The Good, the Bad and the Ugly: Meta-Analysis of Watermarks, Transferable Attacks and Adversarial Defenses

Grzegorz Głuch
UC Berkeley
gluch@berkeley.edu

Berkant Turan
Zuse Institute Berlin and TU Berlin
turan@zib.de

Sai Ganesh Nagarajan
University of Southern Denmark
sgnagarajan@imada.sdu.dk

Sebastian Pokutta
Zuse Institute Berlin and TU Berlin
pokutta@zib.de

Abstract

We formalize and analyze the trade-off between backdoor-based watermarks and adversarial defenses, framing it as an interactive protocol between a verifier and a prover. While previous works have primarily focused on this trade-off, our analysis extends it by identifying transferable attacks as a third, counterintuitive, but necessary option. Our main result shows that for all learning tasks, at least one of the three exists: a *watermark*, an *adversarial defense*, or a *transferable attack*. By transferable attack, we refer to an efficient algorithm that generates queries indistinguishable from the data distribution and capable of fooling *all* efficient defenders. Using cryptographic techniques, specifically fully homomorphic encryption, we construct a transferable attack and prove its necessity in this trade-off. Finally, we show that tasks of bounded VC-dimension allow adversarial defenses against all attackers, while a subclass allows watermarks secure against fast adversaries.

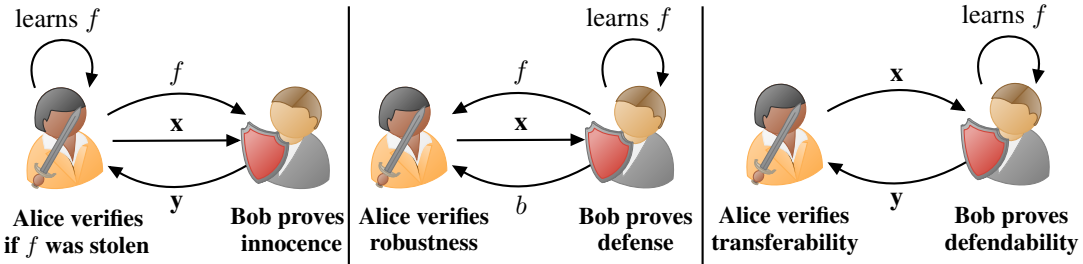
1 Introduction

Backdoor attacks and adversarial robustness are closely related: the former embeds hidden behaviors via subtle input changes, while the latter seeks to ensure stable predictions against worst-case input modifications. Recent works [Weng et al., 2020, Sun et al., 2020, Niu et al., 2024, Fowl et al., 2021, Tao et al., 2024] have empirically explored the trade-offs between adversarial robustness and backdoor attacks. Their general observation is that "models that are made to be robust against certain types of adversarial attacks may become more vulnerable to backdoor attacks."

Outside classic methods such as adversarial training Madry et al. [2018], which apply generally, provable defenses against general adversaries are known only for restricted function classes—particularly when the defense focuses on detecting, rather than classifying, attacks. Provided the attacks are not *indistinguishable* from the data distribution, rejection-based methods can effectively defend against arbitrarily crafted adversarial examples (beyond ℓ_p-norm perturbations) Goldwasser et al. [2020]. Other studies show that backdoors can be planted using cryptographic schemes Goldwasser et al. [2022], making their detection computationally intractable. Conversely, standard post-hoc defenses such as randomized smoothing may inadvertently remove such backdoors Cohen et al. [2019].

Formally analyzing these trade-offs must account for the full spectrum of strategies available to attackers and defenders—each with potentially different computational capacities and resources Bubeck et al. [2019], Garg et al. [2020]—which presents a significant challenge. Recently, Christiano et al. [2024] attempted to characterize function classes for which one can provably defend against

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a)

A **Watermark** is an efficient algorithm that computes a low-error classifier f and a set of queries $\mathbf{x}$ such that (fast) defenders are unable to find low-error answers $\mathbf{y}$ nor distinguish $\mathbf{x}$ from the data distribution.

(b)

An **Adversarial Defense** is an efficient algorithm that computes a low-error classifier f and a detection bit b, such that (fast) adversaries are unable to find queries $\mathbf{x}$, which look indistinguishable from the data distribution and where f is incorrect.

(c)

A **Transferable Attack** is an efficient algorithm that computes queries $\mathbf{x}$ that look indistinguishable from the data distribution, and that fool all efficient defenders.

Figure 1: Schematic overview of the interaction structure, along with short, informal versions of our definitions of (a) Watermark (Definition 3), (b) Adversarial Defense (Definition 4), and (c) Transferable Attack (Definition 5), with (c) tied to cryptography (see Section 6).

backdoor attacks: they demonstrate that classes with bounded VC-dimension admit defenses, and they construct classes for which designing a defense is computationally infeasible.

Finally, studying this trade-off is crucial because backdoor attacks can also serve as watermarks in black-box settings Adi et al. [2018], Zhang et al. [2018], Namba and Sakuma [2019]. Understanding their interplay informs us of the limitations and applicability of both defenses and backdoor-based watermarks across different learning tasks. This paper formalizes these notions and undertakes a meta-analysis of them. In doing so, it led us to identify a third scheme—the *transferable attack*, which is an attack that is *indistinguishable* from the data distribution and can fool all models trainable within given resource constraints.

1.1 Our Contributions

We study classifcation learning tasks and our main result shows that:

For every learning task, at least one of the three must exist:
A Watermark, an Adversarial Defense, or a Transferable Attack.

To prove this, we formalize and extend existing definitions of watermarks and adversarial defenses as an interactive protocol between two players—Alice and Bob, (see Figure 1) [Goldwasser and Sipser, 1986]. This protocol always has at least one winner—either Alice can embed an unremovable watermark, Bob can construct a strong adversarial defense, or a third option emerges: a transferable attack.

Transferable Attack. To understand transferable attacks, consider the following game. Alice interacts with a player who claims to have a secure model for a learning task $\mathcal{D}, h$, where $\mathcal{D}$ is the data distribution and h is the ground truth. Alice sends queries and observes the responses. She wins if she can generate queries that (i) cause significant errors and (ii) remain indistinguishable from samples drawn from $\mathcal{D}$.[1] Whether she succeeds depends on the computational and data resources available to her and the other player. If Alice can defeat *any* equally-resourced player, we call her queries a *Transferable Attack*. Intuitively, the more challenging a query becomes, the easier it should be to detect—but surprisingly, we show that transferable attacks do exist. Specifically, we prove:

[1]We note that what we consider a Transferable Attack is slightly nonstandard - there is no explicit model the attacks on which we consider transferability of. However, we can think that Alice first trains a model, then tries to find adversarial examples for it, and sends those as the queries in the game.

- The existence of a **Transferable Attacks** as defined above. Our construction uses cryptographic techniques, particularly Fully Homomorphic Encryption (FHE) [Gentry, 2009]. This establishes that Transferable Attacks form the third fundamental option in the trade-off.
- That any learning task supporting a Transferable Attack must be computationally complex. More precisely, Transferable Attacks imply the existence of a *cryptographic primitive*.

Notably, Tramèr et al. [2017], suggest a conjecture for the transferability of adversarial attacks: *If two models achieve low error for some task while also exhibiting low robustness to adversarial examples, adversarial examples crafted on one model transfer to the other.* However, they qualify their hypothesis by showing that it is not true in general, but only when the models are of the same class- thus complicating the picture. Our meta-analysis shows that whether the conjecture holds depends crucially on the **computational-resources** available to the attacker and defender. We argue that foregrounding computational resources in the problem of robustness clarifies the landscape considerably.

Constructions. We show that the existence of these properties does not depend on any particular algorithm or a model that is used. It depends on the learning task at hand and the computational resources for Alice and Bob. We give examples of learning tasks that provably support Watermarks, Adversarial Defenses and Transferable Attacks thereby justifying our framework. Concretely: (1) The construction of a learning task with a **Transferable Attack**, where the attacker needs strictly *fewer* resources than the defender. (2) We show that learning tasks with bounded VC dimension allow **Adversarial Defenses** against all (even computationally unbounded) attackers, ruling out Transferable Attacks in these settings. (3) We construct a **Watermark** for a class of learning tasks with bounded VC-dimension. Interestingly, in this case, both a Watermark and an Adversarial Defense coexist. Overall, these examples reiterate that the dependence on resources and the learning task are crucial.

Resource Allocation Implications. Our theorem can provide a rule of thumb for defenders. If an adversary has computational budget T (e.g., time), then allocating T^2 computation on the defender's side suffices (up to constant factors) to construct a defense whenever one exists under our assumptions. Conversely, if a T^2-budgeted procedure fails, this provides evidence that the instance admits *transferable attacks*, which in-turn precludes watermarks in our framework. To our knowledge, this is among the first quantitative attacker–defender resource trade-offs stated in a model-agnostic setting, i.e., beyond capacity-bounded regimes (e.g., finite VC dimension).

2 Related Work

While most trade-offs between backdoor-based attacks and adversarial defenses have been studied empirically, Pal and Vidal [2020] show (theoretically) that *Fast Gradient Methods* (attacks) and *Randomized Smoothing* (defenses) can form a Nash equilibrium under a restricted additive-noise model. They also provide experiments confirming this on datasets such as MNIST.[2] Our theoretical results generalize their findings to a broader class of attacks and defenses.

2.1 Adversarial Robustness

Adversarial robustness research includes techniques like adversarial training [Madry et al., 2018], which improves resilience via adversarial examples, and certified defenses [Raghunathan et al., 2018], which provide provable guarantees within perturbation bounds. Methods such as randomized smoothing [Cohen et al., 2019] extend these guarantees, but mainly as a defense against ℓ_p norm perturbations. Moving beyond this, the work of Goldwasser et al. [2020] establish provable and computationally efficient defenses against arbitrary adversarial examples by detection-based defense mechanisms, but on bounded VC-dimension classes as well.

2.2 Backdoor-Based Watermarks

In black-box settings, where model auditors lack access to internal parameters, watermarking methods often involve embedding backdoors during training. Techniques by Adi et al. [2018] and

[2] Pal and Vidal (2020) consider a game with a slightly different utility than ours.

Zhang et al. [2018] use crafted input patterns as triggers linked to specific outputs, enabling ownership verification by querying the model with these specific inputs. Advanced methods by Merrer et al. [2017] utilize adversarial examples, which are perturbed inputs that yield predefined outputs. Further enhancements by Namba and Sakuma [2019] focus on the robustness of watermarks, ensuring the watermark remains detectable despite model alterations or attacks. In the domain of Natural Language Processing (NLP), backdoor-based watermarks have been studied for Pre-trained Language Models (PLMs)[3], as exemplified by works such as [Gu et al., 2022, Peng et al., 2023] and [Li et al., 2023]. These approaches embed backdoors using rare or common word triggers, ensuring watermark robustness across downstream tasks and resistance to removal techniques like fine-tuning or pruning.

2.3 Undetectable Backdoors

A key related work by Goldwasser et al. [2022] shows how a learner can plant undetectable backdoors in any classifier. The authors propose two frameworks: one employing digital signature schemes [Goldwasser et al., 1985] to make backdoored models indistinguishable from the original to any computationally-bounded observer, and another using Random Fourier Features (RFF) [Rahimi and Recht, 2007], which remains undetectable even with full visibility of the model and training data.

In a very recent work, Christiano et al. [2024] introduce a defendability framework that formalizes the interaction between an attacker planting a backdoor and a defender tasked with detecting it. A major difference from our work, is that in their approach, the attacker chooses the distribution, whereas we keep the distribution fixed. This makes defendability in their model harder since the attacker has more control. However, in their framework, the backdoor trigger x^* is sampled from $\mathcal{D}$, so the attacker does not influence it. In contrast, our model allows the attacker to choose specific x's, making defendability in their model easier in this regard. Thus, the definitions are a priori incomparable. However, there are many interesting connections. They show that computationally unbounded defendability is equivalent to PAC learnability, while we, in a similar spirit, show an Adversarial Defense for all tasks with bounded VC-dimension. Using cryptographic tools, they show that the class of polynomial-size circuits is not efficiently defendable, while we use different cryptographic tools to give a Transferable Attack, which rules out a Defense.

3 Modeling

A key aspect of our formalization is modeling Alice and Bob while accounting for computational resources. We do so by representing them as families of circuits indexed by a size parameter n, a standard approach in complexity theory. Families of Boolean circuits—as used here—are Turing complete and can simulate any algorithm, making them a natural abstraction for studying learning tasks independent of implementation details. Although circuits are less common in computational learning theory than more loosely specified algorithms, this finer granularity is essential for our results.

3.1 Learning

Definition 1 (*Learning Task (Informal)*)**.** Let $\{0,1\}^n$ be an input space[4] A *learning task* $\mathbb{L}$ is defined as a sequence $\{\mathbb{L}_n\}_{n\in\mathbb{N}}$, where each $\mathbb{L}_n$ is a *fixed* distribution over pairs $(\mathcal{D}_n, h_n)$. Concretely, for each n, we draw $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$, where $\mathcal{D}_n$ is a distribution with domain $\{0,1\}^n$, and $h_n : \{0,1\}^n \to \{0,1\}$ is a *ground truth* labeling function.

To every model $f\colon \{0,1\}^n \to \{0,1\}$, we associate $\text{err}(f) := \mathbb{E}_{x\sim\mathcal{D}_n}[f(x) \neq h_n(x)]$. And for $q \in \mathbb{N}, \mathbf{x} \in (\{0,1\}^n)^q$, and predictions $\mathbf{y} \in \{0,1\}^q$, we define the empirical error to be: $\text{err}(\mathbf{x}, \mathbf{y}) := \frac{1}{q}\sum_{i\in[q]} \mathbb{1}_{\{h_n(x_i)\neq y_i\}}$.

Definition 2 (*Computationally Bounded Learnability (Informal)*)**.** Let $\epsilon, \delta : \mathbb{N} \to (0,1)$ be functions that specify the allowable error and confidence levels for each input size n, respectively. A learning

[3]We refer readers to Appendix J.3, where we discuss potential avenues for generalizing our framework to generative tasks. We explore the differences between generation and verification.

[4]We work over $\mathbb{F}_2$ (i.e., inputs in $\{0,1\}^n$) for analytic convenience. Any ML pipeline—processing images, tokens, or graphs—executes a finite sequence of arithmetic and logical operations that can be compiled into polynomial-size Boolean circuits.

task $\mathbb{L} = \{\mathbb{L}_n\}_{n\in\mathbb{N}}$ is said to be *learnable* to error $\epsilon(n)$ with confidence $1-\delta(n)$ and circuit complexity $S(n)$ if there exists a family of circuits $\{C_n\}_{n\in\mathbb{N}}$, where each circuit C_n has size at most $S(n)$, such that for every sufficiently large n, the following condition holds:

$$\mathbb{P}_{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n}\left[\text{err}_{\mathcal{D}_n,h_n}(f_n) \leq \epsilon(n)\right] \geq 1-\delta(n),$$

where $f_n : \{0,1\}^n \to \{0,1\}$ is the hypothesis computed by the circuit C_n when given sample access to $(\mathcal{D}_n, h_n)$, i.e., $f_n \leftarrow C_n$. In other words, with probability at least $1-\delta(n)$ over the choice of $(\mathcal{D}_n, h_n)$ drawn from $\mathbb{L}_n$, the circuit C_n successfully computes a function f_n that achieves an error rate of at most $\epsilon(n)$.

Definition 2 is very similar to the standard definition of efficient PAC learnability Kearns and Vazirani [1994]. The main difference is that instead of defining 'efficient' as polynomial in n (and $1/\epsilon, 1/\delta$) we define it as implementable by a circuit of size given by a fixed function $S(n)$. The reason for this increased generality is that we need finer control over sizes than, e.g., polynomial or exponential (see Theorem 1 where the separation between two circuit families is $S(n)$ *versus* $\sqrt{S(n)}$). A second difference is that compared to the standard definition we bound the size of circuits Arora and Barak [2009], not the running time. Assuming a processing unit without parallel execution the two notions can be thought equivalent. Formal definitions and additional details can be found in Appendix B. In the rest of the main part of the paper, we will often omit the parameter n when it is clear context.

Connections to Existing Models of Learning Definition 1 represents a learner's prior knowledge as a distribution over pairs $(\mathcal{D}_n, h_n)$, where $\mathcal{D}_n$ is a distribution on the domain $\{0,1\}^n$ and $h_n : \{0,1\}^n \to \{0,1\}$ is the ground truth. This models a learning task as a distribution over both input distributions and hypotheses, assuming a realizable scenario with a fixed ground truth.

Unlike distribution-specific or restricted family settings Kalai et al. [2008], Feldman et al. [2006], our definition does not limit the underlying support. While standard PAC learning requires generalization across all domain distributions, it often fails to explain the performance of complex models like DNNs, as their rich hypothesis classes make standard PAC bounds ineffective Zhang et al. [2021], Nagarajan and Kolter [2019]. Our definition aims to bridge this gap by providing a formal framework that aligns with contemporary practical learning scenarios.

3.2 Interaction

Alice and Bob will engage in interaction. To measure their computational resources, we require a specification of how the model f_n is transmitted between them. We assume that before the interaction starts they agree on a family of function classes $\mathcal{F} = \{\mathcal{F}_n\}_n$ as well as an encoding of them into messages of some length. This modeling implies that f_n are sent *white-box*. One example of such a family is the family of neural networks of a given architecture. See Appendix B for details.

3.3 Computational Indistinguishability

A crucial property of interest will be the indistinguishability of distributions. For a pair of distributions $\mathcal{D}^0, \mathcal{D}^1$ consider the following game between a sender and the distinguisher C: (1) The sender samples a bit $b \sim U(\{0,1\})$ and then draws a random sample $x \sim \mathcal{D}^b$, (2) C receives x and outputs $\hat{b} := C(x) \in \{0,1\}$. C wins if $\hat{b} = b$. We define the *advantage* of C for *distinguishing* $\mathcal{D}^0$ from $\mathcal{D}^1$

$$\mathbb{P}_{b\sim U(\{0,1\}),x\sim\mathcal{D}^b}[C(x) = b] = \frac{1}{2} + \gamma.$$

For a pair of families of distributions $\mathcal{D}^0 = \{\mathcal{D}^0_n\}_n, \mathcal{D}^1 = \{\mathcal{D}^1_n\}_n$, a function $\gamma : \mathbb{N} \to (0, \frac{1}{2})$, and a size bound $S : \mathbb{N} \to \mathbb{N}$ we say $\mathcal{D}^0, \mathcal{D}^1$ are γ-*indistinguishable* for circuits of size S if for every n, every circuit C (also known as the distinguisher) of size $S(n)$ the *advantage* of C for *distinguishing* $\mathcal{D}^0_n$ from $\mathcal{D}^1_n$ is at most $\gamma(n)$.

4 Watermarks, Adversarial Defenses and Transferable Attacks

In our protocols, Alice ($\mathbf{A}$, verifier) and Bob ($\mathbf{B}$, prover) engage in interactive communication, with distinct roles depending on the specific task. Each protocol is defined with respect to a learning

task $\mathbb{L}$, an error parameter $\varepsilon \in \left(0, \frac{1}{2}\right)$, and circuit size bounds $S_{\mathbf{A}}$ and $S_{\mathbf{B}}$, which are functions of n. A scheme is successful if the conditions of the protocols are satisfied. We denote the set of such circuits by $\text{SCHEME}(\mathbb{L}, \varepsilon, S_{\mathbf{A}}(n), S_{\mathbf{B}}(n))$, where SCHEME refers to WATERMARK, DEFENSE, or TRANSFATTACK (see Appendix C for the formal versions of all the definitions).

Definition 3 (*Watermark, informal*)**.**

A family of circuits $\{\mathbf{A}_n^{\text{WATERMARK}}\}_n$ of sizes $\{S_{\mathbf{A}}(n)\}_n$, implements a *backdoor-based watermarking scheme* for the learning task $\mathbb{L}$ with error parameter $\epsilon > 0$ if, for every sufficiently large n, an interactive protocol in which first $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$ and then $\mathbf{A}_n^{\text{WATERMARK}}$ computes a classifier $f\colon \{0,1\}^n \to \{0,1\}$ and a sequence of queries $\mathbf{x} \in (\{0,1\}^n)^q$, and a prover $\mathbf{B}_n$ outputs $\mathbf{y} = \mathbf{B}_n(f, \mathbf{x}) \in \{0,1\}^q$, satisfies the following properties:

1. **Correctness:** f has low error, i.e., $\text{err}(f) \leq \epsilon$.
2. **Uniqueness:** There exists a prover $\mathbf{B}_n$, of size $S_{\mathbf{A}}(n)$, which provides low-error answers, such that $\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon$.
3. **Unremovability:** For every prover $\mathbf{B}_n$ of size $S_{\mathbf{B}}(n)$, it holds that $\text{err}(\mathbf{x}, \mathbf{y}) > 2\epsilon$.
4. **Undetectability:** For every prover $\mathbf{B}_n$ of size $S_{\mathbf{B}}(n)$, the advantage of $\mathbf{B}_n$ in distinguishing the queries $\mathbf{x}$ generated by $\mathbf{A}_n^{\text{WATERMARK}}$ from random queries sampled from $\mathcal{D}_n^q$ is small.

Table 1: Backdoor–based Watermarks (Definition 3).

Property in Def. 3	**Classical analogue**	**Why it is needed?**
Correctness	Standard accuracy requirement (e.g., [Adi et al., 2018])	Ensures watermarking does not degrade task performance.
Uniqueness	Verifiability in black-box watermarking	Prevents false positives on independently-trained models.
Unremovability	Robustness to pruning / fine-tuning (e.g., [Namba and Sakuma, 2019])	Captures the usual "cannot be scrubbed" criterion.
Undetectability	Stealth requirement (e.g., [Merrer et al., 2017])	Guarantees watermark triggers look like in-distribution data.

As summarized in Table 1, uniqueness ensures that watermark verification cannot be triggered by independently trained models. Formally, we require that any $\mathbf{B}_n$ (Bob), who did not use f and trained a model f_{Scratch} using the specified procedure, must be accepted as distinct. This reflects realistic settings where multiple models could emerge independently.

Table 2: Adversarial Defenses (Definition 4).

Property in Def. 4	**Classical analogue**	**Why it is needed?**
Correctness	Baseline test-error requirement in certified / detection-based defences	Ensures the defended model remains useful.
Completeness	"No false positive" guarantee in detection frameworks ([Goldwasser et al., 2020])	Prevents trivial defences that reject everything.
Soundness	Detection + robustness guarantee (rejection-based defenses)	Formalises that attacks must both fool and stay indistinguishable.

Definition 4 (*Adversarial Defense, informal*)**.**

A family of circuits$\{\mathbf{B}_n^{\text{DEFENSE}}\}_n$ of sizes $\{S_{\mathbf{B}}(n)\}_n$, implements an *adversarial defense* for the learning task $\mathbb{L}$ with error parameter $\epsilon > 0$, if for every sufficiently large n, an interactive protocol in which first $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$ and then $\mathbf{B}_n^{\text{DEFENSE}}$ computes a classifier $f\colon \{0,1\}^n \to \{0,1\}$, while $\mathbf{A}_n$ replies with $\mathbf{x} = \mathbf{A}_n(f)$, where $\mathbf{x} \in (\{0,1\}^n)^q$, and $\mathbf{B}_n^{\text{DEFENSE}}$ outputs $b = \mathbf{B}_n^{\text{DEFENSE}}(f, \mathbf{x}) \in \{0,1\}$, satisfies the following properties:

1. **Correctness:** f has low error, i.e., $\text{err}(f) \leq \epsilon$.
2. **Completeness:** When $\mathbf{x} \sim \mathcal{D}_n^q$, then $b = 0$.

3. **Soundness:** For every $\mathbf{A}_n$ of size $S_{\mathbf{A}}(n)$, we have $\text{err}(\mathbf{x}, f(\mathbf{x})) \leq 7\epsilon$ or $b = 1$.

The key requirement for a successful defense is the ability to detect when it is being tested (see the soundness and completeness properties in Table 2). To bypass the defense, an $\mathbf{A}_n$ (Alice) must provide samples that are both adversarial, causing the classifier to err, and indistinguishable from samples of $\mathcal{D}_n$.

Definition 5 (*Transferable Attack, informal*)**.**

A family of circuits $\{\mathbf{A}_n^{\text{TRANSFATTACK}}\}_n$ of sizes $\{S_{\mathbf{A}}(n)\}_n$, implements a *transferable attack* for the learning task $\mathbb{L}$ with error parameter $\epsilon > 0$, if for every sufficiently large n, an interactive protocol in which first $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$ and then $\mathbf{A}_n^{\text{TRANSFATTACK}}$ computes $\mathbf{x} \in (\{0,1\}^n)^q$ and $\mathbf{B}_n$ outputs $\mathbf{y} = \mathbf{B}_n(\mathbf{x}) \in \{0,1\}^q$ satisfies the following properties:

1. **Correctness:** Size $S_{\mathbf{B}}(n)$ is sufficient to learn a classifier of low-error, $\text{err}(f) \leq \epsilon$.
2. **Transferability:** For every prover $\mathbf{B}_n$ of size $S_{\mathbf{A}}(n)$, we have $\text{err}(\mathbf{x}, \mathbf{y}) > 2\epsilon$.
3. **Undetectability:** For every prover $\mathbf{B}_n$ of size $S_{\mathbf{B}}(n)$, the advantage of $\mathbf{B}_n$ in distinguishing the queries $\mathbf{x}$ generated by $\mathbf{A}_n^{\text{TRANSFATTACK}}$ from random queries sampled from $\mathcal{D}_n^q$ is small.

Table 3: Transferable Attacks (Definition 5).

Property in Def. 5	**Classical analogue**	**Why it is needed?**
Correctness	Baseline learnability precondition	Ensures a meaningful low-error model exists for the attacker to exploit.
Transferability	Cross-model adversarial transfer ([Tramèr et al., 2017])	Captures worst-case attacks that succeed regardless of defender architecture or training.
Undetectability	Stealth / indistinguishability ([Goldwasser et al., 2020])	Guarantees defenders cannot filter the adversarial queries, aligning with cryptographic indistinguishability.

5 Main Result

We are ready to state an informal version of our main theorem (see Appendix D, for the full version). The key idea is to define a *zero-sum game* between $\mathbf{A}_n$ (Alice) and $\mathbf{B}_n$ (Bob), for every n, where the actions of each player are all possible circuits that can be realized with size $S_{\mathbf{A}}(n)$ and $S_{\mathbf{B}}(n)$. Notably, this game is finite, but there are exponentially many such actions for each player. We rely on some key properties of such large zero-sum games [Lipton and Young, 1994b] to argue about our main result.

Theorem 1 (*Main Theorem, informal*)**.** *For every $\epsilon \in \left(0, \frac{1}{2}\right), S : \mathbb{N} \to \mathbb{N}$ and learning task $\mathbb{L}$ learnable to error ϵ with high confidence with circuit complexity $S(n)$, at least one of these three exists*[5]*:*

$$\text{WATERMARK}\left(\mathbb{L}, \epsilon, S(n), o\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right)\right),$$

$$\text{DEFENSE}\left(\mathbb{L}, \epsilon, o\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right), O(S(n))\right),$$

$$\text{TRANSFATTACK}\Big(\mathbb{L}, \epsilon, S(n), S(n)\Big).$$

Proof (Sketch). The intuition of the proof relies on the complementary nature of Definitions 3 and 4. Specifically, every attempt to remove a fixed Watermark can be transformed to a potential Adversarial Defense, and vice versa. We define a zero-sum game $\mathcal{G}$ between circuits for watermarking $\mathbf{A}_n$ and circuits attempting to remove a watermark $\mathbf{B}_n$. The set of (pure) strategies of each player are all possible circuits that can be realized with size $S_{\mathbf{A}}(n)$ and $S_{\mathbf{B}}(n)$, and the payoff is determined by the probability that the errors and rejections meet specific requirements. It is well known that this

[5]We remark that formally the existence does not hold for all sufficiently large n but only with some 'frequency'. See Theorem 5 for a formal statement.

two-player zero-sum game admits a Nash equilibrium (NE) and the value of the game is unique v. Neumann [1928]. Let $\{\mathbf{A}_n^{\text{NASH}}\}_n$ and $\{\mathbf{B}_n^{\text{NASH}}\}_n$ be the NE strategies of Alice and Bob respectively. For each $n \in \mathbb{N}$, a careful analysis shows that depending on the value of the game, we have a Watermark, an Adversarial Defense, or a Transferable Attack. In the first case, where the expected payoff at the NE is greater than a threshold, we show there is an Adversarial Defense. As an illustration, consider some $n \in \mathbb{N}$, for which we define $\mathbf{B}_n^{\text{DEFENSE}}$ as follows. $\mathbf{B}_n^{\text{DEFENSE}}$ first learns a low-error classifier f, then sends f to the party that is attacking the Defense, then receives queries $\mathbf{x}$, and simulates $(\mathbf{y}, b) = \mathbf{B}_n^{\text{NASH}}(f, \mathbf{x})$. The bit $b = 1$ if $\mathbf{B}_n^{\text{NASH}}$ thinks it is attacked. Finally, $\mathbf{B}_n^{\text{DEFENSE}}$ replies with $b' = 1$ if $b = 1$, and if $b = 0$ it replies with $b' = 1$ if the fraction of queries on which $f(\mathbf{x})$ and $\mathbf{y}$ differ is high. Careful analysis shows $\mathbf{B}_n^{\text{DEFENSE}}$ is an Adversarial Defense. In the second case, where the expected payoff at the NE is below the threshold, we have either a Watermark or a Transferable Attack. The full proof can be found in Appendix D. □

6 Transferable Attacks and Cryptography

In this section, we show that tasks with Transferable Attacks exist. To construct such examples, we use cryptographic tools. But importantly, the fact that we use cryptography is not coincidental. As a second result of this section, we show that every learning task with a Transferable Attack *implies* a certain cryptographic primitive. One can interpret this as showing that Transferable Attacks exist only for *complex learning tasks*, in the sense of computational complexity theory.

6.1 A Cryptography-based Task with a Transferable Attack

Next, we give an example of a cryptography-based learning task with a Transferable Attack. The following is an informal statement of the formal version (Theorem 7) given in Appendix F.

Theorem 2 (*Transferable Attack for a Cryptography-based Learning Task, informal*)**.** *There exists a learning task $\mathbb{L}^{crypto}$ and $\mathbf{A}$ such that for all sufficiently small ϵ*

$$\mathbf{A} \in \textsc{TransfAttack}\left(\mathbb{L}^{crypto}, \epsilon, S_{\mathbf{A}} \approx \frac{1}{\epsilon}, S_{\mathbf{B}} = \Omega\left(\frac{1}{\epsilon^2}\right)\right).$$

Moreover, $\mathbb{L}^{crypto}$ is such that for every ϵ, $\approx \frac{1}{\epsilon}$ time (and $O\left(\frac{1}{\epsilon}\right)$ samples) is enough, and $\Omega\left(\frac{1}{\epsilon}\right)$ samples (and in particular time) is necessary to learn a classifier of error ϵ.

Notably, the parameters are set so that $\mathbf{A}$ (the party computing $\mathbf{x}$) has a *smaller* circuit size than $\mathbf{B}$ (the party computing $\mathbf{y}$), specifically $\approx 1/\epsilon$ compared to $\Omega(1/\epsilon^2)$. Furthermore, because of the cryptography tools used, this is a setting where a single input maps to multiple outputs, which deviates away from the setting of classification learning tasks considered in Theorem 1.

Proof (Sketch). We start with a definition of a learning task that will be later augmented with a cryptographic tool to produce $\mathbb{L}^{\text{crypto}}$.

Lines on Circle Learning Task $\mathbb{L}^{\circ}$ (Figure 2). We associate the input space $\{0,1\}^n$ with vertices of a 2^n regular polygon inscribed in $\{x \in \mathbb{R}^2 \mid \|x\|_2 = 1\}$. Let $\mathcal{H} := \{h_w \mid w \in \mathbb{R}^2, \|w\|_2 = 1\}$, where $h_w(x) := \text{sgn}(\langle w, x\rangle)$. Let $\mathbb{L}^{\circ}$ be a distribution corresponding to the following process: sample $h_w \sim U(\mathcal{H})$, return $(U(\{0,1\}^n), h_w)$. Additionally, let $B_w(\alpha) := \{x \in \{0,1\}^n \mid |\measuredangle(x, w)| \leq \alpha\}$ denote the set of points within an angular distance up to α to w.

Fully Homomorphic Encryption (FHE) (Appendix E). FHE [Gentry, 2009] allows for computation on encrypted data *without* decrypting it. An FHE scheme allows to encrypt x via an efficient procedure $e_x = \text{FHE.}\textsc{Enc}(x)$, so that later, for any algorithm C, it is possible to run C on x *homomorphically*. More concretely, it is possible to produce an encryption of the result of running C on x, i.e., $e_{C,x} := \text{FHE.}\textsc{Eval}(C, e_x)$. Finally, there is a procedure FHE.$\textsc{Dec}$ that, when given a *secret key* sk, can decrypt $e_{C,x}$, i.e., $y := \text{FHE.}\textsc{Dec}(\text{sk}, e_{C,x})$, where y is the result of running C on x. Crucially, encryptions of any two messages are indistinguishable for all efficient adversaries.

Cryptography-based Learning Task $\mathbb{L}^{\text{crypto}}$ (Figure 2). $\mathbb{L}^{\text{crypto}}$ is derived from *Lines on Circle Learning Task* $\mathbb{L}^{\circ}$. $\mathbb{L}^{\text{crypto}}$ corresponds to the following process: $w \sim U(\{w \in \mathbb{R}^2 \mid \|w\|_2 = 1\})$, return the distribution $\mathcal{D}^w$, which is an equal mixture of two parts $\mathcal{D}^w = \frac{1}{2}\mathcal{D}^w_{\text{CLEAR}} + \frac{1}{2}\mathcal{D}^w_{\text{ENC}}$. The first

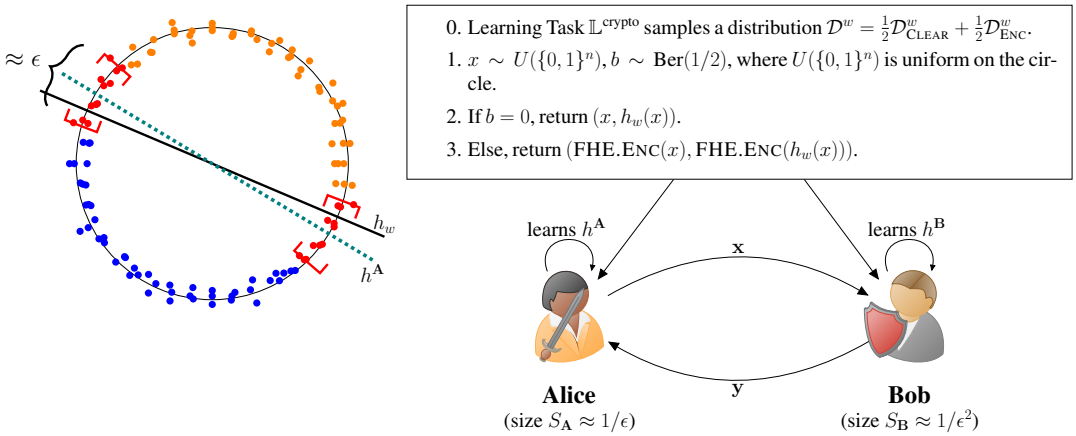


Figure 2: The left part of the figure represents a *Lines on Circle Learning Task* $\mathbb{L}^\circ$ with a ground truth function denoted by h_w. On the right, we define a *cryptography-augmented* learning task derived from $\mathbb{L}^\circ$. In its distribution, a "clear" or an "encrypted" sample is observed with equal probability. Given their respective times, both $\mathbf{A}$ and $\mathbf{B}$ are able to learn a low-error classifier $h^{\mathbf{A}}$, $h^{\mathbf{B}}$ respectively, by learning only on the *clear samples*. $\mathbf{A}$ is able to compute a Transferable Attack by computing an encryption of a point close to the decision boundary of her classifier $h^{\mathbf{A}}$.

part, i.e., $\mathcal{D}^w_{\text{CLEAR}}$, is equal to $x \sim U(\{0,1\}^n)$ with the correct label $y = h_w(x)$. The second part, i.e., $\mathcal{D}^w_{\text{ENC}}$, is equal to $x' \sim U(\{0,1\}^n), y' = h_w(x'), (x, y) = (\text{FHE.ENC}(x'), \text{FHE.ENC}(y'))$,[6] which can be thought of as $\mathcal{D}^w_{\text{CLEAR}}$ under an encryption. See Figure 2 for a visual representation. Note that we omitted the size parameter n for simplicity.

Transferable Attack (Figure 2). Consider the following attack strategy $\mathbf{A}$. First, $\mathbf{A}$ collects $O(1/\epsilon)$ samples from the distribution $\mathcal{D}^w_{\text{CLEAR}}$ and learns a classifier $h^{\mathbf{A}}_{w'} \in \mathcal{H}$ that is consistent with these samples. Since the VC-dimension of $\mathcal{H}$ is 2, the hypothesis $h^{\mathbf{A}}_{w'}$ has error at most ϵ with high probability.[7] Next, $\mathbf{A}$ samples a point x_{BND} uniformly at random from a region close to the decision boundary of $h^{\mathbf{A}}_{w'}$, i.e., $x_{\text{BND}} \sim U(B_{w'}(\epsilon))$. Finally, with equal probability, $\mathbf{A}$ sets as an attack $\mathbf{x}$ either $\text{FHE.ENC}(x_{\text{BND}})$ or a uniformly random point $\mathcal{D}^w_{\text{CLEAR}} = U(\{0,1\}^n)$. We claim[8] that $\mathbf{x}$ satisfies the properties of a Transferable Attack.

Since $h^{\mathbf{A}}_{w'}$ has a low error with high probability, x_{BND} is a uniformly random point from an arc containing the boundary of h_w (see Figure 2). The circuit size of $\mathbf{B}$ is upper-bounded by $\Omega(1/\epsilon^2)$, meaning it can only learn a classifier with error $\gtrapprox 10\epsilon^2$ (see Lemma 3 for details). $\mathbf{B}$'s can only learn (Lemma 3) a classifier of error, $\gtrapprox 10\epsilon^2$. Taking these two facts together, we expect $\mathbf{B}$ to misclassify x' with probability $\approx \frac{1}{2} \cdot \frac{10\epsilon^2}{\epsilon} = 5\epsilon > 2\epsilon$, where the factor $\frac{1}{2}$ takes into account that we send an encrypted sample only half of the time. This implies *transferability*.

Note that $\mathbf{x}$ is encrypted with the same probability as in the original distribution because we send $\text{FHE.ENC}(x_{\text{BND}})$ and a uniformly random $\mathbf{x} \sim \mathcal{D}^w_{\text{CLEAR}} = U(\{0,1\}^n)$ with probability $\frac{1}{2}$. Crucially, $\text{FHE.ENC}(x_{\text{BND}})$ is indistinguishable, for efficient adversaries, from $\text{FHE.ENC}(x)$ for any other $x \in \{0,1\}^n$. This follows from the security of the FHE. Consequently, *undetectability* holds. □

6.2 Tasks with Transferable Attacks Imply Cryptography

In this section, we show that a Transferable Attack for any task implies a *cryptographic primitive*.

EFID Pairs. In cryptography, an *EFID pair* [Goldreich, 1990] is a pair of ensembles of distributions $\mathcal{D}^0, \mathcal{D}^1$, that are **E**fficiently samplable, statistically **F**ar, and computationally **I**ndistinguishable. By

[6] Note that because FHE encryption is probabilistic there are many valid answers for a given x.

[7] $\mathbf{A}$ can also evaluate $h^{\mathbf{A}}_{w'}$ homomorphically (i.e., run FHE.EVAL) on $\text{FHE.ENC}(x)$ to obtain $\text{FHE.ENC}(y)$ of error ϵ on $\mathcal{D}^w_{\text{ENC}}$ also. This means that $\mathbf{A}$ is able to learn a low-error classifier on $\mathcal{D}^w$.

[8] In this proof sketch, we set $q = 1$, i.e., $\mathbf{A}$ sends only one x to $\mathbf{B}$. This is not true for the formal scheme.

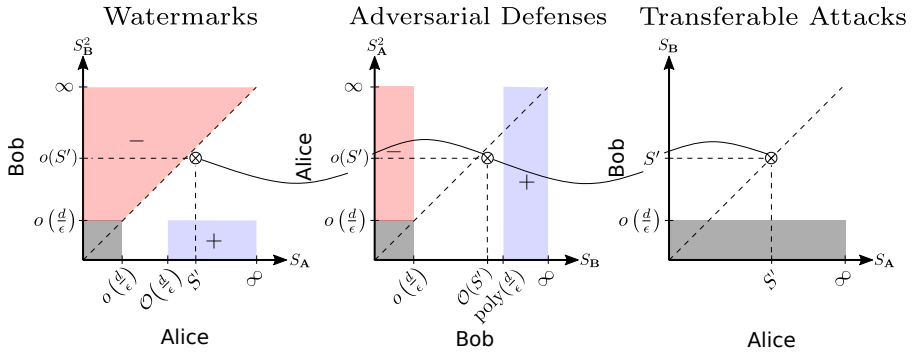


Figure 3: Overview of the taxonomy of learning tasks, illustrating the presence of Watermarks, Adversarial Defenses, and Transferable Attacks for learning tasks of bounded VC dimension. The axes represent the size bound for the parties in the corresponding schemes. The blue regions depict positive results, the red negative, and the gray regimes of parameters which are not of interest. See Lemma 5 and 6 for details about blue regions. The curved line represents a potential application of Theorem 1, which says that at least one of the three points should be blue.

a seminal result [Goldreich, 1990], we know that the existence of EFID pairs is equivalent to the existence of *Pseudorandom Generators* (PRG), which can be used for tasks including encryption and key generation [Goldreich, 1990], which makes EFID pairs a useful primitive. We consider a slight modification of the standard definition of EFID pairs, where instead of defining security to hold against polynomial time adversaries we do it for a fixed size bound function. More concretely, for two size bounds $S, S' : \mathbb{N} \to \mathbb{N}$ we call a pair of ensembles of distributions $(\mathcal{D}^0, \mathcal{D}^1)$ an (S, S')-EFID pair if for every n (i) $\mathcal{D}^0_n, \mathcal{D}^1_n$ are samplable by circuits of size $S(n)$, (ii) $\mathcal{D}^0_n, \mathcal{D}^1_n$ are statistically far, (iii) $\mathcal{D}^0_n, \mathcal{D}^1_n$ are indistinguishable for circuits of size $S'(n)$.

Tasks with Transferable Attacks imply EFID Pairs. The second result shows that any task with a Transferable Attack implies the existence of a type of EFID pair. This guarantees that any learning task with a Transferable Attack has to be computationally complex. The proof is in Appendix G.

Theorem 3 (*Transferable Attacks imply EFID pairs, informal*)**.** *For every $\epsilon \in (0,1), S_{\mathbf{A}}, S_{\mathbf{B}} : \mathbb{N} \to \mathbb{N}, S_{\mathbf{A}} \leq S_{\mathbf{B}}$, every learning task $\mathbb{L}$ learnable to error ϵ with high confidence and circuit complexity $S_{\mathbf{A}}$ if there exists* $\textsc{TransAttack}(\mathbb{L}, \epsilon, S_{\mathbf{A}}, S_{\mathbf{B}})$ *then there exists an $(S_{\mathbf{A}}, S_{\mathbf{B}})$-EFID pair.*

We note that it is unclear if the existence of EFID-pairs guaranteed by Theorem 3 implies PRGs because the sampling of $\mathcal{D}^0, \mathcal{D}^1$ requires oracle access to $\mathbb{L}$. Therefore, the standard construction of PRGs from EFID pairs does not automatically transfer.

7 Tasks with Watermarks and Adversarial Defenses

As the final pair of results, we present tasks exhibiting Watermarks and Adversarial Defenses. In the first, hypothesis classes with polynomially bounded VC-dimension admit polynomial-size Adversarial Defenses against all attackers. In the second, a learning task of polynomially bounded VC-dimension admits a Watermark secure against fast adversaries. These lemmas highlight the importance of bounding the sizes of $\mathbf{A}$ and $\mathbf{B}$. See Figure 3 for a visual summary; formal statements and proofs appear in Appendices H and I.

Lemma 1 (*Adversarial Defense for bounded VC-dimension, informal*)**.** *There exists $\mathbf{B}$ such that for every n, every learning task $\mathbb{L}$ of VC-dimension n[9], every sufficiently small ϵ,*

$$\mathbf{B} \in \textsc{Defense}\left(\mathbb{L}, \epsilon, S_{\mathbf{A}} = \infty, S_{\mathbf{B}} = \texttt{poly}\left(\tfrac{n}{\epsilon}\right)\right).$$

Lemma 2 (*Watermark for bounded VC-dimension against fast adversaries, informal*)**.** *For every d, there exists a learning task $\mathbb{L}$ of VC-dimension d and $\mathbf{A}$ such that for every sufficiently small ϵ,*

$$\mathbf{A} \in \textsc{Watermark}\big(\mathbb{L},\ \epsilon,\ q = O\big(\tfrac{1}{\epsilon}\big)\,,\ S_{\mathbf{A}} = O\big(\tfrac{d}{\epsilon}\big)\,,\ S_{\mathbf{B}} = \tfrac{d}{100}\big).$$

[9] It means that the ground truth sampled from $\mathbb{L}$ belongs to a class of VC-dimension n.

Acknowledgement

This research was partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689). We thank the anonymous reviewers for their thoughtful comments and suggestions, which improved the paper.

References

Lukáš Adam, Rostislav Horčík, Tomáš Kasl, and Tomáš Kroupa. Double oracle algorithm for computing equilibria in continuous games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5070–5077, 2021.

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1615–1631, 2018.

Noga Amit, Shafi Goldwasser, Orr Paradise, and Guy Rothblum. Models that prove their own correctness. *arXiv preprint arXiv:2405.15722*, 2024.

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks, 2024.

Cem Anil, Guodong Zhang, Yuhuai Wu, and Roger Grosse. Learning to give checkable answers with prover-verifier games. *arXiv preprint arXiv:2108.12099*, 2021.

Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009. ISBN 0521424267.

Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 309–325, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311520. doi: 10.1145/2090236.2090262. URL https://doi.org/10.1145/2090236.2090262.

Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. Scalable ai safety via doubly-efficient debate. *arXiv preprint arXiv:2311.14125*, 2023.

Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pages 831–840. PMLR, 2019.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 4971–5012. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/burns24b.html.

Nicholas Carlini, Milad Nasr, Christopher A. Choquette-Choo, Matthew Jagielski, Irena Gao, Anas Awadalla, Pang Wei Koh, Daphne Ippolito, Katherine Lee, Florian Tramèr, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *ArXiv*, abs/2306.15447, 2023. URL https://api.semanticscholar.org/CorpusID:259262181.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries, 2023.

Jiefeng Chen, Yang Guo, Xi Wu, Tianqi Li, Qicheng Lao, Yingyu Liang, and Somesh Jha. Towards adversarial robustness via transductive learning. *arXiv preprint arXiv:2106.08387*, 2021.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*, 2023.

Paul Christiano, Jacob Hilton, Victor Lecomte, and Mark Xu. Backdoor defense, learnability and obfuscation. *arXiv preprint arXiv:2409.03077*, 2024.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cohen19c.html.

Anne Condon, Joan Feigenbaum, Carsten Lund, and Peter Shor. Probabilistically checkable debate systems and approximation algorithms for pspace-hard functions. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of Computing*, pages 305–314, 1993.

Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: An end-to-end watermarking framework for ownership protection of deep neural networks. In *Proceedings of the twenty-fourth international conference on architectural support for programming languages and operating systems*, pages 485–497, 2019.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4307–4316, 2019. URL https://api.semanticscholar.org/CorpusID:102350868.

Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *ArXiv*, abs/1712.02779, 2017. URL https://api.semanticscholar.org/CorpusID:21929206.

Yousof Erfani, Ramin Pichevar, and Jean Rouat. Audio watermarking using spikegram and a two-dictionary approach. *IEEE Transactions on Information Forensics and Security*, 12(4):840–852, 2017. doi: 10.1109/TIFS.2016.2636094.

Jon Feldman, Rocco A Servedio, and Ryan O'Donnell. Pac learning axis-aligned mixtures of gaussians with no separation assumption. In *Learning Theory: 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006. Proceedings 19*, pages 20–34. Springer, 2006.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22466–22477, 2023.

Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. *Advances in Neural Information Processing Systems*, 34:30339–30351, 2021.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Sanjam Garg, Somesh Jha, Saeed Mahloujifar, and Mahmoody Mohammad. Adversarially robust learning could leverage computational hardness. In *Algorithmic Learning Theory*, pages 364–385. PMLR, 2020.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585062. doi: 10.1145/1536414.1536440. URL https://doi.org/10.1145/1536414.1536440.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian J. Goodfellow. Adversarial spheres. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, 2018. URL https://openreview.net/forum?id=SkthlLkPf.

Greg Gluch and Shafi Goldwasser. A cryptographic perspective on mitigation vs. detection in machine learning, 2025. URL https://arxiv.org/abs/2504.20310.

Oded Goldreich. A note on computational indistinguishability. *Information Processing Letters*, 34 (6):277–281, 1990. ISSN 0020-0190. doi: https://doi.org/10.1016/0020-0190(90)90010-U. URL https://www.sciencedirect.com/science/article/pii/002001909090010U.

S Goldwasser and M Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, page 59–68, New York, NY, USA, 1986. Association for Computing Machinery. ISBN 0897911938. doi: 10.1145/12130.12137. URL https://doi.org/10.1145/12130.12137.

S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, page 291–304, New York, NY, USA, 1985. Association for Computing Machinery. ISBN 0897911512. doi: 10.1145/22145.22178. URL https://doi.org/10.1145/22145.22178.

Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 555–564, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320290. doi: 10.1145/2488608.2488678. URL https://doi.org/10.1145/2488608.2488678.

Shafi Goldwasser, Adam Tauman Kalai, Yael Tauman Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models. *ArXiv*, abs/2204.06974, 2022. URL https://api.semanticscholar.org/CorpusID:248177888.

Chenxi Gu, Chengsong Huang, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. Watermarking pre-trained language models with backdooring. *arXiv preprint arXiv:2210.07543*, 2022.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate, 2018. URL https://arxiv.org/abs/1805.00899.

Zhengyuan Jiang, Jinghuai Zhang, and Neil Zhenqiang Gong. Evading watermark based detection of ai-generated content. *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023. URL https://api.semanticscholar.org/CorpusID:258557682.

Adam Tauman Kalai, Adam R Klivans, Yishay Mansour, and Rocco A Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

Michael J. Kearns and Umesh V. Vazirani. *An introduction to computational learning theory*. MIT Press, Cambridge, MA, USA, 1994. ISBN 0262111934.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-Verifier Games improve legibility of LLM outputs, 2024. URL https://arxiv.org/abs/2407.13692.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *CoRR*, abs/2307.15593, 2023. doi: 10.48550/ARXIV.2307.15593. URL https://doi.org/10.48550/arXiv.2307.15593.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Peixuan Li, Pengzhou Cheng, Fangqi Li, Wei Du, Haodong Zhao, and Gongshen Liu. Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14991–14999, 2023.

Richard J. Lipton and Neal E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '94, page 734–740, New York, NY, USA, 1994a. Association for Computing Machinery. ISBN 0897916638. doi: 10.1145/195058.195447. URL https://doi.org/10.1145/195058.195447.

Richard J Lipton and Neal E Young. Simple strategies for large zero-sum games with applications to complexity theory. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 734–740, 1994b.

Chang Liu, Jie Zhang, Han Fang, Zehua Ma, Weiming Zhang, and Nenghai Yu. Dear: A deep-learning-based audio re-recording resilient watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13201–13209, 2023.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Stephen McAleer, John B Lanier, Kevin A Wang, Pierre Baldi, and Roy Fox. Xdo: A double oracle algorithm for extensive-form games. *Advances in Neural Information Processing Systems*, 34: 23128–23139, 2021.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically, 2024.

Erwan Le Merrer, Patrick Pérez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32:9233 – 9244, 2017. URL https://api.semanticscholar.org/CorpusID:11008755.

Omar Montasser, Steve Hanneke, and Nathan Srebro. Transductive robust learning guarantees. In *International Conference on Artificial Intelligence and Statistics*, pages 11461–11471. PMLR, 2022.

Yuki Nagai, Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7:3–16, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Ryota Namba and Jun Sakuma. Robust watermarking of neural network with exponential weighting. *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019. URL https://api.semanticscholar.org/CorpusID:58028915.

Noam Nisan. Pseudorandom generators for space-bounded computations. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 204–212, 1990.

Zhenxing Niu, Yuyao Sun, Qiguang Miao, Rong Jin, and Gang Hua. Towards unified robustness against both backdoor and adversarial attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7589–7605, 2024. doi: 10.1109/TPAMI.2024.3392760.

Ambar Pal and René Vidal. A game theoretic analysis of additive adversarial attacks and defenses. *Advances in Neural Information Processing Systems*, 33:1345–1355, 2020.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark. *arXiv preprint arXiv:2305.10036*, 2023.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=Bys4ob-Rb`.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL `https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf`.

Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 84–93. ACM, 2005.

R. Rivest, L. Adleman, and M. Dertouzos. On data banks and privacy homomorphisms. In *Foundations of Secure Computation*, page 169–179, New York, NY, USA, 1978. Academic Press.

Mingjie Sun, Siddhant Agarwal, and J Zico Kolter. Poisoned classifiers are not only backdoored, they are fundamentally broken. *arXiv preprint arXiv:2010.09080*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6199`.

Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. Better Safe than Sorry: Preventing Delusive Adversaries with Adversarial Training. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713845393.

Stuart A. Thompson Tiffany Hsu. Disinformation researchers raise alarms about a.i. chatbots. `https://scottaaronson.blog/?p=6823`, 2023. Accessed: March 2024.

Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.

Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *Advances in neural information processing systems*, 33:1633–1645, 2020.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on international conference on multimedia retrieval*, pages 269–277, 2017.

J v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Vinod Vaikuntanathan. Computing blindfolded: New developments in fully homomorphic encryption. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, page 5–16, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 9780769543001. doi: 10.1109/FOCS.2011.98. URL `https://doi.org/10.1109/FOCS.2011.98`.

Stephan Wäldchen, Kartikey Sharma, Berkant Turan, Max Zimmer, and Sebastian Pokutta. Interpretability Guarantees with Merlin-Arthur Classifiers. In *International Conference on Artificial Intelligence and Statistics*, pages 1963–1971. PMLR, 2024.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *ArXiv*, abs/2307.02483, 2023. URL https://api.semanticscholar.org/CorpusID:259342528.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 51008–51025. Curran Associates, Inc., 2023a.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust. *ArXiv*, abs/2305.20030, 2023b. URL https://api.semanticscholar.org/CorpusID:258987524.

Cheng-Hsin Weng, Yan-Ting Lee, and Shan-Hung (Brandon) Wu. On the trade-off between adversarial and backdoor robustness. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11973–11983. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8b4066554730ddfaa0266346bdc1b202-Paper.pdf.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5283–5292. PMLR, 2018. URL http://proceedings.mlr.press/v80/wong18a.html.

Yi-Hsuan Wu, Chia-Hung Yuan, and Shan-Hung Wu. Adversarial robustness via runtime masking and cleansing. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10399–10409. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/wu20f.html.

Cihang Xie, Zhishuai Zhang, Jianyu Wang, Yuyin Zhou, Zhou Ren, and Alan Loddon Yuille. Improving transferability of adversarial examples with input diversity. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2725–2734, 2018. URL https://api.semanticscholar.org/CorpusID:3972825.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. *arXiV*, abs/2311.04378, 2023. doi: 10.48550/ARXIV.2311.04378. URL https://doi.org/10.48550/arXiv.2311.04378.

Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph. Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, page 159–172, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355766. doi: 10.1145/3196494.3196550. URL https://doi.org/10.1145/3196494.3196550.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. *CoRR*, abs/2306.17439, 2023a. doi: 10.48550/ARXIV.2306.17439. URL https://doi.org/10.48550/arXiv.2306.17439.

Xuandong Zhao, Kexun Zhang, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. 2023b. URL https://api.semanticscholar.org/CorpusID:259075167.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *ArXiv*, abs/2303.10137, 2023c. URL https://api.semanticscholar.org/CorpusID:257622907.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *ArXiv*, abs/2307.15043, 2023. URL https://api.semanticscholar.org/CorpusID:260202961.

A Additional Methods in Related Work

This section provides an overview of the main areas relevant to our work: Watermarking techniques, adversarial defenses, and transferable attacks on Deep Neural Networks (DNNs). Each subsection outlines important contributions and the current state of research in these areas, offering additional context and details beyond those covered in the main body

A.1 Watermarking

Watermarking techniques are crucial for protecting the intellectual property of machine learning models. These techniques can be broadly categorized based on the type of model they target. We review watermarking schemes for both classification and generative models, with a primary focus on classification models, as our work builds upon these methods.

A.1.1 Watermarking Schemes for classification Models

classification models, which are designed to categorize input data into predefined classes, have been a major focus of watermarking research. The key approaches in this domain can be divided into black-box and white-box approaches.

Black-Box Setting. In the black-box setting, the model owner does not have access to the internal parameters or architecture of the model, but can query the model to observe its outputs. This setting has seen the development of several watermarking techniques, primarily through backdoor-like methods.

Adi et al. [2018] and Zhang et al. [2018] proposed frameworks that embed watermarks using specifically crafted input data (e.g., unique patterns) with predefined outcomes. These watermarks can be verified by feeding these special inputs into the model and checking for the expected outputs, thereby confirming ownership.

Another significant contribution in this domain is by Merrer et al. [2017], who introduced a method that employs adversarial examples to embed the backdoor. Adversarial examples are perturbed inputs that cause the model to produce specific outputs, thus serving as a watermark.

Namba and Sakuma [2019] further enhanced the robustness of black-box watermarking schemes by developing techniques that withstand various model modifications and attacks. These methods ensure that the watermark remains intact and detectable even when the model undergoes transformations.

Provable undetectability of backdoors was achieved in the context of classification tasks by Goldwasser et al. [2022]. Unfortunately, it is known ([Goldwasser et al., 2022]) that some undetectable watermarks are easily removed by simple mechanisms similar to randomized smoothing [Cohen et al., 2019].

The popularity of black-box watermarking is due to its practical applicability, as it does not require access to the model's internal workings. This makes it suitable for scenarios where models are deployed as APIs or services. Our framework builds upon these black-box watermarking techniques.

White-Box Setting. In contrast, the white-box setting assumes that the model owner has full access to the model's parameters and architecture, allowing for direct examination to confirm ownership. The initial methodologies for embedding watermarks into the weights of DNNs were introduced by Uchida et al. [2017] and Nagai et al. [2018]. Uchida et al. [2017] presented a framework for embedding watermarks into the model weights, which can be examined to confirm ownership.

An advancement in white-box watermarking is provided by Darvish Rouhani et al. [2019], who developed a technique to embed an N-bit ($N \geq 1$) watermark in DNNs. This technique is both *data-* and *model-dependent*, meaning the watermark is activated only when specific data inputs are fed into the model. For revealing the watermark, activations from intermediate layers are necessary in the case of white-box access, whereas only the final layer's output is needed for black-box scenarios.

Our work does not focus on white-box watermarking techniques. Instead, we concentrate on exploring the interaction between backdoor-like watermarking techniques, adversarial defenses, and transferable attacks. Overall, watermarking through backdooring has become more popular due to its applicability in the black-box setting.

A.1.2 Watermarking Schemes for Generative Models

Watermarking techniques for generative models have attracted considerable attention with the advent of Large Language Models (LLMs) and other advanced generative models. This increased interest has led to a surge in research and diverse contributions in this area.

Backdoor-Based Watermarking for Pre-trained Language Models. In the domain of Natural Language Processing (NLP), backdoor-based watermarks have been increasingly studied for Pre-trained Language Models (PLMs), as exemplified by works such as [Gu et al., 2022] and [Li et al., 2023]. These methods leverage rare or common word triggers to embed watermarks, ensuring that they remain robust across downstream tasks and resilient to removal techniques like fine-tuning or pruning. While these approaches have demonstrated promising results in practical applications, they are primarily empirical, with theoretical aspects of watermarking and robustness requiring further exploration.

Watermarking the Output of LLMs. Watermarking the generated text of LLMs is critical for mitigating potential harms. Significant contributions in this domain include [Kirchenbauer et al., 2023], who proposed a watermarking framework that embeds signals into generated text that are invisible to humans but detectable algorithmically. This method promotes the use of a randomized set of "green" tokens during text generation, and detects the watermark without access to the language model API or parameters.

Kuditipudi et al. [2023] introduced robust distortion-free watermarks for language models. Their method ensures that the watermark does not distort the generated text, providing robustness against various text manipulations while maintaining the quality of the output.

Zhao et al. [2023a] presented a provable, robust watermarking technique for AI-generated text. This approach offers strong theoretical guarantees for the robustness of the watermark, making it resilient against attempts to remove or alter it without significantly changing the generated text.

However, Zhang et al. [2023] highlighted vulnerabilities in these watermarking schemes. Their work demonstrates that current watermarking techniques can be effectively broken, raising important considerations for the future development of robust and secure watermarking methods for LLMs.

Image Generation Models. Various watermarking techniques have been developed for image generation models to address ethical and legal concerns. Fernandez et al. [2023] introduced a method combining image watermarking with Latent Diffusion Models, embedding invisible watermarks in generated images for future detection. This approach is robust against modifications such as cropping. Wen et al. [2023b] proposed Tree-Ring Watermarking, which embeds a pattern into the initial noise vector during sampling, making the watermark robust to transformations like convolutions and rotations. Jiang et al. [2023] highlighted vulnerabilities in watermarking schemes, showing that human-imperceptible perturbations can evade watermark detection while maintaining visual quality. Zhao et al. [2023c] provided a comprehensive analysis of watermarking techniques for Diffusion Models, offering a recipe for efficiently watermarking models like Stable Diffusion, either through training from scratch or fine-tuning. Additionally, Zhao et al. [2023b] demonstrated that invisible watermarks are vulnerable to regeneration attacks that remove watermarks by adding random noise and reconstructing the image, suggesting a shift towards using semantically similar watermarks for better resilience.

Audio Generation Models. Watermarking techniques for audio generators have been developed for robustness against various attacks. Erfani et al. [2017] introduced a spikegram-based method, embedding watermarks in high-amplitude kernels, robust against MP3 compression and other attacks while preserving quality. Liu et al. [2023] proposed DeAR, a deep-learning-based approach resistant to audio re-recording (AR) distortions.

A.2 Adversarial Defense

The field of adversarial robustness has a rich and extensive literature [Szegedy et al., 2014, Gilmer et al., 2018, Raghunathan et al., 2018, Wong and Kolter, 2018, Engstrom et al., 2017]. Adversarial defenses are essential for ensuring the security and reliability of machine learning models against adversarial attacks that aim to deceive them with carefully crafted inputs.

For classification models, there has been significant progress in developing adversarial defenses. Techniques such as adversarial training [Madry et al., 2018], which involves training the model on adversarial examples, have shown promise in improving robustness. Certified defenses [Raghunathan et al., 2018] provide provable guarantees against adversarial attacks, ensuring that the model's predictions remain unchanged within a specified perturbation bound. Additionally, methods like *randomized smoothing* [Cohen et al., 2019] offer robustness guarantees.

A particularly relevant work for our study is [Goldwasser et al., 2020], which considers a different model for generating adversarial examples. This approach has significant implications for the robustness of watermarking techniques in the face of adversarial attacks.

In the context of LLMs, there is a rapidly growing body of research focused on identifying adversarial examples [Zou et al., 2023, Carlini et al., 2023, Wen et al., 2023a]. This research is closely related to the notion of *jailbreaking* [Andriushchenko et al., 2024, Chao et al., 2023, Mehrotra et al., 2024, Wei et al., 2023], which involves manipulating models to bypass their intended constraints and protections.

A.3 Transferable Attacks and Transductive Learning

Transferable attacks refer to adversarial examples that are effective across multiple models. Moreover, *transductive learning* has been explored as a means to enhance adversarial robustness, and since our Definition 5 captures some notion of transductive learning in the context of Transferable Attacks, we highlight significant contributions in these areas.

Adversarial Robustness via Transductive Learning. Transductive learning [Gammerman et al., 1998] has shown promise in improving the robustness of models by utilizing both training and test data during the learning process. This approach aims to make models more resilient to adversarial perturbations encountered at test time.

One significant contribution is by Goldwasser et al. [2020], which explores learning guarantees in the presence of arbitrary adversarial test examples, providing a foundational framework for transductive robustness. Another notable study by Chen et al. [2021] formalizes transductive robustness and proposes a bilevel attack objective to challenge transductive defenses, presenting both theoretical and empirical support for transductive learning's utility.

Additionally, Montasser et al. [2022] introduce a transductive learning model that adapts to perturbation complexity, achieving a robust error rate proportional to the VC dimension. The method by Wu et al. [2020] improves robustness by dynamically adjusting the network during runtime to mask gradients and cleanse non-robust features, validated through experimental results. Lastly, Tramer et al. [2020] critique the standard of adaptive attacks, demonstrating the need for specific tuning to effectively evaluate and enhance adversarial defenses.

Transferable Attacks on DNNs. Transferable attacks exploit the vulnerability of models to adversarial examples that generalize across different models. For classification models, significant works include Liu et al. [2016], which investigates the transferability of adversarial examples and their effectiveness in black-box attack scenarios, [Xie et al., 2018], who propose input diversity techniques to enhance the transferability of adversarial examples across different models, and [Dong et al., 2019], which presents translation-invariant attacks to evade defenses and improve the effectiveness of transferable adversarial examples.

In the context of generative models, including LLMs and other advanced generative architectures, relevant research is rapidly emerging, focusing on the transferability of adversarial attacks. This area is crucial as it aims to understand and mitigate the risks associated with adversarial examples in these powerful models. Notably, Zou et al. [2023] explored universal and transferable adversarial attacks on aligned language models, highlighting the potential vulnerabilities and the need for robust defenses in these systems.

A.4 Interactive Proof Systems in Machine Learning

Interactive Proof Systems [Goldwasser and Sipser, 1986] have recently gained considerable attention in machine learning for their ability to formalize and verify complex interactions between agents, models, or even human participants. A key advancement in this area is the introduction of *Prover-*

Verifier Games (PVGs) [Anil et al., 2021], which employ a game-theoretic approach to guide learning agents towards decision-making with verifiable outcomes. Building on PVGs, Kirchner et al. [2024] enhance this framework to improve the legibility of Large Language Models (LLMs) outputs, making them more accessible for human evaluation. Similarly, Wäldchen et al. [2024] apply the prover-verifier setup to offer interpretability guarantees for classifiers. Extending these concepts, self-proving models Amit et al. [2024] introduce generative models that not only produce outputs but also generate proof transcripts to validate their correctness. In the context of AI safety, scalable *debate protocols* [Condon et al., 1993, Irving et al., 2018, Brown-Cohen et al., 2023] leverage interactive proof systems to enable complex decision processes to be broken down into verifiable components, ensuring reliability even under adversarial conditions.

		Undetectability	**Unremovability**	**Uniqueness**
Classification	Goldwasser et al. [2022]	✔	robust to some smoothing attacks	✔(E)
	Adi et al. [2018], Zhang et al. [2018]	✔(E)	✗	✔(E)
	Merrer et al. [2017]	✔(E)	robust to fine tunning attacks	✔(E)
LLMs	Christ et al. [2023], Kuditipudi et al. [2023]	✔	✗	✔
	Zhao et al. [2023a]	✗	robust to edit distance attacks only	✔
	Tiffany Hsu [2023]	✔(E)	✗	✔
	Kirchenbauer et al. [2023]	✗	✗	✔

Table 4: Overview of properties across various watermarking schemes. The symbol ✔ denotes properties with formal guarantees or where proof is plausible, whereas ✗ indicates the absence of such guarantees. Entries marked with ✔(E) represent properties observed empirically; these lack formal proof in the corresponding literature, suggesting that deriving such proof may present substantial challenges. The LLM watermarking schemes refer to those applied to text generated by these models.

B Preliminaries

For $n \in \mathbb{N}$ we define $[n] := \{1, \dots, n\}$. We say a boolean sequence $a : \mathbb{N} \to \{0,1\}$ is true with frequency $\alpha \in [0,1]$ if

$$\liminf_{n\to\infty} \frac{\sum_{i\in[n]} a(i)}{n} \geq \alpha.$$

For two sequences $a, b : \mathbb{N} \to \mathbb{R}$ we say they agree with frequency at least $\alpha \in [0,1]$ if the sequence $(a \stackrel{?}{=} b) : \mathbb{N} \to \{0,1\}$, i.e. $(a \stackrel{?}{=} b)(n) = \mathbb{1}_{a(n)=b(n)}$, is true with frequency α.

Learning. For a set Ω, we write $\Delta(\Omega)$ to denote the set of all probability measures defined on the measurable space $(\Omega, \mathcal{F})$, where $\mathcal{F}$ is some fixed σ-algebra that is implicitly understood. For a parameter n, we denote by $\{0,1\}^n$ the input space and by $\{0,1\}$ the output space. A *model* is a function $f : \{0,1\}^n \to \{0,1\}$.

Definition 6 (*Learning Task*)**.** A *learning task* $\mathbb{L}$ is a family $\{\mathbb{L}_n\}_{n\in\mathbb{N}}$, where for every n, $\mathbb{L}_n$ is an element of $\Delta\left(\Delta(\{0,1\}^n) \times \{0,1\}^{\{0,1\}^n}\right)$.

For a *distribution* $\mathcal{D}_n \in \Delta(\{0,1\}^n)$ and a *ground truth* $h_n : \{0,1\}^n \to \{0,1\}$, we define an *error* of f as $\mathrm{err}_{\mathcal{D}_n,h_n}(f) := \mathbb{E}_{x\sim\mathcal{D}_n}[f(x) \neq h(x)]$, where the index of err will often be understood implicitly and omitted in notation. For $\mathcal{D}_n \in \Delta(\{0,1\}^n)$, $h_n : \{0,1\}^n \to \{0,1\}$ we define an *example oracle* $\mathrm{Ex}(\mathcal{D}_n, h_n)$ as an oracle that samples $x \sim \mathcal{D}_n$ and returns $(x, h_n(x))$.

Interaction. When $\mathrm{Ex}(\mathcal{D}, h)$ generates $(x, h(x))$ it is encoded as an $n+1$ bit-string, because $x \in \{0,1\}^n$ and the label space is $\{0,1\}$. For a *message space* $\mathcal{M} = \{\mathcal{M}_n\}_n = \{\{0,1\}^{m(n)}\}_n$ a *representation class* is a collection of mappings $\{\mathcal{R}_n\}_n$, where for every n, $\mathcal{R}_n : \mathcal{M}_n \to \{0,1\}^{\{0,1\}^n}$. Thus, there is a function class corresponding to a representation, i.e., for every n there is a function class $\mathcal{F}_n$, which is an image of $\mathcal{R}_n$. Note that h_n (which is the ground truth) may or may not be in $\mathcal{F}_n$. All function classes considered in this work have an implicit representation class and an underlying message space.

Computation. We work with the collection of Boolean circuits over the standard basis B_2, the set of all two-bit Boolean functions. The size of a circuit C is measured by its number of gates; let $|C|$ denote the size of C. For a circuit family $\mathcal{C} = \{C_n\}_n$ we say it has a circuit complexity $S(n)$ if for every n, $|C_n| \leq S(n)$.

For a distribution $\mathcal{D}_n$ over $\{0,1\}^n$, and a ground truth $h_n : \{0,1\}^n \to \{0,1\}$ we denote by $C^{\mathrm{Ex}(\mathcal{D}_n, h_n)}$ a circuit with some[10] number of specified input gates that are initialized with samples $(x, h(x))$ sampled from $x \sim \mathcal{D}_n$. We will also by interested in interaction between circuits. When messages are exchanged between circuits we assume that there are specified input (output) gates that correspond to outgoing (ingoing) messages. Also, when a circuit is randomized we assume there are designated input gates that are initialized with random bits.

Definition 7 (*Computationally Bounded Learnability*)**.** For $\epsilon, \delta : \mathbb{N} \to (0,1)$ we say that a learning task $\mathbb{L} = \{\mathbb{L}_n\}_{n\in\mathbb{N}}$ is learnable to error ϵ with confidence $1-\delta$ and with circuit complexity $S : \mathbb{N} \to \mathbb{N}$ by a function class $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ (with a corresponding representation class $\mathcal{R}$), or $(\epsilon, \delta, S, \mathcal{F})$-learnable in short, if there exists a circuit family $\mathcal{C} = \{C_n\}_{n\in\mathbb{N}}$ with complexity $S(n)$ such that for every sufficiently large n, with probability $1-\delta$ over the choice of $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$, $C_n^{\mathrm{Ex}(\mathcal{D}_n, h_n)}$ computes an $m(n)$ bit message $m_{f_n} \in \mathcal{M}_n$ such that $\mathcal{R}_n(m_{f_n}) \in \mathcal{F}_n$ has error at most ϵ, i.e. for every sufficiently large n

$$\mathbb{P}_{(\mathcal{D}_n, h_n)\sim\mathbb{L}_n, m_{f_n} \leftarrow C_n^{\mathrm{Ex}(\mathcal{D}_n, h_n)}} \left[\mathrm{err}_{\mathcal{D}_n, h_n}(\mathcal{R}_n(m_{f_n})) \leq \epsilon(n) \right] \geq 1 - \delta(n).$$

We often abuse the notation and use f_n to denote both m_{f_n} as well as $\mathcal{R}_n(m_{f_n})$.

C Formal Definitions

Definition 8 (*Watermark*)**.** Let $\mathbb{L} = \{\mathbb{L}_n\}_n$ be a learning task, and $\mathcal{F} = \{\mathcal{F}_n\}_n$ a function class. Let $S_{\mathbf{A}}, S_{\mathbf{B}}, q : \mathbb{N} \to \mathbb{N}, \epsilon \in \left(0, \frac{1}{2}\right), l, c, s \in (0,1), s < c$, where $S_{\mathbf{B}}(n)$ bounds the circuit size of $\mathbf{B}_n$, and $S_{\mathbf{A}}(n)$ the circuit size of $\mathbf{A}_n$, $q(n)$ the number of queries, ϵ the risk level, c probability that *uniqueness* holds, s probability that *unremovability* and *undetectability* holds, l the learning probability.

We say that a family of circuits $\mathbf{A}^{\textsc{Watermark}} = \{\mathbf{A}_n^{\textsc{Watermark}}\}_n$ *with complexity* $S_{\mathbf{A}}(n)$ implements a watermarking scheme for $\mathbb{L}$ with frequency α, denoted by

$$\mathbf{A}^{\textsc{Watermark}} \in_\alpha \textsc{Watermark}\left(\mathbb{L}, \mathcal{F}, \epsilon, q, S_{\mathbf{A}}, S_{\mathbf{B}}, l, c, s\right),$$

if the following is true with frequency α over parameter n. An interactive protocol in which first $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$ and then $\mathbf{A}_n^{\textsc{Watermark}}$ computes $(f, \mathbf{x})$, $f : \{0,1\}^n \to \{0,1\}, \mathbf{x} \in (\{0,1\}^n)^{q(n)}$, and $\mathbf{B}_n$ outputs $\mathbf{y} = \mathbf{B}_n(f, \mathbf{x}), \mathbf{y} \in \{0,1\}^{q(n)}$, where f is sent using the representation $\mathcal{R}_n$, satisfies the following

- **Correctness** (f has low error). With probability at least l

$$\mathrm{err}(f) \leq \epsilon.$$

- **Uniqueness** (models trained from scratch give low-error answers). There exists a circuit $\mathbf{B}_n$ of size $S_{\mathbf{A}}(n)$ such that with probability at least c

$$\mathrm{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon.$$

[10] We will not specify the sample complexity explicitly. In this paper, we focus only on circuit complexity. The sample complexity is an important parameter to analyze and we leave it for future work. We emphasize that the circuit complexity is an upper bound on the sample complexity.

- **Unremovability** (fast $\mathbf{B}_n$ give high-error answers). For every circuit $\mathbf{B}_n$ *of size at most* $S_{\mathbf{B}}(n)$ with probability at most s

$$\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon.$$

- **Undetectability** (fast $\mathbf{B}_n$ cannot detect that they are tested). On average over $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$, distributions $\mathcal{D}_n^{q(n)}$ and $\mathbf{x} \sim \mathbf{A}_n^{\text{WATERMARK}}$ are $\frac{s}{2}$-indistinguishable for a class of circuits $\mathbf{B}_n$ *of size at most* $S_{\mathbf{B}}(n)$, i.e., for every circuit $\mathbf{B}_n$ of size at most $S_{\mathbf{B}}(n)$ returning one bit,

$$\left| \mathbb{P}_{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \mathbf{x}' \sim \mathcal{D}_n^{q(n)}, (f, \mathbf{x}) \leftarrow \mathbf{A}_n^{\text{WATERMARK}}} [\mathbf{B}(f, \mathbf{x}') = 0] - \mathbb{P}_{(\mathcal{D}_n, h_n) \sim \mathbb{L}, (f, \mathbf{x}) \leftarrow \mathbf{A}_n^{\text{WATERMARK}}} [\mathbf{B}(f, \mathbf{x}) = 0] \right| \leq \frac{s}{2}.$$

Definition 9 (*Adversarial Defense*)**.** Let $\mathbb{L} = \{\mathbb{L}_n\}_n$ be a learning task, and $\mathcal{F} = \{\mathcal{F}_n\}_n$ a function class. Let $S_{\mathbf{A}}, S_{\mathbf{B}}, q : \mathbb{N} \to \mathbb{N}$, $\epsilon \in \left(0, \frac{1}{2}\right)$, $l, c, s \in (0, 1)$, with $s < c$, where $S_{\mathbf{A}}(n)$ bounds the circuit size of $\mathbf{A}_n$, and $S_{\mathbf{B}}(n)$ the circuit size of $\mathbf{B}_n$, $q(n)$ the number of queries, ϵ the error parameter, c the completeness, s the soundness, and l the learning probability.

We say that a family of circuits $\mathbf{B}^{\text{DEFENSE}} = \{\mathbf{B}_n^{\text{DEFENSE}}\}_n$ *with complexity* $S_{\mathbf{A}}(n)$ implements an adversarial defense for $\mathbb{L}$ with frequency α, denoted by

$$\mathbf{B}^{\text{DEFENSE}} \in_\alpha \text{DEFENSE}\left(\mathbb{L}, \mathcal{F}, \epsilon, q, S_{\mathbf{A}}, S_{\mathbf{B}}, l, c, s\right),$$

if the following is true with frequency α over parameter n. An interactive protocol in which first $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$, $\mathbf{B}_n^{\text{DEFENSE}}$ computes $f : \{0,1\}^n \to \{0,1\}$, $\mathbf{A}_n$ replies with $\mathbf{x} = \mathbf{A}_n(f_n)$, $\mathbf{x} \in (\{0,1\}^n)^{q(n)}$, and $\mathbf{B}_n^{\text{DEFENSE}}$ outputs $b = \mathbf{B}_n^{\text{DEFENSE}}(f, \mathbf{x})$, $b \in \{0,1\}$, satisfies the following:

- **Correctness** (f_n has low error). With probability at least l

$$\text{err}(f) \leq \epsilon.$$

- **Completeness** (natural inputs are not flagged as adversarial). When $\mathbf{x} \sim \mathcal{D}_n^{q(n)}$, with probability at least c

$$b = 0.$$

- **Soundness** (adversarial inputs are detected). For every circuit $\mathbf{A}_n$ of size at most $S_{\mathbf{A}}(n)$, with probability at most s

$$\text{err}(\mathbf{x}, f(\mathbf{x})) > 7\epsilon \text{ and } b = 0.$$

Definition 10 (*Transferable Attack*)**.** Let $\mathbb{L} = \{\mathbb{L}_n\}_n$ be a learning task and $\mathcal{F} = \{\mathcal{F}_n\}_n$ a function class. Let $S_{\mathbf{A}}, S_{\mathbf{B}}, q : \mathbb{N} \to \mathbb{N}$, $\epsilon \in \left(0, \frac{1}{2}\right)$, and $c, s \in (0, 1)$, with $s < c$, where $S_{\mathbf{A}}(n)$ bounds the circuit size of $\mathbf{A}_n$, and $S_{\mathbf{B}}$ the circuit size of $\mathbf{B}_n$, $q(n)$ the number of queries, ϵ the error parameter, c the *transferability* probability, and s the *undetectability* probability.

We say that a family of circuits $\mathbf{A}^{\text{TRANSFATTACK}} = \{\mathbf{A}_n^{\text{TRANSFATTACK}}\}$ *with complexity* $S_{\mathbf{A}}(n)$ implements a transferable attack for $\mathbb{L}$ with frequency α, denoted by

$$\mathbf{A}^{\text{TRANSFATTACK}} \in_\alpha \text{DEFENSE}\left(\mathbb{L}, \mathcal{F}, \epsilon, q, S_{\mathbf{A}}, S_{\mathbf{B}}, l, c, s\right),$$

if the following is true with frequency α over parameter n. An interactive protocol in which first $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$, $\mathbf{A}_n^{\text{TRANSFATTACK}}$ computes $\mathbf{x} \in (\{0,1\}^n)^{q(n)}$, and $\mathbf{B}_n$ outputs $\mathbf{y} = \mathbf{B}_n(\mathbf{x})$, $\mathbf{y} \in (\{0,1\})^{q(n)}$, satisfies the following:

- **Transferability** (fast provers return high-error answers). For every circuit $\mathbf{B}_n$ of size at most $S_{\mathbf{B}}(n)$, with probability at least c

$$\text{err}(\mathbf{x}, \mathbf{y}) > 2\epsilon.$$

- **Undetectability** (fast provers cannot detect that they are tested). On average over $(\mathcal{D}_n, h_n) \sim \mathbb{L}_n$, distributions $\mathbf{x} \sim \mathcal{D}_n^{q(n)}$ and $\mathbf{x} := \mathbf{A}_n^{\text{TRANSFATTACK}}$ are $\frac{s}{2}$-indistinguishable for every circuit $\mathbf{B}_n$ of size at most $S_{\mathbf{B}}(n)$, i.e.,

$$\left| \mathbb{P}_{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \mathbf{x}' \sim \mathcal{D}_n^{q(n)}} [\mathbf{B}_n(\mathbf{x}') = 0] - \mathbb{P}_{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n} [\mathbf{B}_n(\mathbf{x}) = 0] \right| \leq \frac{s}{2}.$$

D Main Theorem

Before proving our main theorem we recall a result from Lipton and Young [1994a] about simple strategies for large zero-sum games.

Game theory. A *two-player zero-sum game* is specified by a payoff matrix $\mathcal{G}$. $\mathcal{G}$ is an $r \times c$ matrix. MIN, the row player, chooses a probability distribution p_1 over the rows. MAX, the column player, chooses a probability distribution p_2 over the columns. A row i and a column j are drawn from p_1 and p_2 and MIN pays $\mathcal{G}_{ij}$ to MAX. MIN tries to minimize the expected payment; MAX tries to maximize it.

By the Min-Max Theorem, there exist optimal strategies for both MIN and MAX. Optimal means that playing first and revealing one's mixed strategy is not a disadvantage. Such a pair of strategies is also known as a Nash equilibrium. The expected payoff when both players play optimally is known as the value of the game and is denoted by $\mathcal{V}(\mathcal{G})$.

We will use the following theorem from Lipton and Young [1994a], which says that optimal strategies can be approximated by uniform distributions over sets of pure strategies of size $O(\log(c))$.

Theorem 4 (Lipton and Young [1994a])**.** *Let $\mathcal{G}$ be an $r \times c$ payoff matrix for a two-player zero-sum game. For any $\eta \in (0,1)$ and $k \geq \frac{\log(c)}{2\eta^2}$ there exists a multiset of pure strategies for the* MIN *(row player) of size k such that a mixed strategy p_1 that samples uniformly from this multiset satisfies*

$$\max_j \sum_i p_1(i)\mathcal{G}_{ij} \leq \mathcal{V}(\mathcal{G}) + \eta(\mathcal{G}_{max} - \mathcal{G}_{min}),$$

where $\mathcal{G}_{max}, \mathcal{G}_{min}$ denote the maximum and minimum entry of $\mathcal{G}$ respectively. The symmetric result holds for the MAX *player.*

We are ready to prove our main theorem.

Theorem 5. *Let $\epsilon \in \left(0, \frac{1}{2}\right), \delta \in \left(0, \frac{1}{48}\right), S : \mathbb{N} \to \mathbb{N}$. For every learning task $\mathbb{L} = \{\mathbb{L}_n\}_n$ learnable to error ϵ with confidence $1-\delta$ and circuit complexity $O\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right)$ and for every family of function classes $\mathcal{F} = \{\mathcal{F}_n\}_n$, every query bound $q(n)$ such that $\frac{\sqrt{S(n)}}{\log(S(n))} = \Omega(m(n) + q(n) \cdot n)$ at least one of the three*

$$\textsc{Watermark}\left(\mathbb{L}, \mathcal{F}, \epsilon, q, S(n), o\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right), l = \frac{10}{24}, c = \frac{21}{24}, s = \frac{19}{24}\right),$$

$$\textsc{Defense}\left(\mathbb{L}, \mathcal{F}, \epsilon, q, o\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right), O(S(n)), l = 1 - \frac{1}{48}, c = \frac{13}{24}, s = \frac{11}{24}\right),$$

$$\textsc{TransfAttack}\left(\mathbb{L}, \mathcal{F}, \epsilon, q, S(n), S(n), c = \frac{3}{24}, s = \frac{19}{24}\right)$$

exists with frequency $\frac{1}{3}$.

Proof. Let $\epsilon \in (0, \frac{1}{2})$ and $q : \mathbb{N} \to \mathbb{N}$ be a query bound. Let $\mathbb{L}$ be a learning task learnable to error ϵ with confidence $1-\delta$ and complexity $S(n)$.

We will consider every n separately and show that for every n, one of the three schemes exists. This automatically implies that one of the schemes exists with frequency at least $\frac{1}{3}$.

Let $s(n) = \Theta\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right)$, where the exact constants will be determined later. Let $\mathfrak{Candidate}_{\mathfrak{W}}(n)$ be a set of $s(n)$-sized circuits computing $(f, \mathbf{x})$. Recall that the execution of a $\mathbf{A}_n \in \mathfrak{C}_{\mathfrak{W}}(n)$ proceeds by first sampling from $\mathrm{Ex}(\mathcal{D}_n, h_n)$ and providing these samples as inputs to $\mathbf{A}_n$ and then running $\mathbf{A}_n$ to obtain $m + q \cdot n$ bits. The first m bits are interpreted as a representation of f (according to $\mathcal{R}_n$), and the following consecutive blocks of n bits each are interpreted as q elements of $\{0,1\}^n$. Similarly, let $\mathfrak{C}_{\mathfrak{D}}(n)$ be a set of $s(n)$-sized circuits accepting as input $(f, \mathbf{x})$ and outputting $(\mathbf{y}, b)$,

where $\mathbf{y} \in \{0,1\}^q, b \in \{0,1\}$. Formally, this is a set of circuits with up to $s(n)$ input gates and $q+1$ output gates. We interpret $\mathfrak{C}_{\mathfrak{W}}(n)$ as candidate algorithms for a watermark, and $\mathfrak{C}_{\mathfrak{D}}(n)$ as candidate algorithms for attacks on watermarks.

For every n define a zero-sum game $\mathcal{G}_n$ between $\mathbf{A}_n \in \mathfrak{C}_{\mathfrak{W}}(n), \mathbf{B}_n \in \mathfrak{C}_{\mathfrak{D}}(n)$. The payoff is given by

$$\mathcal{G}_n(\mathbf{A}_n, \mathbf{B}_n) = \frac{1}{2}\,\mathbb{P}_{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,(f,\mathbf{x}):=\mathbf{A}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)},(\mathbf{y},b):=\mathbf{B}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)}}\Big[\mathrm{err}(f) > \epsilon \text{ or } \mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon \text{ or } b = 1\Big]$$
$$+ \frac{1}{2}\,\mathbb{P}_{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,f:=\mathbf{A}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)},\mathbf{x}\sim\mathcal{D}_n^{q(n)},(\mathbf{y},b):=\mathbf{B}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)}}\Big[\mathrm{err}(f) > \epsilon \text{ or } \Big(\mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon \text{ and } b = 0\Big)\Big],$$

where $\mathbf{A}_n$ tries to minimize and $\mathbf{B}_n$ maximize the payoff.

Then the number of possible circuits is bounded by

$$|\mathfrak{C}_{\mathfrak{W}}| \le (3s(n)^2)^{s(n)} \le 2^{3s(n)\log(s(n))},$$

because every internal gate of a circuit is one of AND, OR, and NOT, and is connected to 2 gates out of at most $s(n)$ choices.

Applying Theorem 4 to $\mathcal{G}_n$ with $\eta = 2^{-5}$ we get two probability distributions, p over a multiset of pure strategies in $\mathfrak{C}_{\mathfrak{W}}$ and r over a multiset of pure strategies in $\mathfrak{C}_{\mathfrak{D}}$ that lead to a 2^{-5}-approximate Nash equilibrium. The size $k(n)$ of the multisets is bounded

$$\begin{aligned} k(n) &\le 2^6 \log\left(|\mathfrak{C}_{\mathfrak{W}}|\right) \\ &\le O(s(n)\log(s(n))). \end{aligned} \tag{1}$$

Next, observe that the mixed strategy corresponding to the distribution p can be represented by a circuit of size

$$\begin{aligned} &k(n)\cdot s(n)\cdot O(\log(k(n))) \\ &\le O(s^2(n)\cdot\log^3(s(n))) && \text{By equation (1)} \\ &\le S(n), \end{aligned}$$

because we can create a circuit that is a collection of $k(n)$ circuits corresponding to the multiset of p, where each one is of size $s(n)$ with additional gadgets of size $O(\log(k))$ activating the corresponding gate depending on the randomness determining a strategy. This implies that p can be implemented by a $S(n)$-sized circuit. The same holds for r. Let's call the strategy corresponding to p, $\mathbf{A}^n_{\text{Nash}}$, and the strategy corresponding to r, $\mathbf{B}^n_{\text{Nash}}$.

Consider cases:

Case $\mathcal{G}(\mathbf{A}_n^{\text{NASH}}, \mathbf{B}_n^{\text{NASH}}) \ge \frac{19}{24}$. Define $\mathbf{B}_n^{\text{DEFENSE}}$ to work as follows:

1. Simulate the circuit of size $O\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right)\mathbf{L}_n$ that learns f, such that
$$\mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ f\leftarrow\mathbf{L}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)}}}\Big[\mathrm{err}(f) \le \epsilon\Big] \ge 1 - \frac{1}{48}.$$
2. Send f to $\mathbf{A}_n$.
3. Receive $\mathbf{x}$ from $\mathbf{A}_n$.
4. Simulate $(\mathbf{y}, b) := \mathbf{B}_n^{\text{NASH}}(f, \mathbf{x})$.
5. Return $b' = 1$ if $b = 1$ or $d(f(\mathbf{x}), \mathbf{y}) > 3\epsilon \cdot q(n)$ and $b' = 0$ otherwise,

where $d(\cdot,\cdot)$ is the Hamming distance. $\mathbf{B}_n^{\text{DEFENSE}}$ can be implemented by circuit of size $O(S(n))$, because it simulates a circuit of size $O\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right)$, then simulating $\mathbf{B}_n^{\text{NASH}}$ of size $S(n)$, and computing a predicate $d(f(\mathbf{x}), \mathbf{y}) > 3\epsilon q$, which can be done in size $\log(q(n))$.

We claim that

$$\textsc{Defense}\left(\mathbb{L}_n, \mathcal{F}_n, \epsilon, q(n), o\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right), O(S(n)), l = 1 - \frac{1}{48}, c = \frac{13}{24}, s = \frac{11}{24}\right). \quad (2)$$

Assume towards contradiction that completeness or soundness of $\mathbf{B}_n^{\textsc{Defense}}$ as defined in Definition 9 does not hold.

If completeness of $\mathbf{B}_n^{\textsc{Defense}}$ does not hold, then

$$\mathbb{P}_{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \mathbf{x} \sim \mathcal{D}_n^q}\Big[b' = 0\Big] < \frac{13}{24}. \quad (3)$$

Let us compute the payoff of $\mathbf{A}_n$, which first runs $f \leftarrow \mathbf{L}_n^{\text{Ex}(\mathcal{D}_n, h_n)}$ (where $\mathbf{L}_n$ is the learning circuit) and sets $\mathbf{x} \sim \mathcal{D}^q$, in the game $\mathcal{G}_n$, when playing against $\mathbf{B}_n^{\textsc{Nash}}$

$$\begin{aligned}
&\mathcal{G}(\mathbf{A}_n, \mathbf{B}_n^{\textsc{Nash}}) \\
&= \frac{1}{2}\mathbb{P}_{\substack{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \\ (f, \mathbf{x}) \leftarrow \mathbf{A}_n^{\text{Ex}(\mathcal{D}_n, h_n)}}}\Big[\text{err}(f) > \epsilon \text{ or } \text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon \text{ or } b' = 1\Big] \\
&+ \frac{1}{2}\mathbb{P}_{\substack{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \\ f \leftarrow \mathbf{A}_n^{\text{Ex}(\mathcal{D}_n, h_n)}, \\ \mathbf{x} \sim \mathcal{D}_n^q}}\Big[\text{err}(f) > \epsilon \text{ or } \Big(\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon \text{ and } b' = 0\Big)\Big] \\
&\leq \delta + \frac{1}{2}\mathbb{P}_{\substack{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \\ f \leftarrow \mathbf{L}_n^{\text{Ex}(\mathcal{D}_n, h_n)}, \\ \mathbf{x} \sim \mathcal{D}_n^q}}\Big[\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon \text{ or } b' = 1\Big] \\
&+ \frac{1}{2}\mathbb{P}_{\substack{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \\ f \leftarrow \mathbf{L}_n^{\text{Ex}(\mathcal{D}_n, h_n)}, \\ \mathbf{x} \sim \mathcal{D}_n^q}}\Big[\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon \text{ and } b' = 0\Big] && \text{Def. of } \mathbf{A}_n, \mathbf{B}_n^{\textsc{Defense}}, \mathbb{P}\Big[\text{err}(f) \leq \epsilon\Big] \geq \frac{47}{48} \\
&< \frac{1}{48} + \frac{1}{2} + \frac{\frac{13}{24}}{2} && \text{By equation (3)} \\
&= \frac{38}{48} \\
&\leq \mathcal{G}(\mathbf{A}_n^{\textsc{Nash}}, \mathbf{B}_n^{\textsc{Nash}}), \ \lightning
\end{aligned}$$

where the contradiction is with the properties of Nash equilibria.

Assume that $\mathbf{A}_n$ breaks the soundness of $\mathbf{B}_n^{\textsc{Defense}}$, which translates to

$$\mathbb{P}_{\substack{(\mathcal{D}_n, h_n) \sim \mathbb{L}_n, \\ \mathbf{x} \leftarrow \mathbf{A}_n(f)}}\Big[\text{err}(\mathbf{x}, f(\mathbf{x})) > 7\epsilon \ \text{ and } \ b = 0 \ \text{ and } \ d(f(\mathbf{x}), \mathbf{y})) > 3\epsilon q\Big] > \frac{11}{24}. \quad (4)$$

Let $\mathbf{A}'_n$ first simulate $f \leftarrow \mathbf{L}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)}$, then runs $\mathbf{x} \leftarrow \mathbf{A}_n(f)$, and returns $(f,\mathbf{x})$. We have

$$
\begin{aligned}
&\mathcal{G}(\mathbf{A}'_n, \mathbf{B}_n^{\text{NASH}}) \\
&= \frac{1}{2}\,\mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ (f,\mathbf{x})\leftarrow\mathbf{A}'_n}}\Big[\mathrm{err}(f) > \epsilon \text{ or } \mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon \text{ or } b' = 1\Big] \\
&+ \frac{1}{2}\,\mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ f\leftarrow\mathbf{A}'_n,\\ \mathbf{x}\sim\mathcal{D}_n^q}}\Big[\mathrm{err}(f) > \epsilon \text{ or } \Big(\mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon \text{ and } b' = 0\Big)\Big] \\
&= \frac{1}{2}\,\mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ f\leftarrow\mathbf{L}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)},\\ \mathbf{x}=\mathbf{A}_n(f)}}\Big[\mathrm{err}(f) > \epsilon \text{ or } \mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon \text{ or } b' = 1\Big] \\
&+ \frac{1}{2}\,\mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ f\leftarrow\mathbf{L}_n^{\mathrm{Ex}(\mathcal{D}_n,h_n)},\\ \mathbf{x}\sim\mathcal{D}_n^q}}\Big[\mathrm{err}(f) > \epsilon \text{ or } \Big(\mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon \text{ and } b' = 0\Big)\Big] && \text{By def. of } \mathbf{A}'_n \\
&< \frac{1}{2} + \frac{1 - \frac{11}{24}}{2} && \text{By equation (4)} \\
&= \frac{37}{48} \\
&\le \mathcal{G}_n(\mathbf{A}_n^{\text{NASH}}, \mathbf{B}_n^{\text{NASH}}),\ \lightning
\end{aligned}
$$

where the contradiction is with the properties of Nash equilibria. Thus equation (2) holds.

Case $\mathcal{G}_n(\mathbf{A}_n^{\text{NASH}}, \mathbf{B}_n^{\text{NASH}}) < \frac{19}{24}$. Consider $\mathbf{B}_n$ that returns $(f(\mathbf{x}), b)$ for a uniformly random b. We have

$$
\mathcal{G}_n(\mathbf{A}_n^{\text{NASH}}, \mathbf{B}_n) \ge \left(1 - \mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ f\leftarrow\mathbf{A}_n^{\text{NASH}}}}\Big[\mathrm{err}(f) \le \epsilon\Big]\right) + \mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ f\leftarrow\mathbf{A}_n^{\text{Nash}}}}\Big[\mathrm{err}(f) \le \epsilon\Big] \cdot \frac{1}{2},
$$

because when $\mathbf{x} \sim \mathcal{D}_n^q$ and $\mathrm{err}(f) \le \epsilon$ the probability that $\mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon$ and $b = 0$ is $\frac{1}{2}$, and similarly when $\mathbf{x} \leftarrow \mathbf{A}_n^{\text{NASH}}$ then the probability that $b = 1$ is equal $\frac{1}{2}$. The assumption that $\mathcal{G}_n(\mathbf{A}_n^{\text{Nash}}, \mathbf{B}_n) < \frac{19}{24}$ and properties of Nash equilibria imply that $\mathbb{P}_{\substack{(\mathcal{D}_n,h_n)\sim\mathbb{L}_n,\\ f\leftarrow\mathbf{A}_n^{\text{Nash}}}}[\mathrm{err}(f) \le \epsilon] \ge \frac{10}{24}$. This implies that *correctness* holds for $\mathbf{A}_n^{\text{Nash}}$ with $l = \frac{10}{24}$.

Next, assume towards contradiction that *unremovability* of $\mathbf{A}_n^{\text{NASH}}$ does not hold, i.e., there is $\mathbf{B}_n$ running in time $o\left(\sqrt{S(n)}/\log(S(n))\right)$ such that $\mathbb{P}\big[\mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon\big] > \frac{19}{24}$. Consider $\mathbf{B}'_n$ that on input $(f,\mathbf{x})$ returns $(\mathbf{B}_n(f,\mathbf{x}), 0)$. Then by definition of $\mathcal{G}_n$, $\mathcal{G}_n(\mathbf{A}_{\text{NASH}}, \mathbf{B}'_n) > \frac{19}{24}$, which is a contradiction $\lightning$.

Next, assume towards contradiction that *undetectability* of $\mathbf{A}_n^{\text{NASH}}$ does not hold, i.e., there exists $\mathbf{B}_n$ such that it distinguishes $\mathbf{x} \sim \mathcal{D}_n^q$ from $\mathbf{x} \leftarrow \mathbf{A}_n^{\text{NASH}}$ with probability higher than $\frac{19}{24}$. Consider $\mathbf{B}'_n$ that on input $(f,\mathbf{x})$ returns $(f(\mathbf{x}), \mathbf{B}_n(f,\mathbf{x}))$.[11] Then by definition of $\mathcal{G}_n$, $\mathcal{G}_n(\mathbf{A}_n^{\text{NASH}}, \mathbf{B}'_n) > \frac{19}{24}$, which is a contradiction $\lightning$.

There are two further subcases. If $\mathbf{A}_n^{\text{NASH}}$ satisfies *uniqueness* then

$$
\mathbf{A}_n^{\text{NASH}} \in \textsc{Watermark}\left(\mathbb{L}_n, \mathcal{F}_n, \epsilon, q(n), S(n), o\left(\frac{\sqrt{S(n)}}{\log(S(n))}\right), l = \frac{10}{24}, c = \frac{21}{24}, s = \frac{19}{24}\right).
$$

If $\mathbf{A}_n^{\text{NASH}}$ does not satisfy *uniqueness*, then, by definition, every succinctly representable circuit $\mathbf{B}_n$ of size $o\left(\sqrt{S(n)}/\log(S(n))\right)$ satisfies $\mathrm{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon$ with probability at most $\frac{21}{24}$. Consider the following $\mathbf{A}_n$. It computes $(f,\mathbf{x}) \leftarrow \mathbf{A}_n^{\text{Nash}}$, ignores f and sends $\mathbf{x}$ to $\mathbf{B}_n$. By the assumption that *uniqueness* is not satisfied for $\mathbf{A}_n^{\text{NASH}}$ *transferability* of Definition 5 holds for $\mathbf{A}_n$ with $c = \frac{3}{24}$. Note

[11] Formally $\mathbf{B}_n$ receives as input $(f,\mathbf{x})$ and not only $\mathbf{x}$.

that $\mathbf{B}_n$ in the transferable attack does not receive f but it makes it no easier for it to satisfy the properties. Note that *undetectability* still holds with the same parameter. Thus

$$\mathbf{A}_n^{\text{NASH}} \in \text{TRANSFATTACK}\left(\mathbb{L}_n, \mathcal{F}_n, \epsilon, q(n), S(n), S(n), c = \frac{3}{24}, s = \frac{19}{24}\right).$$

□

E Fully Homomorphic Encryption (FHE)

We include a definition of fully homomorphic encryption based on the definition from Goldwasser et al. [2013]. The notion of fully homomorphic encryption was first proposed by Rivest, Adleman and Dertouzos Rivest et al. [1978] in 1978. The first fully homomorphic encryption scheme was proposed in a breakthrough work by Gentry in 2009 Gentry [2009]. A history and recent developments on fully homomorphic encryption is surveyed in [Vaikuntanathan, 2011].

E.1 Preliminaries

We say that a function f is *negligible* in an input parameter λ, if for all $d > 0$, there exists K such that for all $\lambda > K$, $f(\lambda) < \lambda^{-d}$. For brevity, we write: for all sufficiently large λ, $f(\lambda) = \text{negl}(\lambda)$. We say that a function f is *polynomial* in an input parameter λ, if there exists a polynomial p such that for all λ, $f(\lambda) \leq p(\lambda)$. We write $f(\lambda) = \text{poly}(\lambda)$. A similar definition holds for $\text{polylog}(\lambda)$. For two polynomials p, q, we say $p \leq q$ if for every $\lambda \in \mathbb{N}$, $p(\lambda) \leq q(\lambda)$.

When saying that a Turing machine $\mathcal{A}$ is p.p.t. we mean that $\mathcal{A}$ is a non-uniform probabilistic polynomial-time machine.

E.2 Definitions

Definition 11 (Goldwasser et al. [2013])**.** A homomorphic (public-key) encryption scheme FHE is a quadruple of polynomial time algorithms (FHE.KEYGEN, FHE.ENC, FHE.DEC, FHE.EVAL) as follows:

- FHE.$\text{KEYGEN}(1^\lambda)$ is a probabilistic algorithm that takes as input the security parameter 1^λ and outputs a public key pk and a secret key sk.
- FHE.$\text{ENC}(pk, x \in \{0,1\})$ is a probabilistic algorithm that takes as input the public key pk and an input bit x and outputs a ciphertext ψ.
- FHE.$\text{DEC}(sk, \psi)$ is a deterministic algorithm that takes as input the secret key sk and a ciphertext ψ and outputs a message $x^* \in \{0,1\}$.
- FHE.$\text{EVAL}(pk, C, \psi_1, \psi_2, \ldots, \psi_n)$ is a deterministic algorithm that takes as input the public key pk, some circuit C that takes n bits as input and outputs one bit, as well as n ciphertexts $\psi_1, \ldots, \psi_n$. It outputs a ciphertext ψ_C.

Compactness: For all security parameters λ, there exists a polynomial $p(\cdot)$ such that for all input sizes n, for all $x_1, \ldots, x_n$, for all C, the output length of FHE.EVAL is at most $p(n)$ bits long.

Definition 12 (*C-homomorphism, Goldwasser et al. [2013]*)**.** Let $C = \{C_n\}_{n\in\mathbb{N}}$ be a class of boolean circuits, where C_n is a set of boolean circuits taking n bits as input. A scheme FHE is C-homomorphic if for every polynomial $n(\cdot)$, for every sufficiently large security parameter λ, for every circuit $C \in C_n$, and for every input bit sequence $x_1, \ldots, x_n$, where $n = n(\lambda)$,

$$\mathbb{P}\left[\begin{array}{c}(pk, sk) \leftarrow \text{FHE.KEYGEN}(1^\lambda);\\ \psi_i \leftarrow \text{FHE.ENC}(pk, x_i) \text{ for } i = 1 \ldots n;\\ \psi \leftarrow \text{FHE.EVAL}(pk, C, \psi_1, \ldots, \psi_n) :\\ \text{FHE.DEC}(sk, \psi) \neq C(x_1, \ldots, x_n)\end{array}\right] = \text{negl}(\lambda),$$

where the probability is over the coin tosses of FHE.KEYGEN and FHE.ENC.

Definition 13 (*Fully homomorphic encryption*)**.** A scheme FHE is fully homomorphic if it is homomorphic for the class of all arithmetic circuits over $\mathbb{GF}(2)$.

Definition 14 (*Leveled fully homomorphic encryption*)**.** A leveled fully homomorphic encryption scheme is a homomorphic scheme where FHE.KEYGEN receives an additional input 1^d and the resulting scheme is homomorphic for all depth-d arithmetic circuits over $\mathbb{GF}(2)$.

Definition 15 (*IND-CPA security*)**.** A scheme FHE is IND-CPA secure if for any p.p.t. adversary $\mathcal{A}$,

$$\Big| \mathbb{P}\left[(pk, sk) \leftarrow \text{FHE.KEYGEN}(1^\lambda) : \mathcal{A}(pk, \text{FHE.ENC}(pk, 0)) = 1\right] + \\ - \mathbb{P}\left[(pk, sk) \leftarrow \text{FHE.KEYGEN}(1^\lambda) : \mathcal{A}(pk, \text{FHE.ENC}(pk, 1)) = 1\right] \Big| = \text{negl}(\lambda).$$

We now state the result of Brakerski, Gentry, and Vaikuntanathan [Brakerski et al., 2012] that shows a leveled fully homomorphic encryption scheme based on a standard assumption in cryptography called Learning with Errors [Regev, 2005]:

Theorem 6 (*Fully Homomorphic Encryption, definition from Goldwasser et al. [2013]*)**.** *Assume that there is a constant $0 < \epsilon < 1$ such that for every sufficiently large ℓ, the approximate shortest vector problem gapSVP in ℓ dimensions is hard to approximate to within a $2^{O(\ell^\epsilon)}$ factor in the worst case. Then, for every n and every polynomial $d = d(n)$, there is an IND-CPA secure d-leveled fully homomorphic encryption scheme where encrypting n bits produces ciphertexts of length poly$(n, \lambda, d^{1/\epsilon})$, the size of the circuit for homomorphic evaluation of a function f is size$(C_f) \cdot$ poly$(n, \lambda, d^{1/\epsilon})$ and its depth is depth$(C_f) \cdot$ poly$(\log n, \log d)$.*

F Existence of Transferable Attacks

Learning Theory Preliminaries. For the next lemma, we will consider a slight generalization of learning tasks to the case where there are many valid outputs for a given input. This can be understood as the case of generative tasks. More concretely, we assume that for the input space $\mathcal{X}_n$ the output space is $\mathcal{Y}_n$ instead of $\{0,1\}$. It will always be the case that $\mathcal{X}_n$ and $\mathcal{Y}_n$ are equal to $\{0,1\}^{p(n)}$ for some polynomial p. For a distribution $\mathcal{D}_n$ over $\mathcal{X}_n$ we call a function $h : \mathcal{X}_n \times \mathcal{Y}_n \to \{0,1\}$ an error oracle if the error of a function $f : \mathcal{X}_n \to \mathcal{Y}_n$ is defined as

$$\text{err}(f) := \mathbb{E}_{x \sim \mathcal{D}}[h(x, f(x))],$$

where the randomness of expectation includes the potential randomness of f. The example oracle Ex provides access to samples $(x, y) \in \mathcal{X}_n \times \mathcal{Y}_n$, where $x \sim \mathcal{D}_n$ and $y \in \mathcal{Y}_n$ is some y such that $h(x, y) = 0$.

The following learning task will be crucial for our construction.

Definition 16 (*Lines on a Circle Learning Task $\mathbb{L}^\circ$*)**.** We define $\mathbb{L}^\circ = \{\mathbb{L}^\circ_n\}_n$. For every n we define $\mathcal{X}_n = \{0,1\}^n$ and associate $\mathcal{X}_n$ with vertices of a 2^n regular polygon inscribed in the unit circle $\{x \in \mathbb{R}^2 \mid \|x\|_2 = 1\}$. The output space is $\{-1, +1\}$ for all n. Let $\mathcal{H} := \{h_w \mid w \in \mathbb{R}^2, \|w\|_2 = 1\}$, where $h_w(x) := \text{sgn}(\langle w, x\rangle)$. For every n, let $\mathbb{L}^\circ_n$ be the distribution corresponding to the following process: sample $h_w \sim U(\mathcal{H})$, return $(U(\mathcal{X}_n), h_w)$. Note that $\mathcal{H}$ has VC-dimension equal to 2 so $\mathbb{L}$ is learnable to error ϵ with $O(\frac{1}{\epsilon})$ samples for every n and every ϵ.

Moreover, for $n \in \mathbb{N}$ define $B^w_n(\alpha) := \{x \in \mathcal{X}_n \mid |\measuredangle(x, w)| \le \alpha\}$.

Lemma 3 (*Learning lower bound for $\mathbb{L}^\circ$*)**.** *Let $n \in \mathbb{N}$. Let $\mathbf{L}_n$ be a learning algorithm for $\mathbb{L}^\circ_n$ (Definition 16) that uses K samples and returns a classifier $f : \mathcal{X}_n \to \{-1, +1\}$. Then*

$$\mathbb{P}_{(\mathcal{D}_n, h_n) \sim \mathbb{L}^\circ_n, f \leftarrow \mathbf{L}^{Ex(\mathcal{D}_n, h_n)}}\left[\mathbb{P}_{x \sim \mathcal{D}_n}[f(x) \neq h_w(x)] \le \frac{1}{2K}\right] \le \frac{3}{100}.$$

Proof. Let $n \in \mathbb{N}$. Consider the following algorithm $\mathcal{A}$. It first simulates $\mathbf{L}_n$ on K samples to compute f. Next, it performs a smoothing of f, i.e., computes

$$f_\eta(x) := \begin{cases} +1, & \text{if } \mathbb{P}_{x' \sim U(B^x_n(2\pi\eta))}[f(x') = +1] > \mathbb{P}_{x' \sim U(B^x_n(2\pi\eta))}[f(x') = -1] \\ -1, & \text{otherwise.} \end{cases}$$

Note that if $\text{err}(f) \le \eta$ for a ground truth h_w then for every $x \in \mathcal{X}_n \setminus B^x_n(2\pi\eta)$ we have $f_\eta(x) = h_w(x)$. This implies that $\mathcal{A}$ can be adapted to an algorithm that with probability 1 finds w' such that $|\measuredangle(w, w')| \le \text{err}(f)$.

Assuming towards contradiction that the statement of the lemma does not hold it means that there is an algorithm using K samples that with probability $\frac{3}{100}$ locates w up to angle $\frac{1}{2K}$.

Consider any algorithm $\mathcal{A}$ using K samples. Probability that $\mathcal{A}$ does not see any sample in $B_n^w(2\pi\eta)$ is at least

$$(1-4\eta)^K \geq \left((1-4\eta)^{\frac{1}{4\eta}}\right)^{4\eta K} \geq \left(\frac{1}{2e}\right)^{4\eta K},$$

which is bigger than $1-\frac{3}{100}$ if we set $\eta = \frac{1}{2K}$. But note that if there is no sample in $B_n^w(2\pi\eta)$ then $\mathcal{A}$ cannot locate w up to η with certainty. This proves the lemma. □

Lemma 4 (*Boosting for $\mathbb{L}^\circ$*)**.** *Let $\eta, \nu \in (0, \frac{1}{4}), n \in \mathbb{N}$, $\mathbf{L}_n$ be a learning algorithm for $\mathbb{L}_n^\circ$ that uses K samples and outputs $f : \mathcal{X}_n \to \{-1, +1\}$ such that with probability δ*

$$\mathbb{P}_{w\sim U(\mathcal{H}), x\sim U(B_n^w(2\pi\eta))}[f(x) \neq h_w(x)] \leq \nu, \tag{5}$$

where $\mathcal{H}$ is as defined earlier $\{h_w \mid w \in \mathbb{R}^2, \|w\|_2 = 1\}$. Then there exists a learning algorithm $\mathbf{L}'_n$ for $\mathbb{L}_n^\circ$ that uses $\max\left(K, \frac{9}{\eta}\right)$ samples such that with probability $\delta - \frac{1}{1000}$ returns f' such that

$$\mathbb{P}_{w\sim U(\mathcal{H}), x\sim U(\mathcal{X}_n)}[f'(x) \neq h_w(x)] \leq 4\eta\nu.$$

Proof. Let $n \in \mathbb{N}$. $\mathbf{L}'_n$ first draws $\max\left(K, \frac{9}{\eta}\right)$ samples Q and defines $g : \mathcal{X}_n \to \{-1, +1, \perp\}$ as follows, g maps to -1 the smallest continuous interval containing all samples from Q with label -1. Similarly g maps to $+1$ the smallest continuous interval containing all samples from Q with label $+1$. The intervals are disjoined by construction. Unmapped points are mapped to $\perp$. Next, $\mathbf{L}'_n$ simulates $\mathbf{L}_n$ with K samples and gets a classifier f that with probability δ satisfies the assumption of the lemma. Finally, it returns

$$f'(x) := \begin{cases} g(x), & \text{if } g(x) \neq \perp \\ f(x), & \text{otherwise.} \end{cases}$$

Consider 4 arcs defined as the 2 arcs constituting $B_n^w(2\pi\eta)$ divided into 2 parts each by the line $\{x \in \mathbb{R}^2 \mid \langle w, x\rangle = 0\}$. Let E be the event that some of these intervals do not contain a sample from Q. Observe that

$$\mathbb{P}[E] \leq 4(1-\eta)^{\frac{9}{\eta}} \leq \frac{1}{1000}.$$

By the union bound with probability $\delta - \frac{1}{1000}$, f satisfies equation (5) and E does not happen. By definition of f' this gives the statement of the lemma. □

Theorem 7 (*Transferable Attack for a Cryptography based Learning Task*)**.** *There exists a learning task $\mathbb{L} = \{\mathbb{L}_\lambda\}_\lambda$ and a function class $\mathcal{F} = \{\mathcal{F}_\lambda\}_\lambda$ such that for every $\epsilon : \mathbb{N} \to \mathbb{N}$ where $1/\epsilon(\lambda)$ is lower-bounded by a sufficiently large polynomial and upper-bounded by some polynomial the following holds.*

1. *$\mathbb{L}$ is $\left(\epsilon, \delta = \frac{1}{10}, S = \frac{10^3}{\epsilon^{1.3}}, \mathcal{F}\right)$-learnable.*
2. *$\mathbb{L}$ is **not** $\left(\epsilon, \delta = \frac{1}{10}, S = \frac{1}{\epsilon}, \mathcal{F}\right)$-learnable*
3. *There exists a circuit family $\mathbf{A} = \{\mathbf{A}_\lambda\}_\lambda$ such that*

$$\mathbf{A} \in_1 \textsc{TransfAttack}\left(\begin{matrix}\mathbb{L}, \mathcal{F}, \epsilon(\lambda),\ q(\lambda) = \frac{16}{\epsilon(\lambda)},\ S_{\mathbf{A}}(\lambda) = \frac{10^3}{\epsilon^{1.3}(\lambda)}, \\ S_{\mathbf{B}}(\lambda) = \frac{1}{10^2\epsilon^2(\lambda)},\ c = \frac{9}{10},\ s = \mathit{negl}(\lambda)\end{matrix}\right).$$

Proof. The learning task is based on $\mathbb{L}^\circ = \{\mathbb{L}_n^\circ\}_n$ from Definition 16.

Setting of Parameters for FHE. Observe that by assumption of the lemma $p \leq 1/\epsilon \leq r$, for some polynomial r, and a polynomial p that we will define later. Let FHE be a fully homomorphic encryption scheme from Theorem 6. We will use the scheme for constant leveled circuits $d = O(1)$. Let $s(n, \lambda, d)$ be the polynomial bounding the size of the encryption of inputs of length n with λ security as well as bounding the size of the circuit for homomorphic evaluation, which is guaranteed to exist by Theorem 6. Let $\beta \in (0, 1)$ and p be a polynomial such that

$$s\left(n^{\beta}, \lambda, d\right) \leq (n \cdot p(\lambda))^{0.1}, \tag{6}$$

which exist because s is a polynomial.

We define $n(\lambda) := \lfloor p^{1/\beta}(\lambda) \rfloor$[12] for the length of inputs in the FHE scheme. Observe that for every λ

$$\begin{aligned} s(n(\lambda), \lambda, d) &\leq (p(\lambda) \cdot p(\lambda))^{0.1} && \text{By equation (6)} \\ &\leq \frac{1}{\epsilon(\lambda)^{0.2}} && \text{By } \epsilon(\lambda) \in \left(\frac{1}{r(\lambda)}, \frac{1}{p(\lambda)}\right). \end{aligned} \tag{7}$$

Learning Task. The learning task will be parametrized by λ, i.e. $\mathbb{L} = \{\mathbb{L}_\lambda\}_\lambda$.

Let $\lambda \in \mathbb{N}$. We define $\mathbb{D}_\lambda := \{\mathcal{D}_\lambda^{(\text{pk,sk})}\}_{(\text{pk,sk})}, \mathcal{H}_\lambda := \{h_\lambda^{(\text{pk,sk,w})}\}_{(\text{pk,sk,w})}$ (for $\mathcal{D}_\lambda^{(\text{pk,sk})}$ and $h_\lambda^{(\text{pk,sk,w})}$ to be defined later), where they are indexed by valid public/secret key pairs of the FHE and $w \in \{x \in \mathbb{R}^2 \mid \|x\|_2 = 1\}$. Let $\mathbb{L}_\lambda$ be defined as corresponding to the following process: sample $(\text{pk,sk}, w) \sim \text{FHE.KeyGen}(1^\lambda) \times U(\{x \in \mathbb{R}^2 \mid \|x\|_2 = 1\})$, return $\left(\mathcal{D}_\lambda^{(\text{pk,sk})}, h_\lambda^{(\text{pk,sk,w})}\right)$.

For a valid (pk,sk) pair we define $\mathcal{D}^{(\text{pk,sk})}$ as the result of the following process: $x \sim U(\{0,1\}^{n(\lambda)})$, with probability $\frac{1}{2}$ return $(0, x, \text{pk})$ and with probability $\frac{1}{2}$ return $(1, \text{FHE.Enc}(\text{pk}, x), \text{pk})$, where the first element of the triple describes if the x is encrypted or not. Formally, in the case that the first element of the triple is 0 one needs to add a padding of size $s(n(\lambda), \lambda, d) - n(\lambda)$ so that descriptions have the same size in both cases.[13]

For a valid (pk,sk) pair and $w \in \{x \in \mathbb{R}^2 \mid \|x\|_2 = 1\}$ we define $h^{(\text{pk,sk,w})}((b, x, \text{pk}), y)$ as a result of the following algorithm: if $b = 0$ return $\mathbb{1}_{h_w(x)=y}$, otherwise let $x_{\text{Dec}} \leftarrow \text{FHE.Dec}(\text{sk}, x)$, $y_{\text{Dec}} \leftarrow \text{FHE.Dec}(\text{sk}, y)$ and if $x_{\text{Dec}}, y_{\text{Dec}} \neq \perp$ (decryption is succesful) return $\mathbb{1}_{h_w(x_{\text{Dec}})=y_{\text{Dec}}}$ and return 1 otherwise.

Note 1 ($\Omega(\frac{1}{\epsilon})$*-sample learning lower bound.*)**.** *By construction any learner using K samples for $\mathbb{L}_\lambda$ (for any λ) can be transformed (potentially computationally inefficiently) into a learner using K samples for $\mathbb{L}^\circ_{n(\lambda)}$ (Defnition 16) that returns a classifier of the same error. This, together with a lower bound for learning from Lemma 3 proves point 2 of the lemma.*

Definition of A (Algorithm 1). $\mathbf{A}_\lambda$ draws $N(\lambda)$ samples $Q = \{((b_i, x_i, \text{pk}), y_i)\}_{i \in [N]}$ for $N(\lambda) := \frac{900}{\epsilon(\lambda)}$.

Next, $\mathbf{A}_\lambda$ chooses a subset $Q_{\text{CLEAR}} \subseteq Q$ of samples for which $b_i = 0$. It trains a classifier $f_{w'}(\cdot) := \text{sgn}(\langle w', \cdot \rangle)$ on Q_{CLEAR} by returning any $f_{w'}$ consistent with Q_{CLEAR}. This can be done in time

$$N(\lambda) \cdot n(\lambda) \leq \frac{900}{\epsilon(\lambda)} \cdot p^{1/\beta}(\lambda) \leq \frac{900}{\epsilon^{1.1}(\lambda)} \tag{8}$$

by keeping track of the smallest interval containing all samples in Q_{CLEAR} labeled with $+1$ and then returning any $f_{w'}$ consistent with this interval.

Note 2 ($O(\frac{1}{\epsilon^{1.3}})$*-time learning upper bound.*)**.** *First note that $\mathbf{A}_\lambda$ learns well, i.e., with probability at least* $1 - 2\left(1 - \frac{\epsilon(\lambda)}{100}\right)^{\frac{900}{\epsilon(\lambda)}} \geq 1 - \frac{1}{1000}$,

$$|\measuredangle(w, w')| \leq \frac{2\pi\epsilon(\lambda)}{100} \tag{9}$$

[12]Note that this setting allows to represent points in $\{x \in \mathbb{R}^2 \mid \|x\|_2 = 1\}$ up to $2^{-p^{1/\beta}(\lambda)}$ precision and this precision is better than $\frac{1}{r(\lambda)}$ for every polynomial r for sufficiently large λ. This implies that this precision is enough to allow for learning up to error ϵ, because of the setting $\epsilon(\lambda) \geq \frac{1}{r(\lambda)}$.

[13]Note that the domain of the distributions is not $\{0,1\}^\lambda$, i.e. $\mathcal{X}_\lambda \neq \{0,1\}^\lambda$.

Algorithm 1 TRANSFATTACK$(\mathrm{Ex}(\mathcal{D}_\lambda, h_\lambda), \epsilon, \lambda)$

Input: Access to the example oracle $\mathrm{Ex}(\mathcal{D}_\lambda, h_\lambda)$, where $(\mathcal{D}_\lambda, h_\lambda) \sim \mathbb{L}_\lambda$, error level $\epsilon : \mathbb{N} \to \mathbb{N}$, and the security parameter λ.

$N := 900/\epsilon(\lambda), q := 16/\epsilon(\lambda)$

$Q = \{((b_i, x_i, \mathrm{pk}), y_i)\}_{i\in[N]} \sim (\mathcal{D}_\lambda)^{N(\lambda)}$ ▷ $N(\lambda)$ i.i.d. samples from $\mathcal{D}_\lambda$

$Q_{\text{CLEAR}} = \{((b, x, \mathrm{pk}), y) \in Q : b = 0\}$ ▷ $Q_{\text{CLEAR}} \subseteq Q$ of unencrypted x's

$f_{w'}(\cdot) := \mathrm{sgn}(\langle w', \cdot\rangle) \leftarrow$ a line consistent with samples from Q_{CLEAR} ▷ $f_{w'} : \mathcal{X}_n \to \{-1, +1\}$

$\{x'_i\}_{i\in[q(\lambda)]} \sim U\left(\left(\mathcal{X}_{n(\lambda)}\right)^{q(\lambda)}\right)$

$S \sim U(2^{[q(\lambda)]})$ ▷ $S \subseteq [q(\lambda)]$ a uniformly random subset

$E_{\text{BND}}; = \emptyset$

for $i \in [q(\lambda) - |S|]$ **do**

  $x_{\text{BND}} \sim U(B^{w'}_{n(\lambda)}(2\pi(\epsilon(\lambda) + \frac{\epsilon(\lambda)}{100})))$ ▷ x_{BND} is close to the decision boundary of $f_{w'}$

  $E_{\text{BND}} := E_{\text{BND}} \cup \{\mathrm{FHE.ENC}(\mathrm{pk}, x_{\text{BND}})\}$

end for

$\mathbf{x} := \{(0, x'_i, \mathrm{pk}) \mid i \in [q(\lambda)] \setminus S\} \cup \{(1, x', \mathrm{pk}) \mid x' \in E_{\text{BND}}\}$

Return $\mathbf{x}$

Moreover, $f_{w'}(x)$ can be implemented by a circuit $C_{f_{w'}}$ that compares x with the endpoints of the interval. This can be done by a constant leveled circuit. Moreover $C_{f_{w'}}$ can be evaluated with FHE.EVAL *in time*

$$\mathit{size}(C_{f_{w'}})s(n(\lambda), \lambda, d) \le 10n \cdot s(n(\lambda), \lambda, d) \le 10p^{1/\beta}(\lambda)s(n(\lambda), \lambda, d) \le \frac{10}{\epsilon^{0.3}(\lambda)},$$

where the last inequality follows from equation (7). This proves point 1 of the lemma.

Next, $\mathbf{A}_\lambda$ prepares $\mathbf{x}$ as follows. It samples $q(\lambda) = \frac{16}{\epsilon(\lambda)}$ points $\{x'_i\}_{i\in[q]}$ from $\{0,1\}^{n(\lambda)}$ uniformly at random. It chooses a uniformly random subset $S \subseteq [q(\lambda)]$. Next, $\mathbf{A}_\lambda$ generates $q(\lambda) - |S|$ inputs using the following process: $x_{\text{BND}} \sim U(B^{w'}_{n(\lambda)}(2\pi(\epsilon(\lambda) + \frac{\epsilon(\lambda)}{100})))$ (x_{BND} is close to the decision boundary of $f_{w'}$), return $\mathrm{FHE.ENC}(\mathrm{pk}, x_{\text{BND}})$. Call the set of $q(\lambda) - |S|$ points E_{BND}. $\mathbf{A}_\lambda$ defines:

$$\mathbf{x} := \{(0, x'_i, \mathrm{pk}) \mid i \in [q] \setminus S\} \cup \{(1, x', \mathrm{pk}) \mid x' \in E_{\text{BND}}\}.$$

The running time of this phase is dominated by evaluations of FHE.EVAL, which takes

$$q(\lambda) \cdot s(n(\lambda), \lambda, d) \le \frac{16}{\epsilon(\lambda)} \cdot \frac{1}{\epsilon^{0.2}(\lambda)} \le \frac{16}{\epsilon^{1.2}(\lambda)}, \tag{10}$$

where the first inequality follows from equation (7). Taking the sum of equation (8) and equation (10) we get that $\mathbf{A}_\lambda$ can be implemented by a circuit of size $\frac{10^3}{\epsilon^{1.3}(\lambda)}$.

$\mathbf{A}_\lambda$ Constitutes a Transferable Attack. Now, consider $\mathbf{B}_\lambda$ of size $S_{\mathbf{B}}(\lambda) = \frac{1}{\epsilon^2(\lambda)}$. By the assumption $S_{\mathbf{B}}(\lambda) \le r(\lambda)$, which implies that the security guarantees of FHE hold for $\mathbf{B}_\lambda$.

We claim that $\mathbf{x}$ is indistinguishable from $\mathcal{D}^{(\mathrm{pk},\mathrm{sk})}_\lambda$ for $\mathbf{B}_\lambda$. Observe that by construction the distribution of ratio of encrypted and not encrypted x's in $\mathbf{x}$ is identical to that of $\mathcal{D}^{(\mathrm{pk},\mathrm{sk})}_\lambda$. Moreover, the distribution of unencrypted x's is identical to that of $\mathcal{D}^{(\mathrm{pk},\mathrm{sk})}_\lambda$ by construction. Finally, by the IND-CPA security[14] of FHE and the fact that the size of $\mathbf{B}_\lambda$ is bounded by some polynomial in λ, $\mathrm{FHE.ENC}(\mathrm{pk}, x_{\text{BND}})$ is distinguishable from $x \sim \mathcal{X}_n, \mathrm{FHE.ENC}(\mathrm{pk}, x)$ with advantage at most $\mathrm{negl}(\lambda)$. Thus *undetectability* holds with near perfect soundness $s = \frac{1}{2} + \mathrm{negl}(\lambda)$.

Next, we claim that $\mathbf{B}_\lambda$ can't return low-error answers on $\mathbf{x}$.

Assume towards contradiction that with probability $\frac{5}{100}$

$$\mathbb{P}_{\substack{w\sim U(\{z\in\mathbb{R}^2 \mid \|z\|_2=1\}),\\ x\sim U(B^w_{n(\lambda)}(2\pi\epsilon(\lambda)))}}[f(x) \ne h_w(x)] \le 10\epsilon(\lambda). \tag{11}$$

[14]Note that we need security of FHE in the nonuniform model of computation.

We can apply Lemma 4 to get that there exists a learner using $\frac{1}{100\epsilon^2(\lambda)} + \frac{9}{\epsilon(\lambda)} \leq \frac{1}{90\epsilon^2(\lambda)}$ samples that with probability $\frac{4}{100}$ returns f' such that

$$\mathbb{P}_{\substack{w\sim U(\{z\in\mathbb{R}^2 \mid \|z\|_2=1\}),\\ x\sim U(\{0,1\}^{n(\lambda)})}}[f'(x) \neq h_w(x)] \leq 40\epsilon^2(\lambda). \tag{12}$$

Applying Lemma 3 to equation (12) we know that

$$40\epsilon^2 \geq \frac{1}{2(\frac{1}{90\epsilon^2(\lambda)})},$$

which is a contradiction. Thus equation (11) does not hold and in consequence, using equation (9), with probability $1 - \frac{6}{100}$

$$\mathbb{P}_{\substack{w\sim U(\{z\in\mathbb{R}^2 \mid \|z\|_2=1\}),\\ x\sim U(B^{w'}_{n(\lambda)}(2\pi(\epsilon(\lambda)+\frac{\epsilon(\lambda)}{10}))}} [f(x) \neq h_w(x)] \geq \frac{10}{14} \cdot 10\epsilon(\lambda) \geq 7\epsilon(\lambda), \tag{13}$$

where crucially x is sampled from $U(B^{w'}_{n(\lambda)})$ and not $U(B^{w}_{n(\lambda)})$. By Fact 1 we know that $|S| \geq \frac{q(\lambda)}{3}$ with probability at least

$$1 - 2e^{-\frac{q(\lambda)}{72}} = 1 - 2e^{-\frac{1}{8\epsilon(\lambda)}} \geq 1 - \frac{1}{1000}.$$

Using the setting of $q(\lambda) = \frac{16}{\epsilon(\lambda)}$ and applying the Chernoff bound and the union bound we get from equation (13) that with probability at least $1 - \frac{1}{10}$ the error $\mathrm{err}(\mathbf{x}, \mathbf{y})$ is larger than $2\epsilon(\lambda)$.

□

Note 3. *We want to emphasize that it is crucial (for our construction) that the distribution has both an encrypted and an unencrypted part.*

As mentioned before, if there was no $\mathcal{D}_{\text{CLEAR}}$ then $\mathbf{A}_\lambda$ would see only samples of the form

$$(\text{FHE.ENC}(x), \text{FHE.ENC}(y))$$

and would not know which of them lie close to the boundary of h_w, and so it would not be able to choose tricky samples. $\mathbf{A}_\lambda$ would be able to learn a low-error classifier, but only *under the encryption. More concretely, $\mathbf{A}_\lambda$ would be able to homomorphically evaluate a circuit that, given a training set and a test point, learns a good classifier and classifies the test point with it. However, it would* not *be able to, with high probability, generate* $\text{FHE.ENC}(x)$*, for x close to the boundary as it would not know (in the clear) where the decision boundary is.*

If there was no $\mathcal{D}_{\text{ENC}}$ then everything would happen in the clear and so $\mathbf{B}$ would be able to distinguish x's that appear too close to the boundary.

Fact 1 (*Chernoff-Hoeffding*)**.** Let $X_1, \ldots, X_k$ be independent Bernoulli variables with parameter p. Then for every $0 < \epsilon < 1$

$$\mathbb{P}\left[\left|\frac{1}{k}\sum_{i=1}^{k} X_i - p\right| > \epsilon\right] \leq 2e^{-\frac{\epsilon^2 k}{2}}$$

and

$$\mathbb{P}\left[\frac{1}{k}\sum_{i=1}^{k} X_i \leq (1-\epsilon)p\right] \leq e^{-\frac{\epsilon^2 kp}{2}}.$$

Also for every $\delta > 0$

$$\mathbb{P}\left[\frac{1}{k}\sum_{i=1}^{k} X_i > (1+\delta)p\right] \leq e^{-\frac{\delta^2 kp}{2+\delta}}$$

G Transferable Attacks Imply Cryptography

G.1 EFID Pairs

The typical way in which security of EFID pairs is defined, e.g., in [Goldreich, 1990], is that they should be secure against all polynomial-time algorithms. However, for the case of pseudorandom generators (PRGs), which are known to be equivalent (in the standard definition) to EFIDs pairs, more granular notions of security were considered. For instance, in [Nisan, 1990] the existence of PRGs secure against adversaries running in time bounded by a fixed, in contrast to all, polynomial, was studied. In a similar spirit, we consider EFID pairs that are secure against adversaries with fixed circuit complexity bounds.

Definition 17 (*Total Variation*)**.** For two distrbutions $\mathcal{D}_0, \mathcal{D}_1$ over a finite domain $\{0,1\}^n$ we define their *total variation distance* as

$$\triangle(\mathcal{D}_0, \mathcal{D}_1) := \sum_{x \in \{0,1\}^n} \frac{1}{2} |\mathcal{D}_0(x) - \mathcal{D}_1(x)|.$$

Definition 18 (*EFID pairs*)**.** For parameters $\eta, \delta : \mathbb{N} \to (0,1)$ and circuit complexity bounds $S, S' : \mathbb{N} \to \mathbb{N}$ we call a pair of ensembles of distributions $(\mathcal{D}^0 = \{\mathcal{D}^0_n\}_n, \mathcal{D}^1 = \{\mathcal{D}^1_n\}_n)$ over domain $\mathcal{X} = \{\mathcal{X}_n\}_n$ an (S, S', η, δ)-EFID pair if for every n

1. The circuit complexity of sampling $\mathcal{D}^0$ and $\mathcal{D}^1$ is at most S,
2. For every n, $\triangle(\mathcal{D}^0_n, \mathcal{D}^1_n) \geq \eta(n)$,
3. For every n, $\mathcal{D}^0_n, \mathcal{D}^1_n$ are $\delta(n)$-indistinguishable for circuits with complexity $S'(n)$.

Observe that Definition 18 is a generalization of the standard definition. Indeed, for every EFID pair $(\mathcal{D}^0, \mathcal{D}^1)$ according to the standard definition there exists an inverse polynomial function η and a polynomial S such that for all polynomials S' there exists a negligible function δ such that $(\mathcal{D}^0, \mathcal{D}^1)$ is an (S, S', η, δ)-EFID pair.

G.2 Transferable Attacks imply EFID pairs

Theorem 8 (*Tasks with Transferable Attacks Imply EFID pairs*)**.** *For every $\epsilon \in (0,1), q \in \mathbb{N}, S_{\mathbf{A}}, S_{\mathbf{B}} : \mathbb{N} \to \mathbb{N}$ such that $S_{\mathbf{A}} \leq S_{\mathbf{B}}$, every learning task $\mathbb{L}$ learnable to error ϵ with confidence p and circuit complexity $S_{\mathbf{A}}$, every $c, s \in (0,1)$ if*

$$\textsc{TransfAttack}\left(\mathbb{L}, \epsilon, q, S_{\mathbf{A}}, S_{\mathbf{B}}, c, s\right)$$

exists with frequency $\frac{1}{3}$ then there exist $S'_{\mathbf{A}}, S'_{\mathbf{B}} : \mathbb{N} \to \mathbb{N}$ that agree with $S_{\mathbf{A}}$ and $S_{\mathbf{B}}$ respectively with frequency $\frac{1}{3}$ and there exists

$$\left(S'_{\mathbf{A}}, S'_{\mathbf{B}}, \frac{1}{2}\left(p + c - 1 - e^{-\frac{\epsilon q}{3}}\right), \frac{s}{2}\right) - \textit{EFID pair.}$$

Proof. Let $\epsilon, S_{\mathbf{A}}, S_{\mathbf{B}}, q, c, s, p, \mathbb{L}$ be as in the assumption of the theorem. Additionally let $\mathbf{A} = \{\mathbf{A}_n\}_n$ be a family of circuits certifying that a Transferable Attack exists with frequency $\frac{1}{3}$ for $\mathbb{L}$.

For every n, define $\mathcal{D}^0_n := \mathcal{D}^q_n$, where we recall that q is the number of samples $\mathbf{A}_n$ sends in the attack. Define $\mathcal{D}^1_n$ to be the distribution of $\mathbf{x} := \mathbf{A}_n$. Note that $\mathbf{x} \in (\mathcal{X}_n)^q$.

Let $a : \mathbb{N} \to \{0,1\}$ be a sequence certifying that a Transferable Attack exists with frequency $\frac{1}{3}$. Let n be such that $a(n) = 1$. Observe that $\mathcal{D}^0_n, \mathcal{D}^1_n$ are samplable with circuit complexity $S_{\mathbf{A}}(n)$ because $\mathbf{A}_n$ complexity is bounded by $S_{\mathbf{A}}(n)$. Secondly, $\mathcal{D}^0_n, \mathcal{D}^1_n$ are $\frac{s}{2}$-indistinguishable for $S_{\mathbf{B}}(n)$-sized adversaries by *undetectability* of $\mathbf{A}_n$. Finally, the fact that $\mathcal{D}^0_n, \mathcal{D}^1_n$ are statistically far follows from *transferability*. Indeed, the following procedure accepting input $\mathbf{x} \in (\{0,1\}^n)^q$ is a distinguisher:

1. Run the learner (the existence of which is guaranteed by the assumption of the theorem) to obtain f.

2. $\mathbf{y} := f(\mathbf{x})$.

3. If $\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon$ return 0, otherwise return 1.

If $\mathbf{x} \sim \mathcal{D}^0 = \mathcal{D}^q$ then $\text{err}(f) \leq \epsilon$ with probability p. By Fact 1 and the union bound we also know that $\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon$ with probability $p - e^{-\frac{\epsilon q}{3}}$ and so, the distinguisher will return 0 with probability $p - e^{-\frac{\epsilon q}{3}}$. On the other hand, if $\mathbf{x} \sim \mathcal{D}^1 = \mathbf{A}$ we know from *transferability* of $\mathbf{A}_n$ that every algorithm running in time $S_{\mathbf{B}}(n)$ will return $\mathbf{y}$ such that $\text{err}(\mathbf{x}, \mathbf{y}) > 2\epsilon$ with probability at least c. By the assumption that $S_{\mathbf{B}}(n) \geq S_{\mathbf{A}}(n)$ we know that $\text{err}(\mathbf{x}, f(\mathbf{x})) > 2\epsilon$ with probability at least c also. Consequently, the distinguisher will return 1 with probability at least c in this case. By the properties of total variation this implies that $\triangle(\mathcal{D}^0_n, \mathcal{D}^1_n) \geq \frac{1}{2}(p + c - 1 - e^{-\frac{\epsilon q}{3}})$.

We define a pair of families of distributions $\widehat{\mathcal{D}}^0, \widehat{\mathcal{D}}^1$ and functions $S'_{\mathbf{A}}, S'_{\mathbf{B}}$ as follows. For every n such that $a(n) = 1$ we define $\widehat{\mathcal{D}}^0_n = \mathcal{D}^0_n, \widehat{\mathcal{D}}^1 = \mathcal{D}^1_n, S'_{\mathbf{A}}(n) = S_{\mathbf{A}}(n), S'_{\mathbf{B}}(n) = S_{\mathbf{B}}(n)$. For every n sich that $a(n) = 0$ we define $\widehat{\mathcal{D}}^0_n = \mathcal{D}^0_k$ for the smallest $k > n$ such that $a(k) = 1$, and $S'_{\mathbf{A}}(n) = S_{\mathbf{A}}(k)$ And analogously for $\widehat{\mathcal{D}}^1_n$ and $S'_{\mathbf{B}}$.

Simple verification yields that $\widehat{\mathcal{D}}^0_n, \widehat{\mathcal{D}}^1_n$ is an $(S'_{\mathbf{A}}, S'_{\mathbf{B}}, \frac{1}{2}(p + c - 1 - e^{-\frac{\epsilon q}{3}}), \frac{s}{2})$-EFID pair.

Note 4 (*Setting of parameters*)**.** *Observe that if $p \approx 1$, i.e., it is possible to almost surely learn f in time $S_{\mathbf{A}}$ such that err$(f) \leq \epsilon$, c is a constant, $q = \Omega(\frac{1}{\epsilon})$ then η in the parameters for the EFID is a constant and so $\triangle(\mathcal{D}^0, \mathcal{D}^1)$ is a constant.*

Note 5. *We want to emphasize that our distinguisher crucially uses the error oracle in its last step. So it is possible that it is not implementable for all circuit complexity bounds!*

□

H Adversarial Defenses exist

Our result is based on [Goldwasser et al., 2020]. Before we state and prove our result we give an overview of the learning model considered in [Goldwasser et al., 2020]. The authors give a defense against *arbitrary examples* in a transductive model with rejections. In contrast, our model does not allow rejections, but we do require indistinguishability.

H.1 Transductive Learning With Rejections.

In [Goldwasser et al., 2020] the authors consider a model, where a learner $\mathbf{L}$ receives a training set of labeled samples from the original distribution $(\mathbf{x}_{\mathcal{D}}, \mathbf{y}_{\mathcal{D}} = h(\mathbf{x}_{\mathcal{D}})), \mathbf{x} \sim \mathcal{D}^N, \mathbf{y}_{\mathcal{D}} \in \{-1, +1\}^N$, where h is the ground truth, together with a test set $\mathbf{x}_T \in (\{0,1\}^n)^q$. Next, $\mathbf{L}$ uses $(\mathbf{x}_{\mathcal{D}}, \mathbf{y}_{\mathcal{D}}, \mathbf{x}_T)$ to compute $\mathbf{y}_T \in \{-1, +1, \square\}^q$, where $\square$ represents that $\mathbf{L}$ abstains (rejects) from classifying the corresponding x.

Before we define when learning is successful, we will need some notation. For $q \in \mathbb{N}, \mathbf{x} \in (\{0,1\}^n)^q, \mathbf{y} \in \{-1, +1, \square\}^q$ we define

$$\text{err}(\mathbf{x}, \mathbf{y}) := \frac{1}{q} \sum_{i \in [q]} \mathbb{1}_{\left\{h(x_i) \neq y_i, y_i \neq \square, h(x_i) \neq \perp\right\}}, \quad \square(\mathbf{y}) := \frac{1}{q} \left| \left\{ i \in [q] : y_i = \square \right\} \right|,$$

which means that we count $(x, y) \in \{0,1\}^n \times \{-1, +1, \square\}$ as an error if h is well defined on x, y is not an abstantion and $h(x) \neq y$.

Learning is successful if it satisfies two properties.

- If $\mathbf{x}_T \sim \mathcal{D}^q$ then with high probability $\text{err}(\mathbf{x}_T, \mathbf{y}_T)$ and $\square(\mathbf{y}_T)$ are small.
- For *every* $\mathbf{x}_T \in (\{0,1\}^n)^q$ with high probability $\text{err}(\mathbf{x}_T, \mathbf{y}_T)$ is small.[15]

[15] Note that, crucially, in this case $\square(\mathbf{y}_T)$ might be very high, e.g., equal to 1.

The formal guarantee of a result from Goldwasser et al. [2020] are given in Theorem 9. Let's call this model Transductive Learning with Rejections (TLR).

Note the differences between TLR and our definition of Adversarial Defenses. To compare the two models we associate the learner $\mathbf{L}$ from TLR with $\mathbf{B}$ in our setup, and the party producing $\mathbf{x}_T$ with $\mathbf{A}$ in our definition. First, in TLR, $\mathbf{B}$ does not send f to $\mathbf{A}$. Secondly, and most importantly, we do not allow $\mathbf{B}$ to reply with rejections ($\square$) but instead require that $\mathbf{B}$ can "distinguish" that it is being tested (see soundness of Definition 9). Finally, there are no apriori time bounds on either $\mathbf{A}$ or $\mathbf{B}$ in TLR. The models are similar but a priori incomparable and any result for TLR needs to be carefully analyzed before being used to prove that it is an Adversarial Defense.

H.2 Formal guarantee for Transductive Learning with Rejections (TLR)

Theorem 5.3 from Goldwasser et al. [2020] adapted to our notation reads.

Theorem 9 (*TLR guarantee (Goldwasser et al. [2020])*)**.** *For any $N \in \mathbb{N}, \epsilon \in (0,1), h \in \mathcal{H}$ and distribution $\mathcal{D}$ over $\{0,1\}^n$:*

$$\mathbb{P}_{\mathbf{x}_{\mathcal{D}}, \mathbf{x}'_{\mathcal{D}} \sim \mathcal{D}^N}\left[\forall\, \mathbf{x}_T \in \{0,1\}^{nN} : \mathit{err}(\mathbf{x}_T, f(\mathbf{x}_T)) \leq \epsilon^* \wedge \square\left(f\left(\mathbf{x}'_{\mathcal{D}}\right)\right) \leq \epsilon^*\right] \geq 1-\epsilon,$$

where $\epsilon^* = \sqrt{\frac{2d}{N}\log(2N) + \frac{1}{N}\log\left(\frac{1}{\epsilon}\right)}$ and $f = \textsc{Rejectron}(\mathbf{x}_{\mathcal{D}}, h(\mathbf{x}_{\mathcal{D}}), \mathbf{x}_T, \epsilon^*)$, where $f : \{0,1\}^n \to \{-1,+1,\square\}$ and d denotes the VC-dimension on $\mathcal{H}$. Rejectron *is defined in Figure 2. in [Goldwasser et al., 2020].*

Rejectron is an algorithm that accepts a labeled training set $(\mathbf{x}_{\mathcal{D}}, h(\mathbf{x}_{\mathcal{D}}))$ and a test set $\mathbf{x}_T$ and returns a classifier f, which might reject some inputs. The learning is successful if with a high probability f rejects a small fraction of $\mathcal{D}^N$ and for every $\mathbf{x}_T \in \{0,1\}^{nN}$ the error on labeled x's in $\mathbf{x}_T$ is small.

H.3 Adversarial Defense for bounded VC-dimension

We are ready to state the main result of this section.

Lemma 5 (*Adversarial Defense for bounded VC-dimension*)**.** *Let $\{\mathcal{H}_n\}_n$ be a family of hypothesis classes such that there exists a polynomial p such that for every n, $\mathcal{H}_n$ has a VC-dimension bounded by $p(n)$. There exists a family of circuits $\mathbf{B} = \{\mathbf{B}_n\}_n$ such that for every $\mathbb{L}$ satisfying for every n that the support of the marginal of $\mathbb{L}_n$ is contained in $\mathcal{H}_n$, i.e., the ground truth sampled from $\mathbb{L}$ are always in $\mathcal{H}$, such that for every sufficiently small ϵ,*

$$\mathbf{B} \in_1 \textsc{Defense}\left(\mathbb{L}, \epsilon, q = \frac{\mathtt{poly}(n)}{\epsilon^3}, S_{\mathbf{A}} = \infty, S_{\mathbf{B}} = \mathtt{poly}\left(\frac{n}{\epsilon}\right), l = 1-\epsilon, c = 1-\epsilon, s = \epsilon\right).$$

Note that, by the PAC learning bound, this is a setting of parameters, where $\mathbf{B}$ has enough time to learn a classifier of error ϵ. By slightly abusing the notation, we write $S_{\mathbf{A}} = \infty$, meaning that the defense is secure against *all* adversaries regardless of their running time.

Proof. The proof is based on an algorithm from Goldwasser et al. [2020].

Construction of B. Let $\epsilon \in (0,1), n \in N$, $d(n)$ be the VC-dimension of $\mathcal{H}_n$ and

$$N := \frac{d\log^2(d)}{\epsilon^3}.$$

Let $q := N$. First, $\mathbf{B}$, draws N labeled samples $(\mathbf{x}_{\text{FRESH}}, h(\mathbf{x}_{\text{FRESH}}))$. Next, it finds $f \in \mathcal{H}$ consistent with them and sends f to $\mathbf{A}$. Importantly this computation is the same as the first step of Rejectron.

Next, $\mathbf{B}$ receives as input $\mathbf{x} \in (\{0,1\}^n)^q$ from $\mathbf{A}$. $\mathbf{B}$. Let $\epsilon^* := \sqrt{\frac{2d}{N}\log(2N) + \frac{1}{N}\log\left(\frac{1}{\epsilon}\right)}$. Next $\mathbf{B}$ runs $f' = \textsc{Rejectron}(\mathbf{x}_{\text{FRESH}}, h(\mathbf{x}_{\text{FRESH}}), \mathbf{x}, \epsilon^*)$, where Rejectron is starting from the second step of the algorithm (Figure 2 [Goldwasser et al., 2020]). Importantly, for every $x \in \{0,1\}^n$, if $f'(x) \neq \square$ then $f(x) = f'(x)$. In words, f' is equal to f everywhere where f' does not reject.

Finally $\mathbf{B}$ returns 1 if $\square(f'(\mathbf{x})) > \frac{2}{3}\epsilon$, and returns 0 otherwise.

$\mathbf{B}$ is a Defense. First, by the standard PAC theorem, with probability at least $1-\epsilon$, $\text{err}(f) \leq \frac{\epsilon}{2}$. This means that *correctness* holds with probability $l = 1-\epsilon$.

Note that with our setting of N,

$$\epsilon^* \leq \frac{\epsilon}{2}.$$

Theorem 9 guarantees that

- if $\mathbf{x} \in \mathcal{D}^q$ then with probability at least $1-\epsilon$,

  $$\square(f'(\mathbf{x})) \leq \frac{\epsilon}{2}.$$

  which in turn implies that with the same probability $\mathbf{B}$ returns $b = 0$. This implies that *completeness* holds with probability $1-\epsilon$.

- for every $\mathbf{x} \in (\{0,1\}^n)^q$ with probability at least $1-\epsilon$,

  $$\text{err}(\mathbf{x}, f'(\mathbf{x})) \leq \frac{\epsilon}{2}.$$

  To compute soundness we want to upper bound the probability that $\text{err}(\mathbf{x}, f(\mathbf{x})) > 2\epsilon$[16] and $b = 0$. By construction of $\mathbf{B}$ if $b = 0$ then $\square(f'(\mathbf{x})) \leq \frac{2\epsilon}{3}$, which means that with probability at least $1-\epsilon$

  $$\text{err}(\mathbf{x}, \mathbf{y}) \leq \frac{2\epsilon}{3} + \frac{\epsilon}{2} < 2\epsilon \text{ or } b = 1.$$

  This translates to *soundness* holding with $s = \epsilon$.

REJECTRON can be implemented by a circuit of size polynomial in N and makes $O(\frac{1}{\epsilon})$ calls to an Empirical Risk Minimizer on $\mathcal{H}$ (that we assume can be implemented by a circuit of size polynomial in d), which implies the promised circuit complexity. □

I Watermarks exist

Lemma 6 (*Watermark for bounded VC-dimension against fast adversaries*)**.** *There exists a family of hypothesis classes $\{\mathcal{H}_d\}_d$ such that for every d, $\mathcal{H}_d$ has VC-dimension d and a family of distributions $\{\mathcal{D}_d\}_d$ such that for every $\epsilon \in \left(\frac{10000}{d}, \frac{1}{8}\right)$ there exists a family of circuits $\mathbf{A} = \{\mathbf{A}_d\}_d$ and a family of function classes $\mathcal{F}$ for which the following conditions hold. For every learning $\mathbb{L} = \{\mathbb{L}_d\}_d$ that for every d samples $\mathcal{D}_d$ always and $h_d \in \mathcal{H}_d$,*

$$\mathbf{A} \in_1 \text{WATERMARK}\begin{pmatrix}\mathbb{L},\ \mathcal{F},\ \epsilon,\ q = O\left(\frac{1}{\epsilon}\right),\ S_{\mathbf{A}} = O\left(\frac{d}{\epsilon}\right), \\ S_{\mathbf{B}} = \frac{d}{100},\ l = 1 - \frac{1}{100},\ c = 1 - \frac{2}{100},\ s = \frac{56}{100}\end{pmatrix}.$$

Note that the setting of parameters is such that $\mathbf{A}$ can learn (with high probability) a classifier of error ϵ, but $\mathbf{B}$ is *not* able to learn a low-error classifier within its allotted circuit size $S_{\mathbf{B}}$. This contrasts with Lemma 5, where $\mathbf{B}$ has a sufficiently large circuit size to learn. This is the regime of interest for Watermarks, where the scheme is expected to be secure against $\mathbf{B}$ with limited circuit complexity.

Proof. Let $\mathcal{D}$ be the uniform distribution over $[N]$ for $N = 100d^2$, where recall that $[N] = \{1, \ldots, N\}$. Let $\mathcal{H}$ be the concept class of functions that have exactly d $+1$'s in $[N]$. Note that $\mathcal{H}$ has VC-dimension d. Let $h \in \mathcal{H}$ be the ground truth.

[16] Note that we measure the error of f not f'.

Construction of A. $\mathbf{A}$ works as follows. It draws $n = O\left(\frac{d}{\epsilon}\right)$ samples from $\mathcal{D}$ labeled with h. Let's call them $\mathbf{x}_{\text{TRAIN}}$. Let

$$A := \{x \in [N] : \mathbf{x}_{\text{TRAIN}}, h(x) = +1\}, B := \{x \in [N] : x \in \mathbf{x}_{\text{TRAIN}}, h(x) = -1\}.$$

$\mathbf{A}$ takes a uniformly random subset $A_w \subseteq A$ of size q. It defines sets

$$A' := A \setminus A_w,\ B' := B \cup A_w.$$

$\mathbf{A}$ computes f consistent with the training set $\{(x, +1) : x \in A'\} \cup \{(x, -1) : x \in B'\}$. $\mathbf{A}$ samples $S \sim \mathcal{D}^q$. It defines the watermark to be $\mathbf{x} := A_w$ with probability $\frac{1}{2}$ and $\mathbf{x} := S$ with probability $\frac{1}{2}$.

$\mathbf{A}$ sends $(f, \mathbf{x})$ to $\mathbf{B}$. $\mathbf{A}$ can be implemented with circuit complexity $O\left(\frac{d}{\epsilon}\right)$.

A is a Watermark. We claim that $(f, \mathbf{x})$ constitutes a watermark.

It is possible to construct a watermark of prescribed size, i.e., find a subset A_w of a given size, only if $|A| \geq q$. The probability that a single sample from $\mathcal{D}$ is labeled $+1$ is $\frac{d}{N}$, so by the Chernoff bound (Fact 1) $|A|, |B| > \frac{dn}{2N} \geq q$ with probability $1 - \frac{1}{100}$, where we used that $n = O\left(\frac{d}{\epsilon}\right), N = 100d^2, q = O(\frac{1}{\epsilon})$.

Correctness. Let $h'(x) := h(x)$ if $x \in [N] \setminus A_w$ and $h'(x) := -h(x)$ otherwise. Note that h' has exactly $d - q$ $+1$'s in $[N]$. By construction, f is a classifier consistent with h'. By the PAC theorem we know that with probability $1 - \frac{1}{100}$, f has an error at most ϵ wrt to h' (because the hypothesis class of functions with *at most* d $+1$'s has a VC dimension of $O(d)$). h' differs from h on q points, so

$$\text{err}(f) \leq \epsilon + q/N = O\left(\epsilon + \frac{1}{\epsilon d^2}\right) = O(\epsilon). \tag{14}$$

with probability $1 - \frac{1}{100}$, which implies that *correctness* is satisfied with $l = 1 - \frac{1}{100}$.

Distinguishing of $\mathbf{x}$ and $\mathcal{D}^q$. Note that the distribution of A_w is the same as the distribution of a uniformly random subset of $[N]$ of size q (when taking into account the randomness of the choice of $h \sim U(\mathcal{H})$). Observe that the probability that drawing q i.i.d. samples from $U([N])$ we encounter repetitions is at most

$$\frac{1}{N} + \frac{2}{N} + \cdots + \frac{q}{N} \leq \frac{3q^2}{N} \leq \frac{1}{100},$$

because $q < \frac{d}{100} < \frac{\sqrt{N}}{10}$. This means that $\frac{1}{100}$ is an information-theoretic upper bound on the distinguishing advantage between $\mathbf{x} = A_w$ and $\mathcal{D}^q$.

Moreover, $\mathbf{B}$ has access to at most t samples and the probability that the set of samples $\mathbf{B}$ draws from $\mathcal{D}^t$ and A_w have empty intersection is at least $1 - \frac{1}{100}$. It is because it is at least $(1 - \frac{t}{N})^t \geq (1 - \frac{1}{\sqrt{N}})^{\sqrt{N}/10} \geq 1 - \frac{1}{100}$, where we used that $t < \frac{\sqrt{N}}{10}$.[17]

Note that by construction f maps all elements of A_w to -1. The probability over the choice of $F \sim \mathcal{D}^q$ that $F \subseteq h^{-1}(\{-1\})$, i.e., all elements of F have true label -1, is at least

$$\left(1 - \frac{d}{N}\right)^q \geq 1 - \frac{1}{100}.$$

The three above observations and the union bound imply that the distinguishing advantage for distinguishing $\mathbf{x}$ from $\mathcal{D}^q$ of $\mathbf{B}$ is at most $\frac{4}{100}$ and so the *undetectability* holds with $s = \frac{8}{100}$.

Unremovability. Assume, towards contradiction with *unremovability*, that $\mathbf{B}$ can find $\mathbf{y}$ that with probability $s' = \frac{1}{2} + \frac{6}{100}$ satisfies $\text{err}(\mathbf{x}, \mathbf{y}) \leq 2\epsilon$. Notice, that $\text{err}(A_w, f(A_w)) = 1$ by construction.

Consider an algorithm $\mathcal{A}$ for distinguishing A_w from $\mathcal{D}^q$. Upon receiving $(f, \mathbf{x})$ it first runs $\mathbf{y} = \mathbf{B}(f, \mathbf{x})$ and returns 1 iff $d(\mathbf{y}, f(\mathbf{x})) \geq \frac{q}{2}$. We know that the distinguishing advantage is at most $\frac{1}{2} + \frac{4}{100}$, so

$$\frac{1}{2}\mathbb{P}_{\mathbf{x}:=A_w}[\mathcal{A}(f, \mathbf{x}) = 1] + \frac{1}{2}\mathbb{P}_{\mathbf{x}\sim\mathcal{D}^q}[\mathcal{A}(f, \mathbf{x}) = 0] \leq \frac{1}{2} + \frac{4}{100}.$$

[17] If the sets were not disjoint then $\mathbf{B}$ could see it as suspicious because f makes mistakes on all of A_w.

But also note that

$$\begin{aligned}
s' &\le \mathbb{P}_{\mathbf{x}\sim\mathbf{A}}[\text{err}(\mathbf{x},\mathbf{y}) \le 2\epsilon] \\
&\le \frac{1}{2}\mathbb{P}_{\mathbf{x}:=A_w}[d(\mathbf{y}, f(\mathbf{x})) \ge (1-2\epsilon)q] + \frac{1}{2}\mathbb{P}_{\mathbf{x}\sim\mathcal{D}^q}[d(\mathbf{y}, f(\mathbf{x})) \le (2\epsilon + \text{err}(f))q] \\
&\le \frac{1}{2}\mathbb{P}_{\mathbf{x}:=A_w}[d(\mathbf{y}, f(\mathbf{x})) \ge q/2] + \frac{1}{2}\mathbb{P}_{\mathbf{x}\sim\mathcal{D}^q}[d(\mathbf{y}, f(\mathbf{x})) \le q/2] + \frac{1}{100} \\
&\le \frac{1}{2}\mathbb{P}_{\mathbf{x}:=A_w}[\mathcal{A}(f,\mathbf{x}) = 1] + \frac{1}{2}\mathbb{P}_{\mathbf{x}\sim\mathcal{D}^q}[\mathcal{A}(f,\mathbf{x}) = 0] + \frac{1}{100}.
\end{aligned}$$

Combining the two above equations we get a contradiction and thus the *unremovability* holds with $s' = \frac{1}{2} + \frac{6}{100}$.

Uniqueness. The following $\mathbf{B}$ certifies *uniqueness*. It draws $O\left(\frac{d}{\epsilon}\right)$ samples from $\mathcal{D}$, let's call them $\mathbf{x}'_{\text{TRAIN}}$ and trains f' consistent with it. By the PAC theorem $\text{err}(f') \le \epsilon$ with probability at least $1 - \frac{1}{100}$. Next upon receiving $\mathbf{x} \in \{0,1\}^{nq} = [N]^q$ it returns $y = f'(\mathbf{x})$. By the fact that $\mathbf{x}$ is a random subset of $[N]$ of size q by the Chernoff bound, the union bound we know that $\text{err}(\mathbf{x},\mathbf{y}) = \text{err}(\mathbf{x}, f'(\mathbf{x})) \le 2\epsilon$ with probability at least $1 - \frac{2}{100}$ over the choice of h. This proves *uniqueness*. □

J Future Directions

Below we provide some interesting technical and conceptual future directions.

J.1 Alternate Viewpoint for Task Complexity

Here we briefly note a connection to the **Platonic Representation Hypothesis** (PRH) Huh et al. [2024], which posits that as models grow in capacity, their learned representations become increasingly similar—hence properties (and failures) transfer more readily across models. Theorem 3 in our work shows that transferable attacks arise only when the underlying learning task is computationally hard (in the EFID sense). If PRH's convergence of representations holds, one should indeed expect greater transferability of adversarial examples; our result then suggests that such transferability is an indicator of *computational complexity* of the task. In this view, a (plausibly) necessary condition—and a partial explanation—of PRH is that frontier models are solving **increasingly difficult** problems, which in turn induces representation similarities and transfer. Making these connections precise could be a promising direction for future work.

J.2 Practical Aspects of our Main Theorem

At a fundamental level, families of Boolean circuits are Turing complete, meaning they can simulate any algorithm that a Turing machine can. This makes them expressive enough to capture any computable algorithm and a natural abstraction for studying the inherent properties of learning tasks, independent of specific algorithms. The existence result can guide practical efforts by informing where to search. A key part of our proof is the formulation of a zero-sum game between a "watermarking agent" and a "defense agent," where the game's value indicates which of the three properties exists. Moreover, an optimal strategy in this game corresponds to an actual implementation of a watermark, defense, or transferable attack.

Though finding a Nash equilibrium in such a game seems computationally challenging—since the action spaces involve all circuits of a given size—recent iterative algorithms for large-scale games Lanctot et al. [2017], McAleer et al. [2021], Adam et al. [2021] offer promising approaches. These methods work by evaluating only parts of the game at each step, discovering good strategies over time. Recall that our model captures examples like Pal and Vidal [2020].

In practical settings, circuits can be replaced with standard ML models (e.g., deep neural networks). This opens the door to algorithms that (1) determine whether a given task admits a defense, watermark, or transferable attack, and (2) produce an implementation accordingly. In summary, our theory shows that for a given task, one only needs to set up the appropriate loss and apply these iterative algorithms to extract the desired property. We believe this represents an exciting future direction at the intersection of large-scale algorithmic game theory and AI security.

J.3 Beyond Classification

Inspired by Theorem 2, we conjecture a possibility of generalizing our results to generative learning tasks. Instead of a ground truth function, one could consider a ground truth quality oracle Q, which measures the quality of every input and output pair. This model introduces new phenomena *not* present in the case of classification. For example, the task of *generation*, i.e., producing a high-quality output y on input x, is decoupled from the task of *verification*, i.e., evaluating the quality of y as output for x. By decoupled, we mean that there is no clear formal reduction from one task to the other. Conversely, for classification, where the space of possible outputs is small, the two tasks are equivalent. Without going into details, this decoupling is the reason why the proof of Theorem 1 does not automatically transfer to the generative case.

This decoupling introduces new complexities, but it also suggests that considering new definitions may be beneficial. For example, because generation and verification are equivalent for classification tasks, we allowed neither $\mathbf{A}$ nor $\mathbf{B}$ access to h, as it would trivialize the definitions. However, a modification of the Definition 8 (Watermark), where access to Q is given to $\mathbf{B}$ could be investigated in the generative case. Interestingly, such a setting was considered in [Zhang et al., 2023], where access to Q was crucial for mounting a provable attack on "all" strong watermarks. As we alluded to earlier, Theorem 2 can be seen as an example of a task, where generation is easy but verification is hard – the opposite to what Zhang et al. [2023] posits. Furthermore, recent work Gluch and Goldwasser [2025] proved that, in the context of adversarial robustness and safety, there exist generative learning tasks for which *verification* (or *detection*, i.e., identifying adversarial examples) is strictly harder than *generation* (or *mitigation*, i.e., allocating additional inference time to correct outputs of the base model).

We hope that careful formalizations of the interaction and capabilities of all parties might give insights into not only the schemes considered in this work, but also problems like weak-to-strong generalization [Burns et al., 2024] or scalable oversight [Brown-Cohen et al., 2023].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Claims are based on the theoretical derivations in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Our paper focuses on classification tasks, and in Appendix J we discuss the challenges of extending these ideas to generative learning tasks.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See preliminaries, technical overview, and the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [NA]

   Justification: No experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: No experiments.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: No experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: No experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: No experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Guidelines were read and paper does conform.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Discussion of broader impact is in introduction, related literature and in the reflections at the end of the paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Citations are provided throughout the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets are released.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs were used to edit the text.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.