trees is learnable in polynomial time when using carefully synthesized *membership queries*. However, classic membership query algorithms like the Kushilevitz-Mansour algorithm [19] (which suffices to learn decision trees in polynomial time) make queries that can be *tested* and identified as problematic by the OMED.

Therefore, Covert Learning algorithms *bridge the gap* between these two settings; the Covert Learning algorithms constitute attacks that are *benign-looking*, but are actually *adverse*. In other words, they are polynomial time membership query PAC-learning algorithms that synthesize queries in a special way that is *provably* hard to distinguish from the classic PAC-learning with random examples setting. This means that Covert Learning attacks are both undetectable by an (efficient) OMED, *and* run in polynomial time. The VC argument is undetectable by the OMED, but is not efficient, while the membership query attack is efficient, but *is* detectable by the OMED.

Overview of the natural Covert Learning algorithm for decision trees of [6] . Since we use the Covert Learning algorithm for decision trees from [6] as a black box, let us use the remainder of this section to provide a useful overview of the algorithm.

The covert learning algorithm for decision trees begins by using a "masked" Goldreich-Levin algorithm in order to obtain large Fourier coefficients. The Goldreich-Levin algorithm [13] is an algorithm that selects correlated queries in an ingenious way to decode a noisy parity function. The algorithm can also be used learn Fourier coefficients large (i.e., $1/\text{poly}(n)$ magnitude) of any function $f : \{0,1\}^n \to \{-1,1\}$, in time polynomial in n, and has been said to inspire the similar Kushilevitz-Mansour algorithm [19].

The "masked query" technique of [6] works as follows: for any given query $x \in \{0,1\}^n$, generate a pseudo-random string by taking n samples from a carefully crafted Learning Parity with Noise (LPN) distribution (see [24] for a useful introduction on LPN). Letting the n samples be concatenated into string $y \in \{0,1\}^n$, the masked query is taken as the bit-wise xor $m = x \oplus y$. It is easy to see that the LPN distribution is pseudo-random, then so is the masked query m.

Mathematically, the string y can be written as $y = As \oplus e$, where A is a random (but not uniformly random) $n \times n$ binary matrix, $e \in \{0,1\}^n$ is a noise vector sampled by taking an n-wise direct product of a biased Bernoulli random variable with small, constant mean, and $s \in \{0,1\}^n$ is secret sampled according to a "chopped tail" n-wise direct product Bernoulli distribution with minimum entropy $\Theta(\log^2(n)/n)$. This LPN distribution is due to [28], who show its pseudo-randomness assuming the more standard subexponential LPN assumption (defined formally as Definition 4.4).

Then, [6] show that, by augmenting an instance of the Goldreich-Levin algorithm for learning large Fourier coefficients by masking each query x_i with the mask $y_i = As_i \oplus e_i$, then the returned labels $f(x_i \oplus y_i)$ can be post-processed by re-incorporating (to the i^{th} query) the dependency on the secret s_i, and using the approximate linearity of the LPN distribution and large Fourier coefficients, to access a noisy version of f (denoted $\tilde{f}$) which crucially has the property that any large low-degree Fourier coefficient of f is also large for $\tilde{f}$. The technique fails at extracting large Fourier coefficients of any degree, and instead obtaining only those with degree $O(\log n)$. The reason for this is that the noise "overflows" for the higher degree coefficients, while it remains manageable for low-degree coefficients due to the low minimum entropy nature of the LPN secret. Unfortunately, attempting to increase the degree bound of $O(\log n)$ by further decreasing the minimum entropy of the secret does not work without breaking the pseudo-randomness of the masks.

Because each query is masked independently by a pseudo-random string, it is straightforward to see how this Covert Learning algorithm is also a natural Covert Learning algorithm with the uniform distribution as the pre-defined "hard distribution."

Once the set of large $O(\log\, n)$-degree Fourier coefficients is found, one can estimate the magnitudes (using random examples) of each Fourier coefficient, and them produce a hypothesis which is the sign of linear combination of parity functions. Under further analysis, the hypothesis obtains agnostic learning guarantees for the hypothesis class of poly(n) size decision trees.

1.4 More Related Work

Existing OMEDs. We point the reader to Section 3.2 for a detailed discussion on some of the practical OMEDs proposed in the literature.

Secure inference for MLaaS. A somewhat related approach to improving the privacy of Machine Learning as a Service (MLaaS) termed "secure inference" has been proposed. This approach borrows from ideas in the field of Secure Function Evaluation (where parties can securely compute a function without revealing their inputs), and makes use of garbled circuits [25] or fully homomorphic encryption [12]. However, the principle guarantee of the "secure inference" approach only provides hiding of information about the model beyond what can be deduced from the query and the model's output. Hence, a secure inference approach to security against model extraction would implicitly assume (incorrectly) that total leakage from the predictions is little, and that recovering the model from its predictions would be infeasible or impossible. Therefore, the "secure inference" approach does not properly prevent model extraction when considering clients who repeatedly interact with the service.

More natural Covert Learning attacks. The works of [1] and [11] introduce a method for sampling pairs of matrices (A, T) with entries in $\mathbb{Z}_q$, such that A is statistically close to a uniformly random matrix, while M_2 is a low-norm, full-rank trapdoor matrix such that $A \cdot T$ is the all zero matrix. In [27], Vaikuntanathan notes that this sampling algorithm gives an easy, yet somewhat contrived model extraction attack. In fact, it is a natural Covert Learning attack as well. More specifically, for any ML model that is essentially a linear function over $\mathbb{Z}_q$ with added Gaussian noise $e \in \mathbb{Z}_q^m$,[6] the linear function (denoted $s \in \mathbb{Z}_q^n$) can be extracted by querying $sA + e$, and then taking $(sA + e)T = eT$, which can then be used to extract e via Gaussian elimination (T is full rank). Then, given e, s is easily recoverable. However, A is *statistically* close to uniformly random, so the queries are impossible to distinguish from uniformly random queries with any significant advantage. This statistical Covert Learning attack could rule out even *unbounded* OMEDs, however it only works for the very narrow class of noisy linear models, which do not appear frequently in practice. The work of [6] discusses how to similarly sample trapdoors for the low-noise LPN problem of Alekhnovich [2].[7] The techniques gives rise to another natural Covert Learning attack for an LPN variant of the above setting, however the queries are only *computationally* close to uniform. For an elaboration of this technique, we refer the reader to [6].

Related formalisms. We note that the formalisms in this work are inspired by the field of Interactive Proofs [14]. Also, the work of [9], who work on protocols for verifying forecasting algorithms, inspired the drive to prove computational incompleteness theorems in this setting.

[6] This setting is a bit contrived, since the typical ML models would rarely resemble such a noisy inner product mod q.

[7] These techniques closely resembled that of the seminal work of Alekhnovich [2].

1.5 Organization

We define the abstract OMED and consider provable security in Section 3. In Section 4 we first show how to obtain a generic attack on the OMED using a generic natural Covert Learning, and then instantiate the generic implication using the algorithm for Covert Learning of decision trees. In Section 5, we extend the Covert Learning algorithm to product distributions over queries, and obtain an extension of the incompleteness result of OMEDs to this setting.

2 Technical Preliminaries

We consider juntas, decision trees and disjunctive normal form formulas (DNFs). A DNF is function represented by an OR of ANDs. For example, $f(x) = (x_1 \wedge x_7) \vee (x_5 \wedge \neg x_2 \wedge \neg x_1) \vee (\neg x_4 \wedge x_2)$ where $f : \{0,1\}^n \to \{0,1\}$. The class of DNFs also corresponds to depth two AC^0, that is, depth two alternating circuits with AND/OR/NOT gates. The size of the DNF is the number of ANDs (also the number of terms, or disjunctions), while the width of a DNF is the maximum number of variables over all terms. For the definition of binary decision tree, we refer the reader to chapter three of [21]. The size of a decision tree is identified with the number of leaves on the tree. A k-junta is a function that depends on at most k variables of the input. A boolean function $f : \{0,1\}^n \to \{0,1\}$ *depends* on the i^{th} variable if there exist inputs x, y that are the same everywhere except the i^{th} coordinate and satisfy $f(x) \neq f(y)$. The variable x_i is then called *relevant.*

2.1 Computational Indistinguishability

We will use the following standard notion of computational indistinguishability.

Definition 2.1 (Computational Indistinguishability). *Let $\{X_n\}, \{Y_n\}$ be sequences of distributions with X_n, Y_n ranging over $\{0,1\}^{m(n)}$ for some $m(n) = n^{O(1)}$. $\{X_n\}$ and $\{Y_n\}$ are computationally indistinguishable if for every polynomial time algorithm A and sufficiently large n,*

$$|\Pr[A(1^n, X_n) = 1] - \Pr[A(1^n, Y_n) = 1]| \leq \mathrm{negl}(n)$$

Often, n is clear from the context, so the subscript is omitted.

2.2 Learnability

We consider two notions of learnability of boolean concepts.

A boolean concept class $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$ is a sequence of sets where $\mathcal{C}_n$ is a set of boolean functions each taking $x \in \{0,1\}^n$ as input, and outputting a label $y \in \{-1,1\}$. Similarly, a distribution class $\mathcal{D} = \{\mathcal{D}_n\}_{n \in \mathbb{N}}$ is a sequence of sets of distributions, where each $\mathcal{D}_n$ is a set of example distributions over $\{0,1\}^n$.

We define the following learning models that are considered in this work. The oracle $\mathrm{Ex}(f, D)$ samples $x \sim D$ (for a distribution D over $\{0,1\}^n$) and returns $(x, f(x))$. We will write $\mathrm{Ex}(f, D, m)$ to denote m independent sampled from this oracle.

Definition 2.2 (Efficient PAC Learning). *A boolean concept class $\mathcal{C}$ is PAC-learnable with respect to a distribution class $\mathcal{D}$ if there exists an algorithm $\mathcal{A}$ such that for any $n \in \mathbb{N}$, distribution $D \in \mathcal{D}_n$, concept $f \in \mathcal{C}_n$, when $\mathcal{A}$ is given as input n and $\varepsilon, \delta > 0$, plus access to* $\mathrm{Ex}(f, D)$, *it outputs a function h such that*

$$\Pr_{\mathcal{A}}\left[\Pr_{x \sim D}\left[f(x) \neq h(x)\right] \leq \varepsilon\right] \geq 1 - \delta$$

We say that $\mathcal{C}$ is efficiently PAC-learnable with respect to $\mathcal{D}$ if $\mathcal{A}$ runs in time polynomial in $n, \varepsilon^{-1}, \delta^{-1}$, and the number of accesses to $\mathrm{Ex}(f, D)$ *is bounded by a polynomial in $n, \varepsilon^{-1}, \delta^{-1}$.*

The following definition of heuristic PAC learning due to Nanashima [20] can be seen as a variant of many existing average-case learning models, where the distribution over concepts is fixed to be uniform. In his original work, Nanashima defines a distribution over *representation* strings over concepts, but in this work it suffices to consider a distribution over actual concepts. The definition can also be interpreted as requiring PAC-learning for all but some fraction of concepts in the class, where this fraction is given as an input to the learner.

Definition 2.3 (Efficient Heuristic PAC-learning — adapted from [20]). *Let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a boolean concept class, and let $\mathcal{U}_n$ be the uniform distribution over $\mathcal{C}_n$. We say that $\mathcal{C}$ is heuristically PAC-learnable with respect to the distribution class $\mathcal{D}$ if there exists an algorithm $\mathcal{A}$ such that for any $n \in \mathbb{N}, D \in \mathcal{D}_n$, when $\mathcal{A}$ is given as input n and $\varepsilon, \delta, \eta > 0$, plus access to $\mathrm{Ex}(f, D)$ for some f sampeld uniformly at random from $\mathcal{C}_n$, it outputs a function h such that*

$$\Pr_f\left[\Pr_{\mathcal{A}}\left[\Pr_{z\sim\mathcal{D}}\Big[f(x) \neq h(x)\Big] \leq \varepsilon\right] \geq 1-\delta\right] \geq 1-\eta$$

We say that $\mathcal{C}$ is efficiently heuristically PAC-learnable with respect to $\mathcal{D}$ if $\mathcal{A}$ runs in time polynomial in $n, \eta^{-1}, \varepsilon^{-1}, \delta^{-1}$, and the number of accesses to $\mathrm{Ex}(f, D)$ is bounded by a polynomial in $n, \eta^{-1}, \varepsilon^{-1}, \delta^{-1}$. Additionally, we say that $\mathcal{C}$ is (efficiently) (ε', η')-heuristically PAC-learnable if we have fixed the parameters $\varepsilon = \varepsilon', \eta = \eta'$.

2.3 Basic Fourier Analysis

Let $f : \{0,1\}^n \to \mathbb{R}$. In particular, we consider functions $f : \{0,1\}^n \to \{-1,1\}$.

Definition 2.4. *For $S \subseteq [n]$, we define $\chi_S : \{0,1\}^n \to \mathbb{R}$ by*

$$\chi_S(x) = (-1)^{\sum\limits_{i\in S} x_i}$$

Fact 2.5. $\chi_S(x+y) = \chi_S(x)\chi_S(y)$

Definition 2.6. *The Fourier expansion of $f : \{0,1\}^n \to \mathbb{R}$ is*

$$f(x) = \sum_{S\subseteq[n]} \hat{f}(S)\chi_S(x)$$

where $\hat{f}(S)$ is the Fourier coefficient on S.

The Fourier coefficient $\hat{f}(S)$ is equal to the quantity $\mathbb{E}_x[f(x)\chi_S(x)]$, where x is sampled uniformly at random from $\{0,1\}^n$. For a comprehensive introduction to boolean Fourier analysis, see [21].

3 The Abstract Observational Model Extraction Defense

In this section, we formally introduce the Observational Model Extraction Defense (OMED) as a unifying abstraction for the current state of the art MEDs. The main purpose of this section is to establish a formal framework for considering a model extraction attack by an efficient client in the presence of an OMED, and to explain a simple way to argue for provable security in this framework. The argument for provable security is meant to reflect the ideas that appear implicitly in the literature for practical OMEDs (see e.g. Section 3.2). However, as we show later in the paper, it is not likely to be a route to security that can be backed by a provable theory. Furthermore, we do not claim that it is necessarily a very well-modelled way of obtaining security in the first place.

Before we introduce OMEDs formally, let us describe the setting of model extraction in the presence of an OMED in more detail.

At a very high level, the setting begins by fixing an ML model $f : \{0,1\}^n \to \{-1,1\}$ (potentially from some restricted class of functions). Then, we consider the case that an **efficient** client, can interact via an oracle to the ML model f. We denote this oracle by $\mathcal{O}_f$, and the client by $\mathbf{C}$. The goal of the benign client is to obtain some predictions $f(x)$ for queries $x \in \{0,1\}^n$. On the other hand, the goal of the adversarial client is to output a function $\hat{f}$, that approximately minimizes a loss function with respect to the model f. We will refine the behavior of a client and the way it interacts with the ML model after we define the OMED.

The important part is that the adverse client must be able to perform the extraction in the presence of the OMED. In particular, the OMED is able to view all the queries made by $\mathbf{C}$ and the labels that would be returned (see Figure 3). The OMED then (after performing arbitrary computations) outputs a decision "accept" or "reject." In the case of "accept," the OMED also forwards the labels back to the client. Otherwise, no labels are returned back to the client.

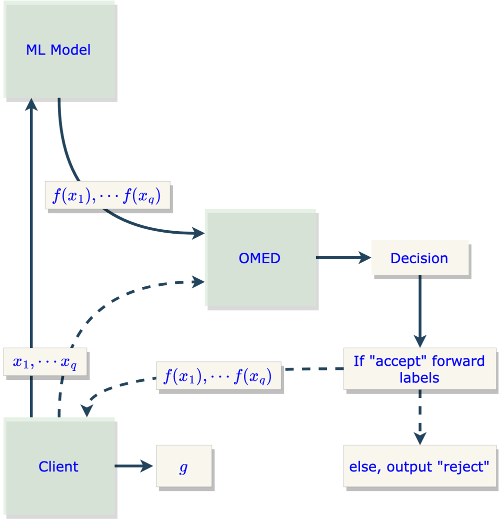

Figure 3: A depiction of the extraction setting. The adverse client queries the model f, attempting to extract an approximation g. The OMED watches over the interaction and outputs a decision to accept (and forward the labels) or reject the client based on whether or not it is deemed adverse or benign.

With this in mind, let us now formally define the OMED. As noted in the introduction and in [15], defense mechanisms for model extraction have mostly split into two tribes: reducing the information gained per client query, and differentiating malicious extraction adversaries from benign users. The OMED mechanism abstracts the latter approach; the implementation is left unspecified. Hence, we define an OMED abstractly as follows.

Let $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ be a class of ML models, where each $\mathcal{F}_n$ in the sequence is a set of functions taking as input an element of the set $\{0,1\}^n$ and giving output in the set $\{-1,1\}$.

Definition 3.1 (OMED). *A probabilistic algorithm $\mathcal{M}$ is a $T(n)$-OMED for a class of ML models $\mathcal{F}$ if for every $n \in \mathbb{N}$, $f \in \mathcal{F}_n$, $\mathcal{M}$ runs in time $T(n)$ and takes as input a list of examples $S = [(x_1, f(x_1)), \cdots (x_m f(x_m))] \subseteq (\{0,1\}^n \times \{-1,1\})^m$ and outputs a decision $\sigma \in \{\text{accept}, \text{reject}\}$.*

Sybil attacks, and other restrictions on the nature of the syntax of OMED. The definition of the OMED is defined as generally as possible from the perspective of the defense, but makes a restriction on the client: the queries are requested in large batches, rather than as an adaptive sequence. This nonadaptive setting (with respect to the query selection) can be viewed as unnecessarily restrictive on the client. However, since we prove negative results on the possibility of MEDs via OMEDs, the restriction on the client actually *strengthens* our results. We note that, as we will see in the next section, a benign client can arguably be considered one who essentially requests queries selected independently from the same distribution. In this case, adaptive query power inherently does not add strength to the client.

Furthermore, the defense could be expanded to a multi-client setting, where all clients must submit their batches simultaneously. This is again a restriction on the power of the client(s) (as opposed to multi-clients who do adaptively make batch requests), and thus strengthens our negative results.

Finally, we mention that with respect to sybil attacks, the underlying assumption for a sybil attack is that the MED will not evaluate the queries requested by each identity *jointly* (perhaps because the defense is assumed not to know the identities of the participating attackers). However, we prove our negative result on an OMED that always has access to *all* queries made by the attacker(s). Hence, our attack works in a more restrictive setting than sybil attacks. Alternatively, one can view our attack as a 1-sybil attack, or a sybil attack that works even when the defense knows the identity of every sybil participating in the attack.

3.1 A Potential Route to Provable Security via OMED

Definition 3.1 for the OMED makes no claims about desirable properties given by the OMED. Thus, what properties should we expect from the OMED? As mentioned in Section 1.1, the goal of the OMED is not just to classify the behavior of the clients, but to actually *confine* the clients to certain predefined benign behaviors. However, it is not enough to simply define security as the event that the client behavior is benign, because this actually needs to be detected and then enforced.

Therefore, it should be that a good OMED guarantees that (with high probability) any benign client is accepted, while any adverse client is rejected (and thus prevented from reverse engineering the underlying model). Naturally, the former requirement resembles completeness in an interactive proof system while the latter requirement resembles soundness. Through this lens of Interactive Proofs, we will formalize a notion of completeness and soundness.

First, let us explain how we model a client's behavior in the context of the model extraction setting depicted in Figure 3 in more detail. It has been noted in the model extraction literature (e.g. [16]) that defenses should consider how a client's queries relate to each other, rather than how they look individually. This idea is implemented by assuming that a client's queries follow a *distribution*[8]. We adopt a similar idea in this work: we assume that a client's queries follow a distribution over the domain of the ML model. To this end, we now identify a *client* $\mathbf{C}$ as a tuple (C, L) consisting of a polynomial time samplable *distribution* over sets of m queries C, as well as a probabilistic polynomial time algorithm L that takes as input a list of m queries drawn according to C and labelled by the ML model f, and outputs an approximate model $\hat{f}$. Thus, informally, the our idea is that a *benign* client request examples according to distributions that share some abstract property. We may formalize this as follows.

Let $\Delta(\{0,1\}^n \times \{-1,1\})$ be the convex polytope of all distributions over $\{0,1\}^n \times \{-1,1\}$.

Definition 3.2. *$\mathcal{P}_n$ is a property of distributions, where $\mathcal{P}_n \subseteq \Delta(\{0,1\}^n \times \{-1,1\})$.*

We then define a benign client as one which samples its queries in an independently and identically distributed fashion from some distribution satisfying the property:

Definition 3.3. *Fix any ML model f. We say that a client $\mathbf{C} = (C, L)$ is $\mathcal{P}_n$-benign if the distribution C is an m-wise direct product of samples drawn i.i.d. from a distribution D such that $D \in \mathcal{P}_n$.*

The i.i.d. restriction on the benign client may be considered harsh, but is rooted in the reality that often a real-world benign client requires labels from some "natural" set of examples. For instance, for an image recognition model, the benign client may simply forward requests for labels on images appearing in i.i.d. fashion according to some organically occurring distribution in Nature.

[8] This model for the basic behavior of a client has appeared previously in the literature (e.g. [16], [22]).

We will also use the terminology $\mathcal{P}_n$-adverse client to describe a client that is not $\mathcal{P}_n$-benign. Note that, a $\mathcal{P}_n$-adverse client need not sample queries i.i.d. from the same distribution; the set of m queries can be sampled using correlated randomness.

Completeness. Towards obtaining a theory of provable security against model extraction, we now argue that, intuitively, a good OMED (and any MED) should satisfy, informally: for any benign client, the defense mechanism does not reject and allows the client to continue to interact with the model, with high probability. A completeness requirement on a MED can be interpreted as formalizing the *usefulness* of the defense. In other words, the defense at the very least provides the opportunity for benign clients to interact with the model.

Formally, we define completeness as follows:

Definition 3.4 (MED completeness). *We say that an OMED $\mathcal{M}$ is δ-complete with respect to the property $\mathcal{P}_n$ if for any $\mathcal{P}_n$-benign client $\mathbf{C} = (C, L)$,*

$$\Pr_{\substack{\mathcal{M} \\ (x_1,\cdots,x_m)\sim C}} \Big[\mathcal{M}((x_1, f(x_1), \cdots, (x_m, f(x_m))) = \text{accept}\Big] \geq 1 - \delta$$

Provable security against benign clients: How should one choose the benign property? The definition of completeness implicitly assumes that, in the case that the client is classified as benign, they are essentially free to interact with the model. Thus, the underlying assumption is that by virtue of the client being benign, the ML model is not considered at risk for being extracted by the server. Therefore, the choice of the benign property is of utmost importance. As discussed in the introduction (see Section 1.1), this leaves room for a theory of provable security. For example, under our framework, a solid choice for a property would be the $\mathcal{P}_n = \mathcal{U}_n$ (i.e., the uniform distribution over examples). $\mathcal{P}_n = \mathcal{U}_n$ makes a solid choice because many interesting classes of models are thought to be hard to learn from uniformly random examples, even for most models in the class (that is, in the heuristic PAC-learning case, rather than just in the worst-case).

Using this idea, we can give a reduction from extracting the ML model (in the average-case over the uniform distribution) using $\mathcal{U}_n$-benign queries to heuristic PAC-learning with respect to the uniform distribution. To formalize this intuition, we prove the following lemma, which essentially says that if PAC learning is impossible for a large fraction of the class, then, given that the OMED accepted a client that used $\mathcal{U}_n$-benign queries, most models can not be extracted (with arbitrarily high fidelity) except with negligible probability. After giving the lemma we will provide more color by further discussing the reasons behind defining security this way. We will later use the lemma to show how to obtain a formal notion of security against model extraction by all clients, whether adverse or benign.

We first need to establish what constitutes a successful model extraction. Similarly to [8], we define the following extraction experiment. Again, let $\mathcal{F}_n$ be a class of boolean ML models, and let $\mathbf{C} = (C, L)$ be a client. Fix a loss function $\mathcal{L}_{D,f} : \mathcal{F}_n \to [0, 1]$ parameterized by an ML model $f \in \mathcal{F}_n$, and a distribution D over $\{0, 1\}^n$.

Definition 3.5 (Extraction experiment). *Let $\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}$ be defined as the output of the following process.*

1. *Sample $x_1 \cdots x_m \sim C$ for $\mathbf{C} = (C, L)$. These examples are paired with the labels to produce the set $S = [x_i, f(x_i)]_{i\in[m]}$.*

2. *Run $L(S)$ to obtain $\hat{f}$.*

3. *If $\mathcal{L}_{C,f}(\hat{f}) < \varepsilon$, output $(S, \text{extracted})$, else output $(S, \text{unextracted})$.*

Lemma 3.6 (Provable security against clients behaving benignly). *Let $\mathcal{M}$ be any OMED satisfying δ-completeness for any $\delta \leq 1 - 1/\text{poly}(n)$, with respect to a property of distributions $\mathcal{P}_n$. Then, if a class of ML models $\mathcal{F}_n$ is* ***not*** *efficiently (η, ε)-heuristic PAC-learnable with respect* ***to any*** *of the singleton distribution classes in the set $\{\{D\} : D \in \mathcal{P}_n\}$, then there exists $F \subseteq \mathcal{F}_n$ of size at least $\eta \cdot |\mathcal{F}_n|$, such that for any client $\mathbf{C} = (D, L)$ that is $\mathcal{P}_n$-benign, and for all $f \in F$,*

$$\Pr_{\substack{\mathcal{M} \\ (S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}} \left[\tau = \text{extracted} \;\middle|\; \mathcal{M}(S) = \text{accept}\right] \leq \text{negl}(n)$$

Proof. We show the contrapositive. Let E, A be the events (over the randomness of $\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}$) that $\tau = \text{extracted}$ and $\mathcal{M}(S) = \text{accept}$, respectively. Thus, suppose that for some $D \in \mathcal{P}_n$, we have a client $\mathbf{C} = (D, L)$ such that for $\varepsilon > 0$, and a δ-complete $\mathcal{M}$,

$$\Pr_{f\sim\mathcal{F}_n}\left[\Pr\left[E \mid A\right] \geq \frac{1}{\text{poly}(n)}\right] \geq 1 - \eta$$

This implies that

$$\Pr_{f\sim\mathcal{F}_n}\left[\ \frac{\Pr\left[E\ \wedge\ A\right]}{\Pr\left[A\right]} \geq \frac{1}{\text{poly}(n)}\ \right] \geq 1 - \eta$$

which is equivalent to

$$\Pr_{f\sim\mathcal{F}_n}\left[\ \Pr\left[E\ \wedge\ A\right] \geq \frac{1-\delta}{\text{poly}(n)}\ \right] \geq 1 - \eta$$

since $\mathcal{M}$ is δ-complete (the inner probabilities are all over the randomness of $\mathcal{M}, (S, \tau) \sim \mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}$; this is omitted due to space constraints). We now may observe that latest equation is the guarantee that there is an algorithm that (η, ε)-heuristically PAC learns $\mathcal{F}_n$ in time poly(n) with accuracy ε and confidence $(1 - \delta)/\text{poly}(n))$, for some singleton distribution class $\{D\}$ (where $D \in \mathcal{P}_n$). The learning algorithm works by sampling a set of labelled examples T according to D, and the applying $L(T)$ to obtain a hypothesis.

Then, it follows that $\mathcal{F}$ is (η, ε)-heuristically PAC-learnable with respect to some singleton distribution class $\{D\}$ (where $D \in \mathcal{P}_n$) by running the learning algorithm poly(n) times to produce many hypotheses, testing each by random sampling, and outputting the most accurate hypothesis. This works as long as $\delta \leq 1 - 1/\text{poly}(n)$. This completes the proof by contrapostive, which only requires we show learnability with respect to at least 1 singleton distribution class $\{D\}$ for $D \in \mathcal{P}_n$. □

We reiterate that the preceding lemma shows that, under an assumption that a class of ML models is hard to learn in the heuristic sense (i.e., most of models in the class are hard concepts) with respect to some property of example distributions, then clients that are benign on that property of distributions cannot extract most of the model in the class. This provides a theoretical foundation for instantiating the OMED paradigm, which assumes that some distributions are indeed benign, based off of hardness of heuristic PAC-learning assumptions. On the other hand, we remark that it is unfortunately not realistic to require hard extraction for *all* models, as many natural classes of models in the class (even for classes that are not even known to be efficiently PAC-learnable, e.g. decision trees), because natural classes still contain very simple models that can be learned super efficiently (e.g. a decision tree that always outputs the label 1). However, the model owner can make the assumption that their (fixed) model is inside the large set of hard-to-learn models over the benign distributions. Indeed, it can be argued that if the model owner is not willing to make this assumption, then the model is not suitable for the OMED paradigm in the first place, since complete OMEDs (with respect to benign property $\mathcal{P}_n \neq \emptyset$) allow the client to obtain some random examples by definition.

Soundness. The notion of provable security from Lemma 3.6 only deals with clients that request sets of examples that are $\mathcal{P}_n$-benign. Thus, we need to provide some guarantees when this is not the case. To this end, we also formalize soundness using the above extraction experiment. Informally, given that an adverse client has attempted an extraction, the OMED defense rejects the client, with high probability.

Definition 3.7 (MED soundness). *We say that an OMED $\mathcal{M}$ for $\mathcal{F}_n$ is δ-sound with respect to the property $\mathcal{P}_n$ if for any $\mathcal{P}_n$-adverse client $\mathbf{C}^* = (C^*, L)$, and for any $f \in \mathcal{F}_n, \varepsilon > 0$, it holds that*

$$\Pr_{\substack{\mathcal{M} \\ (S,\tau)\sim\mathsf{Exp}_{\mathbf{C}^*,f,m,\varepsilon}}} \Big[\mathcal{M}(S) = \text{accept} \;\Big|\; \tau = \text{extracted}\Big] < \delta$$

The soundness requirement can be interpreted as formalizing the *security* of the defense. That is, attackers will not be able to deceive the OMED into granting interaction with the ML model in such a way that allows extraction.

The definition also mirrors soundness from an Interactive Proof, where for any client that is *not* $\mathcal{P}_n$-benign, and given that the adversary would have extracted the model, the probability that the OMED $\mathcal{M}$ errs by accepting the queries is low. In a cryptographic setting, the OMED would set δ to a negligible function of n.

Cryptographically-Hard Model Extraction Against All Clients. To tie it all together, we now argue how combining completeness, soundness, and hardness assumptions for heuristic PAC-learning give a method for obtaining provable security via OMED. We stress that this argument is meant as a unifying abstraction to back up specific and practical implementations of OMEDs — not as a silver bullet for the model extraction problem. Our notion of security against model extraction (defined below) constitutes bounding the probability that two things are simultaneously true: (1) a client wins the extraction game, and (2) the OMED approved the interaction by outputting "accept." In theory, if at least one of these events is false, then the model will not be extracted (this is trivial if (1) is false, and if (2) is false, then the probability of extraction is 0 as the client never learns any labels). Thus, we aim to bound this probability by a function that is negligible in n (the size of the learning problem). We do so by actually requiring something a bit stronger:

Definition 3.8 (Security against Model Extraction by all clients). *We say that an OMED $\mathcal{M}$ for $\mathcal{F}$ is (η, ε)-secure against model extraction if for sufficiently large n, there exists $F \subseteq \mathcal{F}_n$ of size at least $\eta \cdot |\mathcal{F}_n|$, such that for any p.p.t. client $\mathbf{C} = (C, L)$ and for all $f \in F, D_n \in \mathcal{P}_n$,*

$$\Pr_{\substack{\mathcal{M} \\ (S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}} \big[\tau = \text{extracted} \;\big|\; \mathcal{M}(S) = \text{accept}\big] \\ \leq \text{negl}(n)$$

The point of this is that the strength of the hardness assumption on heuristic PAC-learning, i.e. the size of η, therefore directly translates to a better guarantee of hardness of extraction.

Theorem 3.9 (Provable Security from Complete and Sound OMEDs). *Let $\mathcal{M}$ be any p.p.t. OMED satisfying δ-completeness and γ-soundness for any $\delta \leq 1 - 1/\text{poly}(n), \gamma \leq \text{negl}(n)$, with respect to a property of distributions $\mathcal{P}_n$. Then, if a class of ML models $\mathcal{F}_n$ has no p.p.t. (η, ε)-heuristic PAC-learning algorithm with respect to any distribution $D_n \in \mathcal{P}_n$, then $\mathcal{M}$ is (η, ε)-secure against model extraction.*

Proof. By definition, any client $\mathbf{C} = (C, L)$ is either $\mathcal{P}_n$-benign or $\mathcal{P}_n$-adverse. In the former case, Lemma 3.6 (and the assumption of the hardness of learning $\mathcal{F}_n$) implies that

$$\Pr_{\substack{\mathcal{M} \\ (S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}} [\tau = \text{extracted} \mid \mathcal{M}(S) = \text{accept}] \\ \leq \text{negl}(n)$$

In the latter case, when $\mathbf{C}$ is $\mathcal{P}_n$-adverse, we have the guarantee from soundness of $\mathcal{M}$ (Definition 3.7) that

$$\Pr_{\substack{\mathcal{M} \\ (S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}}\Big[\mathcal{M}(S) = \text{accept} \;\Big|\; \tau = \text{extracted}\Big] < \gamma$$

where γ is a quantity bounded above by a negligible function of n. This suffices to complete the proof. □

3.2 Defense Proposals As Special Cases of the OMED

The pathway to provable security against model extraction by OMED from the previous section is purposely defined as generally as possible. However, we view it beneficial to discuss the relation to some concrete MEDs which have been proposed.

In this section, we will review three MEDs, [18], [16] and [22], demonstrating that each are special cases of an OMED (with unproved completeness and soundness guarantees). We start each example with a direct quote from the original paper so as to directly demonstrate the relevance to our OMED framework.

3.2.1 Extraction Monitors: Information Gain and Feature Space Coverage

The work of [18] proposes two different strategies for detecting model extraction attacks. Both strategies "[quantify] the extraction status of models by continually observing the API query and response streams of users" and provide a warning when a certain extraction status is reached. This is indeed the paradigm outlined by the OMED.

The first proposal of [18] seeks to continuously train a "proxy model" for each client, where the client queries are used to train the model. The function of the proxy model is to estimate the information/knowledge gained by a client with respect to a validation set which is given by the server (and when this information reaches some threshold the client is flagged). The distribution of this validation set mimics the training set of the underlying model. It is noted that it may require significant computational resources to train and update the proxy model (for every user and each incoming query), and thus [18] propose to use a lightweight decision tree proxy model.

In the second proposal, the observational keeps a short description of client queries, and estimates the client's learning rate (of the extraction attack) by analyzing the feature space covered by these queries (as they relate to to the class boundaries of the underlying model. It is noted that a drawback of this proposal is that the class boundaries of certain complex models (e.g. neural networks) are not easily found. Thus, it is proposed that the owner of the underlying model uploads a "surrogate" decision tree which has high fidelity with respect to the complex model (class boundaries of decision trees are easily interpreted by their leaf nodes).

3.2.2 PRADA

The MED known as PRADA [16] "analyzes the distribution of consecutive API queries and raises an alarm when this distribution deviates from benign behavior" [16]. Immediately, it is clear that the PRADA method is a candidate for being identified as a special case of the OMED. The defense works under the observation that queries requested by an adversarial client are likely to have a distribution that differs from the characteristic distribution of queries from a benign client. In PRADA, this benign characteristic was chosen to be the property that the distribution over hamming distances between each query in the requested batch should be normally distributed. This choice is backed by observational evidence that certain popular attacks such as the attack of [23] *do not* satisfy this condition. Hence, PRADA tries to satisfy completeness and soundness with respect to the property of all distributions that have a pairwise hamming distance normally distributed (e.g. the uniform distribution over $\{0,1\}^n$).

3.2.3 VarDetect

The work of [22] proposes a MED called VarDetect which is designed "to continuously observational the distribution of queries to [the model] from each user" [22]. Specifically, VarDetect trains a Variational Autoencoder (VAE) to map the "problem domain" (PD) dataset distribution (the PD distribution mimics the distribution of data that was used to train the underlying model) and the adversarial "outlier" (O) data distribution (the distribution of attacker queries) to distinct regions in latent space. Benign clients are assumed to query from the PD distribution while adverse clients are assumed to query from an O distribution. VarDetect purports to separate these two by computing the maximum mean discrepancy (MMD) between the latent mapping of the client's queries and that of the PD distribution (the MMD test flags the client if the result is above a certain threshold).

4 Attacks on Efficient OMEDs from Natural Covert Learning

In this section, we will consider the question:

> Can we efficiently realize the provable security guarantees outlined in the previous section?

Towards a negative answer, we will introduce an attack on the OMED technique for provable security, via a connection to Covert Learning [6]. Our attack will generate a distribution of examples which is *computationally indistinguishable* from a distribution in the property that is accepted by the OMED. In other words, the attacker operates (computationally) indistinguishably from a benign client, in the eyes of the OMED. Still, the attack classifies as adverse, and the labelled queries allow the attacker to extract the model with high fidelity.

4.1 Natural Covert Learning

We will focus on a special case of Covert Learning, which we call *natural Covert Learning.* As described in Section 1.3, a natural Covert Learning algorithm, essentially, is a membership query learning algorithm that satisfies the normal PAC-learning guarantees with respect to an example distribution D, with the added property that distribution over the membership queries and labels requested by the algorithms is computationally indistinguishable from examples sampled according to D.

More formally, let $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ be a boolean concept class, and let $D = \{D_n\}_{n\in\mathbb{N}}$ be a sequence of distributions where each D_n is a distribution over $\{0,1\}^n$.

Definition 4.1 (Natural Covert Learning). *We say that $\mathcal{A}$ is an natural Covert Learning algorithm for $\mathcal{C}$ with respect to D if for every $n \in \mathbb{N}, f \in \mathcal{C}_n, \varepsilon, \delta > 0$, $\mathcal{A}$ satisfies the following when given membership query access to f:*

- *Learning. For the random variable $h = \mathcal{A}^{\mathcal{O}_f}(n, \varepsilon, \delta)$ we have*

$$\Pr_h\left[\Pr_{x\sim D_n}[h(x) \neq f(x)] \leq \varepsilon\right] \geq 1 - \delta$$

- *Privacy. Let the query complexity of $\mathcal{A}$ be m. For every p.p.t. adversary* $\texttt{Adv}$*, and $S \sim \textsc{Ex}(f, D_n, m)$,*

$$\left|\Pr_{\texttt{Adv},S}\left[\texttt{Adv}(S) = 1\right] - \Pr_{\texttt{Adv},\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})}\left[\texttt{Adv}(\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})) = 1\right]\right| \leq \operatorname{negl}(n)$$

where $\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})$ denotes the distribution over the queries made by $\mathcal{A}$ and the responses by the oracle.

We say that $\mathcal{A}$ is efficient if it runs in time polynomial in $n, \varepsilon^{-1}, \delta^{-1}$, and the number of queries m is bounded by polynomial in $n, \varepsilon^{-1}, \delta^{-1}$.

We note how natural Covert Learning is defined here with respect to a sequence of example distributions, rather than a sequence of *sets* of example distributions. This keeps the definition simpler. One can consider for example natural covert learning algorithms for $\mathcal{C}$ with respect to the sequence of uniform distributions $\{\mathcal{U}_n\}_{n\in\mathbb{N}}$, each over $\{0,1\}^n$.

4.2 Natural Covert Learning Attack

In this section, our goal is to show that the existence of a natural Covert Learning algorithm for a particular concept class implies inadequacy of a polynomial time OMED for a class of models equal to the concept class. More specifically, we prove that satisfying soundness for an OMED is impossible, for any reasonable completeness parameter. This suffices to rule out obtaining provable security against model extraction by an OMED by instantiating Theorem 3.9.

Theorem 4.2. *Suppose that there exists an efficient natural Covert Learning algorithm $\mathcal{A}$ (with query complexity m) for a concept class $\mathcal{C}$, with respect to the sequence of distributions $D = \{D_n\}_{n\in\mathbb{N}}$. Then for a property $\mathcal{P}_n$ that contains* $\mathrm{Ex}(f, D_n)$*, there exists a $\mathcal{P}_n$-adverse client $\mathbf{C}$ such that for any $n \in \mathbb{N}$, any* $\mathrm{poly}(n)$*-OMED $\mathcal{M}$ for $\mathcal{C}$ that is δ-complete with respect to $\mathcal{P}_n$, and any $\varepsilon, \delta_{\mathcal{A}} > 0$, and $f \in \mathcal{C}_n$,*

$$\Pr_{\substack{\mathcal{M}\\(S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}}\Big[\mathcal{M}(S) = \text{accept} \;\wedge\; \tau = \text{extracted}\Big] \geq 1 - \delta_{\mathcal{A}} - \delta - \mathrm{negl}(n)$$

where $\mathrm{negl}(n)$ *denotes a negligible function of n and $\delta_{\mathcal{A}}$ is the failure probability of $\mathcal{A}$.*

Proof. Let $D_{\mathcal{A}}$ be the distribution over the set of m examples which is queried by $\mathcal{A}$. Define the adverse client $\mathbf{C}^* = (D_{\mathcal{A}}, \mathcal{A})$. Let $S \sim \mathrm{Ex}(f, D_n, m)$ and $S_{\mathcal{A}} \sim D_{\mathcal{A}}$.

By δ-completeness of $\mathcal{M}$,

$$\Pr_{\substack{\mathcal{M}\\S\sim\mathrm{Ex}(f,D_n,m)}}\Big[\mathcal{M}(S) = \text{accept}\Big] \geq 1 - \delta$$

Using the fact that by the privacy guarantee stemming from the Covert Learning assumption, we have that for every p.p.t. adversary $\mathtt{Adv}$,

$$\left|\Pr_{\mathtt{Adv}}\Big[\mathtt{Adv}(S) = 1\Big] - \Pr_{\mathtt{Adv}}\Big[\mathtt{Adv}(S_{\mathcal{A}}) = 1\Big]\right| \leq \mathrm{negl}(n)$$

Since $\mathcal{M}, f$ are polynomial time computable, we thus get that

$$\Pr_{\substack{\mathcal{M}\\(S_{\mathcal{A}},\tau)\sim\mathsf{Exp}_{\mathbf{C}^*,f,m,\varepsilon}}}\Big[\mathcal{M}(S_{\mathcal{A}}) = \text{accept}\Big] \geq 1 - \delta - \mathrm{negl}(n)$$

Now, because of the learning guarantee of $\mathcal{A}$, it is also true that

$$\Pr_{\substack{\mathcal{M}\\(S_{\mathcal{A}},\tau)\sim\mathsf{Exp}_{\mathbf{C}^*,f,m,\varepsilon}}}\Big[\tau = \text{extracted}\Big] \geq 1 - \delta_{\mathcal{A}}$$

if we just set the desired accuracy parameter of $\mathcal{A}$ to be ε. Above, $\delta_{\mathcal{A}}$ is the failure probability parameter of $\mathcal{A}$.

Therefore, even if $\mathcal{M}(S_{\mathcal{A}}) = \text{accept}$ and $\tau = \text{extracted}$ are correlated events, we have that

$$\begin{aligned}\Pr_{\substack{\mathcal{M}\\(S_{\mathcal{A}},\tau)\sim\mathsf{Exp}_{\mathbf{C}^*,f,m,\varepsilon}}}&\Big[\mathcal{M}(S_{\mathcal{A}}) = \text{accept} \;\wedge\; \tau = \text{extracted}\Big]\\ &\geq 1 - (1 - \delta_{\mathcal{A}}) - (1 - \delta - \mathrm{negl}(n))\\ &\geq 1 - \delta_{\mathcal{A}} - (\delta + \mathrm{negl}(n))\end{aligned}$$

□

We now obtain the generic incompleteness theorem.

Corollary 4.3. *Suppose that there exists an efficient natural Covert Learning algorithm $\mathcal{A}$ (with query complexity m) for a hypothesis class $\mathcal{C}$ with respect to the sequence of distributions $D = \{D_n\}_{n\in\mathbb{N}}$. Then if for any $n \in \mathbb{N}$, $\mathcal{M}$ is a* $\mathrm{poly}(n)$*-OMED for $\mathcal{C}_n$ and is δ-complete with respect to the property $\{\mathrm{Ex}(f, D_n)\}$, then $\mathcal{M}$ is not $(1 - 100\delta/99 - \mathrm{negl}(n))$-sound with respect to $\mathcal{P}_n$.*

Proof. By Theorem 4.2 there exists a $\{D_n\}$-adverse client $\mathbf{C} = (D_{\mathcal{A}}, \mathcal{A})$ such that for any δ-complete OMED $\mathcal{M}$ (with respect to $\{D_n\}$) for $\mathcal{C}_n$, it holds that for any $f \in \mathcal{C}_n, \varepsilon > 0$,

$$\Pr_{\mathcal{M},(S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}\Big[\mathcal{M}(S) = \text{accept} \ \wedge \ \tau = \text{extracted}\Big] \\ \geq 1 - \delta - \delta_{\mathbf{C}} - \mathrm{negl}(n)$$

where $\mathrm{negl}(n)$ is a negligible function of n and $\delta_{\mathbf{C}}$ is the probability $\mathbf{C}$ fails to extract. $\delta_{\mathbf{C}} = \delta_{\mathcal{A}}$ is derived directly from $\delta_{\mathcal{A}}$, the failure probability of the natural covert learning algorithm.

Then, we can deduce that

$$\Pr_{\mathcal{M},(S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}\Big[\mathcal{M}(S) = \text{accept} \mid \tau = \text{extracted}\Big] \\ \geq \frac{1 - \delta - \delta_{\mathbf{C}} - \mathrm{negl}(n)}{1 - \delta_{\mathbf{C}}}$$

Thus by taking an appropriately large n, $\mathcal{M}$ cannot be $(1 - 100\delta/99 - \mathrm{negl}(n))$-sound for a reasonable choice of $\delta_{\mathcal{A}}$, when creating the client $\mathbf{C} = (D_{\mathcal{A}}, \mathcal{A})$ based on $\mathcal{A}$. □

4.3 Concrete Attack and Incompleteness Theorem

In this section, we show that under the subexponential hardness assumption on the standard LPN problem, there exists an attack of the type outlined in the previous section. Let us first formally introduce our assumption. Below, let $\beta_\mu^{m(n)}$ denote the $m(n)$-wise direct product distribution of Bernoulli random variables each with mean $\mu \in [0, 1]$.

Definition 4.4 (Search LPN assumption). *For $\mu \in (0, 0.5), n \in \mathbb{N}$, the $(m(n), T(n)){-}\mathsf{SLPN}_{\mu,n}$ search assumption states that for every inverter $\mathbb{I}$ running in time $T(n)$,*

$$\Pr_{s,\boldsymbol{A},e}[\mathbb{I}(\boldsymbol{A}, \boldsymbol{A}s \oplus e) = s] \leq \frac{1}{T(n)}$$

where $s \xleftarrow{\$} \mathbb{Z}_2^n, \boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_2^{m(n)\times n}, e \xleftarrow{\$} \beta_\mu^{m(n)}$.

Thus, the assumption we adopt is the $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})}){-}\mathsf{SLPN}_{\mu,n}$ assumption. The following theorem is implicit in [6]. Let $\mathsf{DT}[\mathrm{poly}(n)]$ be the set of all decision trees of size $\mathrm{poly}(n)$.

Theorem 4.5 (Covert Learning of decision trees from [6]). *Given query access to a function $f : \{0,1\}^n \to \{-1, 1\}$, there exists an algorithm $\mathcal{A}$ running in time* $\mathrm{poly}(s, 1/\varepsilon, \log(1/\delta))$ *and making* $q(n) = \mathrm{poly}(n, 1/\varepsilon, \log(1/\delta))$ *query accesses such that, unless the $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})}){-}\mathsf{SLPN}_{\mu,n}$ assumption does not hold,*

1. *(Learning) $\mathcal{A}$ outputs $h : \{0,1\}^n \to \{-1, 1\}$ such that*

$$\Pr_{x\sim\mathcal{U}_n}\Big[h(x) \neq f(x)\Big] \leq \min_{g\in\mathsf{DT}[s]} \Pr_{x\sim\{0,1\}^n}\Big[g(x) \neq f(x)\Big] + \varepsilon$$

with probability $1 - \delta$.

2. *(Privacy) The distribution over examples requested by $\mathcal{A}$ is computationally indistinguishable, but statistically distinguishable, from* $\mathrm{Ex}(f, \mathcal{U}_n, q(n))$.

We may now proceed to combine Theorem 4.2 and Theorem 4.5 to obtain a concrete natural Covert Learning attack on models implemented by decision tree classifiers of polynomial size. Let $\mathcal{U} = \{\mathcal{U}_n\}_{n\in\mathbb{N}}$, and let $\mathcal{L}_{D,f}(h) = \Pr_{x\sim D}[h(x) \neq f(x)]$.

Theorem 4.6. *Under the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption, there exists, for any* $n \in \mathbb{N}$, *a* $\mathcal{U}_n$*-adverse client* $\mathbf{C}$ *(for some* $m = \mathrm{poly}(n)$*) such that for any* δ*-complete OMED* $\mathcal{M}$ *(with respect to* $\{\mathcal{U}_n\}$*) for* $\mathsf{DT}[\mathrm{poly}(n)]$, *it holds that for any* $f \in \mathsf{DT}[\mathrm{poly}(n)]$,

$$\begin{aligned}\Pr_{\substack{\mathcal{M}\\(S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}} &\Big[\mathcal{M}(S) = \text{accept} \ \wedge \ \tau = \text{extracted}\Big] \\ &\geq \frac{99}{100} - \delta - \mathrm{negl}(n)\end{aligned}$$

Proof. Observe that the algorithm described in Theorem 4.5 constitutes a natural Covert Learning algorithm for $\mathcal{C}$ with respect to $\mathcal{U}$, and loss function $\mathcal{L}_{D,f}(h) = \Pr_x[h(x) \neq f(x)]$. We can set the failure probability of the underlying natural covert learning algorithm for the client to 1/100. Thus, the statement follows directly from Theorem 4.5 and Theorem 4.2. □

Incompleteness Theorem. The previous theorem can be interpreted as the existence of an attack against any OMED for decision tree classifiers with δ-completeness with respect to the uniform property (up to SLPN assumptions). Hence, we get the following corollary.

Corollary 4.7. *Under the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption, if* $\mathcal{M}$ *is a* $\mathrm{poly}(n)$*-OMED for* $\mathsf{DT}[\mathrm{poly}(n)]$ *and* $\mathcal{U}$*, then if* $\mathcal{M}$ *is* δ*-complete it is not* $(1 - 100\delta/99 - \mathrm{negl}(n))$*-sound.*

Proof. By Theorem 4.6, there exists, for any $n \in \mathbb{N}$, a $\mathcal{U}_n$-adverse client $\mathbf{C}$ such that for any δ-complete OMED $\mathcal{M}$ (with respect to $\{\mathcal{U}_n\}$) for $\mathsf{DT}[\mathrm{poly}(n)]$, it holds that for any $f \in \mathsf{DT}[\mathrm{poly}(n)]$,

$$\Pr_{\substack{\mathcal{M}\\(S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}} \Big[\mathcal{M}(S) = \text{accept} \ \wedge \ \tau = \text{extracted}\Big] \geq \frac{99}{100} - \delta - \mathrm{negl}(n)$$

Then, we can deduce that

$$\begin{aligned}\Pr_{\substack{\mathcal{M}\\(S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}} \Big[\mathcal{M}(S) = \text{accept} \ \Big| \ \tau = \text{extracted}\Big] &= \frac{\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[E\Big]}{\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathsf{Exp}_{\mathcal{C},\mathcal{U}_n f,q(n),\varepsilon} = \text{extracted}\Big]} \\ &\geq \frac{\frac{99}{100} - \delta - \mathrm{negl}(n)}{\frac{99}{100}} \\ &\geq 1 - \frac{100\delta}{99} - \mathrm{negl}(n)\end{aligned}$$

which completes the proof. □

5 Extension to Product Distributions

In this section, we show how to extend the result of [6] (expressed as Theorem 4.5 in this work) to "concise" product distributions over examples, while the caveat is that learning is with respect to the concept class of $O(\log n)$-juntas. The resulting algorithm is also in the realizable setting rather than agnostic.[9]

More generally, we show how to construct a natural Covert Learning algorithm for a concept class $\mathcal{C}$ with respect to any ensemble of "concise" product distributions over $\{0,1\}^n$, given any natural Covert Learning algorithm for a concept class $\mathcal{C}'$ over the uniform distribution, provided that $\mathcal{C}'$ satisfies some efficient "closure" property when composed with a sampling machine for the product distribution. This general reduction then implies the special case by incorporating the algorithm of [6] that witnesses Theorem 4.5. Using this extended natural Covert Learning algorithm, we also get a new impossibility result on the viability of OMEDs that monitor product distributions.

Before stating and proving the results, we introduce the necessary tools and definitions. First, we restrict our attention to "concise" product distributions, which are those that sample $x \in \{0,1\}^n$ by sampling each bit $x_1, \cdots x_n$ independently from possibly different Bernoulli random variables, each with mean representable by a limited number of bits.

Definition 5.1 (k-concise product distributions). *A k-concise product distribution $\mu_{p,k,n}$ over $\{0,1\}^n$ is identified by a list $p = (p_1, \cdots, p_n)$ of n k-bit strings (interpreted as integers), where the distribution $\mu_{p,k,n}$ samples $x \in \{0,1\}^n$ by sampling x_i to be 1 with probability $p_i/2^k$ and 0 with probability $1 - p_i/2^k$. We denote by $\mu_{p,k} = \{\mu_{p,k,n}\}_{n\in\mathbb{N}}$ the ensemble of k-concise product distributions.*

We say that the distribution $\mu_{p,k,n}$ is concise if k is a constant.

Now, we prove the following theorem, that gives a reduction from natural Covert Learning with respect to any ensemble of concise product distributions to natural Covert Learning with respect to the uniform distribution. The reduction also assumes that the concept remains within the designated class, even when composed with the sampling algorithm for the product distribution. Note, each k-concise product distribution $\mu_{p,k,n}$ can be sampled by a (multi-output) DNF $\bar{\mu}_{p,k,n}$, which takes as input kn random bits.

Definition 5.2 (Closure under composition of sampling). *We say that a concept class $\mathcal{C}$ is closed under composition with $\mu_{p,k}$ if for any $f \in \mathcal{C}$, the function $g : \{0,1\}^{\mathrm{poly}(n)} \to \{-1,1\}$ defined by $g(x) = f(\bar{\mu}_{p,k,n}(x))$ remains contained in $\mathcal{C}$.*

Theorem 5.3 (Natural Covert Learning over concise product distributions). *Let $\mathcal{C}$ be a concept class that is closed under composition with any $\mu_{p,k}$, for constant k. If there exists a natural Covert Learning algorithm for the concept class $\mathcal{C}$ with respect to the uniform distribution ensemble $\mathcal{U}$, then there exists a natural Covert Learning algorithm for $\mathcal{C}$ with respect to any ensemble of concise product distribution $\mu_{p,k}$.*

The proof of this theorem can be sketched as follows. We will use an object called a *distributional inverter* for concise product distributions, which (roughly) is an algorithm that takes as input a sample from $x \sim \mu_{p,k,n}$, and outputs a uniformly random pre-image z of x, under the sampling function $\bar{\mu}_{p,k,n}$. In other words, the distributional inverter outputs a "fake" sample of coins z used to sample the instance $x \sim \mu_{p,k,n}$ (hence $\bar{\mu}_{p,k,n}(z) = x$). Using this object, we can obtain a new Covert Learning algorithm by transforming the learning problem from being over a product distribution to being over the uniform distribution, but with a new concept being the composition of $\bar{\mu}_{p,k,n}$ and the original concept. Since the concept is assumed to be closed under composition with $\mu_{p,k,n}$, then this new learning problem is still handled by the underlying Covert Learning algorithm.

The definition of a distributional inverter is given formally below:

Definition 5.4 (Distributional Inverters). *A function $f : \{0,1\}^* \to \{0,1\}^*$ is γ-distributionally invertible if there is an efficient probabilistic algorithm $\mathtt{I}$ such that the distributions $\langle x, f(x)\rangle$ and $\langle \mathtt{I}(f(x)), f(x)\rangle$ where*

[9] We note that $O(\log n)$-juntas are a concept class that is not known to be PAC-learnable in polynomial time (not even when fixing the uniform distribution, or any concise product distribution), so it is an appropriate setting for our work. On the other hand, learning with respect to *smoothed* product distributions [17] or *multiple* product distributions [3] has been considered solved in polynomial time.

$x \sim \mathcal{U}$ are statistically indistinguishable within γ for all but finitely many lengths n. That is,

$$\big\langle x, f(x) \big\rangle \equiv_\gamma \big\langle \mathtt{I}(f(x)), f(x) \big\rangle$$

To prove Theorem 5.3, we will use the distributional inverter for concise product distributions of [4]:

Lemma 5.5 (Distributional inverter for concise product distributions, from [4])*. Concise product distributions have distributional inverters. Moreover, these inverters are computable in* AC^0 *with auxiliary random bits.*

For an overview of how the inverter works, see [4]. This distributional inverter is denoted by the name `ProdInv`. We are now ready to prove Theorem 5.3.

Proof of Theorem 5.3. Let r be the number of bits required to sample $\mu_{p,k,n}$ with the circuit $\bar{\mu}_{p,k,n}$. Let $g_f : \{0,1\}^r \to \{0,1\}$ be the composed concept defined by $g_f(x) = f(\bar{\mu}_{p,k,n}(x))$. By assumption, $g_f \in \mathcal{C}$ for every $f \in \mathcal{C}$.

Now, since we know that there is a natural Covert Learning algorithm $\mathcal{A}$ for $\mathcal{C}$ with respect to the uniform distribution, then we can obtain a new algorithm $\mathcal{A}'$ for $\mathcal{C}$ with respect to $\mu_{p,k}$ as follows.

1. Input: ε, δ, n, an oracle to $f \in \mathcal{C}$.
2. Prepare a simulated oracle to g_f by using the circuit $\bar{\mu}_{p,k,n}$ to first generate "intermediate" representations $y = \bar{\mu}_{p,k,n}(x)$ of a query x, and then query the oracle to f for y.
3. Using the simulated oracle, run $\mathcal{A}$ (with parameters $\varepsilon/2, \delta, r$) to obtain a hypothesis $h : \{0,1\}^r \to \{0,1\}$ for g_f with respect to the uniform distribution.
4. Output: $h \circ$ `ProdInv`, with statistical closeness parameter $\gamma = \varepsilon/2$.

Let us prove that the above algorithm has the desired learning and privacy guarantees, starting with learning. First, observe that the simulated oracle from step 1 is equivalent to a membership query oracle for g_f. Thus, the hypothesis h obtained in step 2 satisfies

$$\Pr_{z \sim \mathcal{U}} \big[h(z) \neq g_f(z)\big] \leq \varepsilon/2 \tag{1}$$

with probability at least $1 - \delta$. This means that if we consider the quantity

$$\mathrm{err}_h = \Pr_{x \sim \mu_{p,k,n}} \big[h \circ \mathtt{ProdInv}(x) \neq f(x)\big]$$

then the distributional inversion property of `ProdInv` (see Definition 5.4) allows one to conclude that

$$\begin{aligned}
\mathrm{err}_h &= \Pr_{z \sim \mathcal{U}} \big[h \circ \mathtt{ProdInv}(\bar{\mu}_{p,k,n}(z)) \neq f(\bar{\mu}_{p,k,n}(z))\big] \\
&\equiv_{\varepsilon/2} \Pr_{z \sim \mathcal{U}} \big[h(z) \neq f(\bar{\mu}_{p,k,n}(z))\big] \\
&\equiv_{\varepsilon/2} \Pr_{z \sim \mathcal{U}} \big[h(z) \neq g_f(z)\big]
\end{aligned}$$

By (1) this final quantity is bounded by $\varepsilon/2$, and therefore we can see that

$$\mathrm{err}_h \leq \varepsilon/2 + \varepsilon/2 \leq \varepsilon$$

Finally, we need to prove that $\mathcal{A}'$ satisfies privacy. To see this, first consider that the distribution over queries that are requested by $\mathcal{A}'$ is determined by applying $\bar{\mu}_{p,k,n}$ to each query in the set of $q(n)$ queries sampled from the distribution over queries $\mathcal{D}_{\mathcal{A}}$ made by $\mathcal{A}$. By the privacy guarantee of $\mathcal{A}$,

$\mathcal{D}_{\mathcal{A}}$ is computationally indistinguishable from $\text{Ex}(f, \mathcal{U}, q(n))$. Therefore, one can conclude that $\mathcal{D}_{\mathcal{A}'}$ is computationally indistinguishable from $\text{Ex}(f, \mu_{p,k,n}, q(n))$. This follows from a simple reduction, which takes into account that $\mu_{p,k,n}$ is polynomial time samplable by computing $\bar{\mu}_{p,k,n}$. This suffices to prove the statement. □

We will now use Theorem 5.3 to obtain Covert Learning for $O(\log n)$-juntas over concise product distributions. We will use the fact that $O(\log n)$-juntas are a special case of polynomial size decision trees (since they are always computable in $O(\log n)$ depth).

Definition 5.6. *Let* $\mathsf{DNF}[s, w]$ *be the class of functions* $f : \{0,1\}^n \to \{0,1\}$ *that are computable by a DNF with at most* s *terms of maximum width* w. $\mathsf{DNF}[s]$ *denotes the class of functions with no limit on the maximum term width.*

Definition 5.7. *Let* $\mathsf{Jun}[r]$ *be the class of functions* $f : \{0,1\}^n \to \{0,1\}$ *that are computable by a* r*-junta, which is a function that depends on at most* r *out of* n *variables in the input.*

Theorem 5.8 (Natural Covert Learning $O(\log n)$-juntas over concise product distributions). *Given query access to a function* $f \in \mathsf{Jun}[O(\log n)]$, *there exists an algorithm* $\mathcal{A}$ *running in time* $\text{poly}(n, 1/\varepsilon, \log(1/\delta))$ *and making* $q(n) = \text{poly}(n, 1/\varepsilon, \log(1/\delta))$ *query accesses such that, unless the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption does not hold, then for any* $\mu_{p,k}$ *for* $k = O(1)$,

1. *(Learning)* $\mathcal{A}$ *outputs* $h : \{0,1\}^n \to \{0,1\}$ *such that*

$$\Pr_{x \sim \mu_{p,k,n}} \Big[h(x) \neq f(x) \Big] \leq \varepsilon$$

   with probability $1 - \delta$.

2. *(Privacy) The distribution over examples requested by* $\mathcal{A}$ *is computationally indistinguishable from* $\text{Ex}(f, \mu_{p,k,n}, q(n))$.

In order to prove the theorem, we recall that we can use the algorithm from Theorem 4.5 in place of the generically assumed Covert Learning algorithm in Theorem 5.3, and that the class $\mathsf{Jun}[O(\log n)]$ maintains the necessary closure under composition condition, when considering concise product distributions.

Proof of Theorem 5.8. To apply Theorem 5.3, we only need to argue that for any $f \in \mathsf{Jun}[O(\log n)]$ we have that $g_f \in \mathsf{Jun}[O(\log n)]$ where $g_f = f \circ \bar{\mu}_{p,k,n}$. After that, we can invoke Theorem 5.3 to complete the proof.

To see that for any $f \in \mathsf{Jun}[O(\log n)]$, $g_f \in \mathsf{Jun}[O(\log n)]$, recall that each k-concise product distribution $\mu_{p,k,n}$ can be sampled by a (multi-output) DNF $\bar{\mu}_{p,k,n}$. This DNF takes as input kn random bits, and samples all n bits $x_1, \cdots x_n$ in parallel by taking the OR of every possible k-bit string (encoded by a k-wise AND) larger than the binary representation p_i (for parallel execution $i \in [n]$). The resulting DNF is thus of size $\text{poly}(n)$ since k is a constant.

Now, if we consider $f \in \mathsf{Jun}[O(\log n)]$, we can write $g_f(z) = f(\bar{\mu}_{p,k,n}(z))$. Initially, this appears to be a junta on top of a multi-output DNF. However, it can be compressed to a DNF, which only reads $O(\log n)$ different variables (and possibly their negations) with only a polynomial blow-up in size — therefore proving that it is still contained in $\mathsf{Jun}[O(\log n)]$.

To see this, consider a DNF that computes f (canonically, this could just be a "brute-force" DNF acts as a lookup table for each possible $O(\log n)$-bit string that determines the setting of each of the relevant variables). Now, observe that every variable in this DNF can be replaced by a k-wise AND that encodes a string that maps to a 1 under $\bar{\mu}_{p,k,n}$ if the variable is not negated, and 0 otherwise, and then repeating for all possible choices of strings and taking the OR of all the instances. The resulting circuit is a DNF, and the number of terms increases by a factor of $(2^k)^{O(\log n)}$ (at most 2^k possible choices of ANDs, for each relevant variable). More importantly, the number of relevant variables is $k \cdot O(\log n)$ (k relevant variables to sample the value of a single relevant variable of f, of which there are $O(\log n)$). Since g_f is a function of kn input bits for constant k, this proves that $g_f \in \mathsf{Jun}[O(\log n)]$. In other words, the class of $O(\log n)$-juntas is closed under composition with concise product distributions. This completes the proof of the statement. □

5.1 Incompleteness Theorem for OMED on Product Distributions

The theorems obtained in this section so far can now be applied to obtain incompleteness theorems for OMEDs that protect ML models computable by $O(\log n)$-juntas, where the benign property is any concise product distribution.

Theorem 5.9. *Under the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption, if* $\mathcal{M}$ *is a* $\mathrm{poly}(n)$*-OMED for* $\mathsf{Jun}[O(\log n)]$ *with respect to concise product distribution* $\mu_{p,k}$*, then if* $\mathcal{M}$ *is* δ*-complete it is not* $(1 - 100\delta/99 - \mathrm{negl}(n))$*-sound.*

Proof. Define the client $\mathbf{C} = (D_{\mathcal{A}}, \mathcal{A})$, where $\mathcal{A}$ is the algorithm that witnesses Theorem 5.8, and $D_{\mathcal{A}}$ is the distribution over $q(n)$ queries requested by $\mathcal{A}$. Fix some failure probability for $\mathcal{A}$, denoted by $\delta_{\mathcal{A}}$.

By the privacy guarantee of $\mathcal{A}$, we have that for any δ-complete OMED $\mathcal{M}$ (with respect to $\{\mu_{p,k,n}\}$) for $\mathsf{Jun}[O(\log n)]$, it holds that for any $f \in \mathsf{Jun}[O(\log n)]$,

$$\begin{aligned} \Pr_{\substack{\mathcal{M} \\ (S_{\mathcal{A}},\tau)\sim\mathsf{Exp}_{\mathbf{C},f,q(n),\varepsilon}}} \Big[\mathcal{M}(S_{\mathcal{A}}) = \text{accept}\Big] &\geq \Pr_{\substack{\mathcal{M} \\ S\sim\mathrm{Ex}(f,\mu_{p,k,n},q(n))}} \Big[\mathcal{M}(S) = \text{accept}\Big] - \mathrm{negl} \\ &\geq 1 - \delta - \mathrm{negl}(n) \end{aligned}$$

At the same time, the learning guarantee of $\mathcal{A}$ gives that

$$\Pr_{\substack{\mathcal{M} \\ (S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,q(n),\varepsilon}}} \Big[\tau = \text{extracted}\Big] \geq 1 - \delta_{\mathcal{A}}$$

where $\delta_{\mathcal{A}}$ is the failure probability set as a parameter of $\mathcal{A}$, and the accuracy parameter ε given to $\mathcal{A}$ is the same as the subscript in Exp.

Thus, by a union bound, we can conclude that

$$\Pr_{\substack{\mathcal{M} \\ (S_{\mathcal{A}},\tau)\sim\mathsf{Exp}_{\mathbf{C},f,q(n),\varepsilon}}} \Big[\mathcal{M}(S_{\mathcal{A}}) = \text{accept} \ \wedge \ \tau = \text{extracted}\Big] \geq 1 - \delta_{\mathcal{A}} - \delta - \mathrm{negl}(n)$$

Then, we can deduce that

$$\begin{aligned} \Pr_{\substack{\mathcal{M} \\ (S,\tau)\sim\mathsf{Exp}_{\mathbf{C},f,m,\varepsilon}}} \Big[\mathcal{M}(S) = \text{accept} \mid \tau = \text{extracted}\Big] &\geq \frac{1 - \delta_{\mathcal{A}} - \delta - \mathrm{negl}(n)}{1 - \delta_{\mathcal{A}}} \\ &\geq 1 - 100\delta/99 - \mathrm{negl}(n) \end{aligned}$$

for appropriately chosen $\delta_{\mathcal{A}} = 1/100$. □

6 Conclusion

In this work, we have (from a theoretical perspective) explored the problem of defending against model extraction attacks using an observational defense. Our first contribution is the formalization of the theory of observational defenses, and in particular the abstraction of the methods implicitly used by the existing literature for how to choose the right benign client query distributions. Our main conceptual contribution connects existing natural Covert Learning algorithms to the Model Extraction problem to prove that, under cryptographic assumptions, the technique for obtaining provable security via OMEDs cannot work for some natural special cases of the technique (monitoring for uniform and product distributions over examples). Finally, the extension of Covert Learning to product distributions is our main technical contribution.

We suggest the following two directions for future work. First, it would be interesting to expand the natural Covert Learning attacks to more distributions than just product distributions. Doing so would strengthen our incompleteness results further by generalizing the requirements of completeness of the OMED with respect to product distributions. Second, it would be very interesting to expand natural Covert Learning attacks to other real world classes of ML models, such as neural networks, as many other model extraction attacks target neural networks. As it currently stands, our attacks only work for ML models computable by polynomial size decision trees or juntas. However, we observe that since the natural Covert Learning algorithm of [6] is actually *agnostic*, meaning that it finds a hypothesis (nearly) as good as the best decision tree, then if the ML model is a neural network that can at least be (weakly) approximated by a decision tree, then some level of extraction would still be possible (if not with arbitrary accuracy).

Acknowledgements

I would like to thank Ran Canetti for several illuminating conversations about this work. I would also like to thank anonymous referees for substantial comments that improved the presentation of this work.

References

[1] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, 1996.

[2] Michael Alekhnovich. More on average case vs approximation complexity. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 298–307. IEEE, 2003.

[3] Jan Arpe and Elchanan Mossel. Multiple random oracles are better than one. *arXiv preprint arXiv:0804.3817*, 2008.

[4] Eric Binnendyk, Marco Carmosino, Antonina Kolokolova, Ramyaa Ramyaa, and Manuel Sabin. Learning with distributional inverters. In *International Conference on Algorithmic Learning Theory*, pages 90–106. PMLR, 2022.

[5] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Annual International Cryptology Conference*, pages 278–291. Springer, 1993.

[6] Ran Canetti and Ari Karchmer. Covert learning: How to learn with an untrusted intermediary. In *Theory of Cryptography Conference*, pages 1–31. Springer, 2021.

[7] Nicholas Carlini, Matthew Jagielski, and Ilya Mironov. Cryptanalytic extraction of neural network models. *arXiv preprint arXiv:2003.04884*, 2020.

[8] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1309–1326, 2020.

[9] Kai-Min Chung, Edward Lui, and Rafael Pass. Can theories be tested? a cryptographic treatment of forecast testing. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 47–56, 2013.

[10] Adam Dziedzic, Muhammad Ahmad Kaleem, Yu Shen Lu, and Nicolas Papernot. Increasing the cost of model extraction with calibrated proof of work. *arXiv preprint arXiv:2201.09243*, 2022.

[11] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206, 2008.

[12] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210. PMLR, 2016.

[13] Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, 1989.

[14] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

[15] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1345–1362, 2020.

[16] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. Prada: protecting against dnn model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 512–527. IEEE, 2019.

[17] Adam Tauman Kalai, Alex Samorodnitsky, and Shang-Hua Teng. Learning and smoothed analysis. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 395–404. IEEE, 2009.

[18] Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, and Sameep Mehta. Model extraction warning in mlaas paradigm. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 371–380, 2018.

[19] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[20] Mikito Nanashima. A theory of heuristic learnability. In *Conference on Learning Theory*, pages 3483–3525. PMLR, 2021.

[21] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[22] Soham Pal, Yash Gupta, Aditya Kanade, and Shirish Shevade. Stateful detection of model extraction attacks. *arXiv preprint arXiv:2107.05166*, 2021.

[23] 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, 2017.

[24] Krzysztof Pietrzak. Cryptography from learning parity with noise. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 99–114. Springer, 2012.

[25] M Sadegh Riazi, Christian Weinert, Oleksandr Tkachenko, Ebrahim M Songhori, Thomas Schneider, and Farinaz Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 707–721, 2018.

[26] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016.

[27] Vinod Vaikuntanathan. Secure computation and ppml: Progress and challenges. `https://www.youtube.com/watch?v=y2iYEHLY2xE&ab_channel=TheIACR`, 2021.

[28] Yu Yu and Jiang Zhang. Cryptography with auxiliary input and trapdoor from constant-noise lpn. In *Annual International Cryptology Conference*, pages 214–243. Springer, 2016.