# IS PREVENTING MODEL DISTILLATION POSSIBLE? A THEORETICAL PERSPECTIVE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Proprietary AI models are publicly accessible via an API. Third parties can use this limited black-box access to help develop their own AI models by extracting training data. This practice is known as *distillation* (aka *model extraction*) and measures to prevent it are known as *anti-distillation*. Building on prior theoretical formalisms of passive observational model extraction defenses, we study active output-perturbation and stateful anti-distillation defenses in the language of learning theory and ask: *when is it possible to provably prevent distillation of a model?* We show that the feasibility of anti-distillation depends critically on the computational hardness of the underlying agnostic PAC learning problem, the quantifier ordering between the server/distribution and the adversary, and the statefulness of the server. For memoryless servers, we prove an equivalence between universal distillation attacks and distribution-free agnostic PAC learning, whereas for stateful servers, we show that provable fixed-server average-case defenses exist for some model classes and natural query distributions, while undetectable attacks exist for others.

## 1 INTRODUCTION

Frontier artificial intelligence (AI) models – such as those from OpenAI, Anthropic, and Google – are served to the general public exclusively through Application Programming Interfaces (APIs), without releasing the model weights or source code. This service-based approach allows developers to detect and prevent misuse of their models and also to recoup the costs of model development. However, exposing the input-output behavior of these models introduces a profound vulnerability: the extraction of the model's underlying knowledge and capabilities by using the API to generate synthetic training data. This practice, widely known as "knowledge distillation" (Hinton et al., 2015) or "model extraction" (Tramèr et al., 2016), has evolved into a standard methodology for training smaller, open-weights models (Taori et al., 2023; Chiang et al., 2023; Xu et al., 2023). While distillation democratizes access to high-performing AI models and accelerates open-source research, it simultaneously threatens the commercial viability of proprietary model developers. Perhaps more importantly, as AI models become more potent, their proliferation becomes dangerous – bad actors can use distillation to develop state-of-the-art models that are deployed without safeguards and which cannot be effectively regulated. That is, distillation poses AI safety risks.

In response, API providers have sought to implement "anti-distillation" measures. Currently, these primarily take the form of legal deterrents, such as explicit Terms of Service agreements forbidding the use of API outputs to train competing models (OpenAI, 2026; Anthropic, 2025; Google, 2026). However, because these agreements may in some cases be difficult to put into practice, the research community has begun exploring technical countermeasures. These range from output watermarking (Kirchenbauer et al., 2023; Sander et al., 2024) designed to detect AI-generated training data post-hoc, to active perturbation strategies that attempt to degrade the utility of the outputs for gradient descent while preserving subjective quality for human readers (Orekondy et al., 2020; Savani et al., 2025; Li et al., 2025; Fang et al., 2026). However, these technical countermeasures alone may not be effective in the face of determined distillation efforts (Anthropic, 2026; Zhang et al., 2026; Kratsios, 2026). We expect attackers and defenders to continually increase the sophistication of their attacks and defenses.

On the theoretical front, Karchmer (2023) (building on (Canetti & Karchmer, 2021)) initiated the formal study of model extraction defenses by introducing *Observational Model Extraction Defenses (OMEDs)*—detectors that inspect a single batch of queries and either return exact, noiseless labels or



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reject the batch—and showing strong impossibility results when the target class is PAC learnable. Yet, as noted above, modern anti-distillation proposals increasingly go beyond passive batch rejection by *actively injecting bounded noise* into responses and maintaining *state* across queries. This motivates our central theoretical question: *Can active output perturbation and server statefulness provably prevent model distillation, and what are the exact complexity-theoretic barriers governing attacks and defenses?*

## 2 Our Contributions

In this paper, we study the (im)possibility of preventing model distillation from a theoretical perspective. Extending the passive observational framework of Karchmer (2023), we formalize interactive, noisy anti-distillation as a multi-round game between an API provider (the server/defender) and a querying third party (the client/attacker). By grounding this interaction in learning theory, we map the theoretical boundaries of model protection across different quantifier regimes and server capabilities.

**Formalization and Quantifier Hierarchy (§3).** We formulate the multi-round *Distillation Game* (Definition 3.1) between a stateful, possibly noisy server $\mathcal{S}$ holding a (possibly randomized) model $h \in \mathcal{H}$ and an adaptive adversary $\mathcal{A}$. The server must balance two competing requirements with respect to a loss function $\mathrm{L}$ and honest query distribution $\mathcal{D}$:

1. **$\alpha$-Accuracy (Definition 3.2):** Queries drawn i.i.d. from $\mathcal{D}$ by an honest client receive responses whose conditional distribution is within average loss $\alpha$ of $h$'s true outputs.
2. **$\gamma$-Anti-distillation security (Definitions 3.3–3.4):** Any computationally bounded adversary $\mathcal{A}$ outputs a hypothesis $\hat{h}$ with expected population loss $\mathop{\mathbb{E}}_{x \leftarrow \mathcal{D}} \left[ \mathrm{L}\left( \hat{h}(x), h(x) \right) \right] \geq \gamma$.

Crucially, we distinguish two regimes based on whether the defender or the attacker moves first. In *Fixed-Server Security* (Definition 3.3), the defender moves first: a single server $\mathcal{S}$ on a fixed distribution $\mathcal{D}$ must defeat all efficient adversaries $\mathcal{A}$ (either for a worst-case model $h \in \mathcal{H}$ or on average over $h \leftarrow \mu$). In *Distribution-Free (i.e. "Universal") Security* (Definition 3.4), the attacker moves first: for each adversary $\mathcal{A}$, there exist a distribution $\mathcal{D}$, a server $\mathcal{S}$, and a model $h$ on which $\mathcal{A}$ fails.

**Information-Theoretic Baseline (§4).** We show that any $\alpha$-accurate server (even if stateful) is universally $\gamma$-distillation insecure against computationally unbounded adversaries for $\gamma = 2(\alpha + \sqrt{\log|\mathcal{H}|/n})$ using $n = O(\log|\mathcal{H}|/\alpha^2)$ honest i.i.d. queries (Theorem 4.1). Thus, any effective anti-distillation defense *must* rely on the computational hardness of learning.

**Equivalence of Memoryless Distillation and Agnostic PAC Learning (§5).** For memoryless servers (which answer each query independently), we prove that universal distillation is computationally equivalent to (distribution-free) Gap PAC learning (Theorems 5.2 and 5.3). By contraposition, the hardness of Gap PAC learning implies *Distribution-Free Anti-Distillation Security* (Corollary 5.4).

**Positive Example: Fixed-Server Defense via Stateful Caching and LPN (§6).** Distribution-free PAC hardness in Section 5 only yields an adversary-dependent defense ($\forall \mathcal{A} \exists \mathcal{D}, \mathcal{S}$). In Section 6, we construct a *single fixed server* that achieves security on a natural distribution ($\exists \mathcal{D}, \mathcal{S} \forall \mathcal{A}$). For hashed $d$-bit parities under the uniform distribution $\mathcal{D}$, our stateful server $\mathcal{S}_p$ is $o(1)$-accurate and achieves *Fixed-Server Average-Case* $(1/2 - \mathrm{negl}(d))$*-Security* over $h \leftarrow \mathrm{Uniform}(\mathcal{H})$ against all $\mathrm{poly}(d)$-time adversaries in the random oracle model under LPN (Theorem 6.3). By contrast, the noiseless class is easily learned via Gaussian elimination (Proposition 6.2).

**Negative Example: An Undetectable Attack (§7).** For standard parities under the uniform distribution $\mathcal{D}$ on $\{0,1\}^d$, we show that *any* sufficiently accurate, computationally bounded server—even a stateful one—is insecure (Theorem 7.2). Under the Planted XOR assumption (Christ & Gunn, 2024), our attack recovers the exact parity function using queries that are computationally indistinguishable from honest i.i.d. uniform queries. This strengthens the covert learning attack of Canetti & Karchmer (2021); Karchmer (2023), which only handles independent fixed-rate random noise.

Our formalization highlights the fundamental tension between accuracy and security and establishes a connection with statistical learning theory. Crucially, prior theoretical works (Canetti & Karchmer, 2021; Karchmer, 2023) only constructed extraction *attacks* and proved impossibility results for passive defenses; no provably secure anti-distillation defense was previously known. Our work

provides the *first provably secure anti-distillation defenses* (Corollary 5.4 and Section 6), showing how computational hardness and server statefulness can help prevent distillation.

Additional discussion on related work, including comparison of our results to Canetti & Karchmer (2021); Karchmer (2023), can be found in Appendix A. All proofs are deferred to the appendix.

## 3 Formal Problem: The Distillation Game

We first establish the model characteristics: Let $\mathcal{X}$ be the domain of inputs and let $\mathcal{Y}$ be the range of outputs. We assume a probability distribution $\mathcal{D}$ over $\mathcal{X}$ that represents "normal" or "honest" inputs. Throughout the paper, we write $x \leftarrow \mathcal{D}$ to denote drawing a sample $x$ from $\mathcal{D}$ (and for a finite set $S$, $x \leftarrow S$ denotes a uniform sample from $S$). Let $\mathcal{H}$ be a class of (possibly randomized) functions $h : \mathcal{X} \to \mathcal{Y}$ or $h : \mathcal{X} \to \Delta(\mathcal{Y})$, where $\Delta(\mathcal{Y})$ denotes the space of probability distributions over $\mathcal{Y}$. For example, $\mathcal{H}$ could represent LLMs under a certain size (e.g., 1TB), in which case $\mathcal{X} = \mathcal{Y}$ are all strings of text (possibly limited by length). In the PAC learning setting (Valiant, 1984), $\mathcal{Y} = \{0, 1\}$ and $\mathcal{H}$ consists of deterministic classifiers.

We also need a loss function $\mathrm{L} : \Delta(\mathcal{Y}) \times \Delta(\mathcal{Y}) \to \mathbb{R}_{\geq 0}$. For simplicity, we will mostly work with the total variation distance $\mathrm{TV}(P, Q) := \sup_S P(S) - Q(S)$, which is the natural distributional analog of 0-1 loss. For many applications it is natural instead to work with KL divergence (or cross entropy) $\mathrm{KL}(P\|Q) := \mathop{\mathbb{E}}_{X \leftarrow P}[\log(P(X)/Q(X))]$.

We next define the interaction between an adversary/client $\mathcal{A}$ and a server/defender $\mathcal{S}$.

**Definition 3.1** (The Distillation Game)**.** *Let Distill$(\mathcal{A}, \mathcal{S}, h, \mathcal{D}, n, R)$ denote the following distillation game played between an adversary $\mathcal{A}$ and a server $\mathcal{S}$ holding a secret target function $h \in \mathcal{H}$. The game involves $R$ rounds and a total of $n$ queries.*

1. ***Initialization:*** *The server $\mathcal{S}$ initializes its internal state $st_0 \in \Sigma$. The server also has access to the model $h \in \mathcal{H}$ and the distribution $\mathcal{D}$. The adversary $\mathcal{A}$ does not have access to $h$, but is given $n$, $R$, and sample access to $\mathcal{D}$.*
2. ***Query Phase:*** *For rounds $r = 1, \ldots, R$:*
   (a) *$\mathcal{A}$ computes a batch of $n_r$ queries $X_r = (x_{r,1}, \ldots, x_{r,n_r}) \in \mathcal{X}^{n_r}$, where $\sum_{k=1}^{R} n_k \leq n$. The adversary $\mathcal{A}$ is stateful and its queries can depend on the results of prior rounds of interaction. $\mathcal{A}$ sends $X_r$ to $\mathcal{S}$.*
   (b) *$\mathcal{S}$ processes the queries, updating its state and generating labels:*

$$(Y_r, st_r) \leftarrow \mathcal{S}(X_r, h, \mathcal{D}, st_{r-1}). \tag{3.1}$$

   (c) *$\mathcal{A}$ receives $Y_r = (y_{r,1}, \ldots, y_{r,n_r}) \in \mathcal{Y}^{n_r}$ from $\mathcal{S}$ (and updates its state accordingly).*
3. ***Output:*** *$\mathcal{A}$ outputs a model $\hat{h} : \mathcal{X} \to \mathcal{Y}$ or $\hat{h} : \mathcal{X} \to \Delta(\mathcal{Y})$.*

*The* transcript *of the game is $(X, Y, \hat{h}) \leftarrow$ Distill$(\mathcal{A}, \mathcal{S}, h, \mathcal{D}, n, R)$, where $X = (X_1, \ldots, X_R) \in \mathcal{X}^n$ and $Y = (Y_1, \ldots, Y_R) \in \mathcal{Y}^n$.*

The state space $\Sigma$ of the server $\mathcal{S}$ captures the ability of the server to "remember" prior interactions. If $|\Sigma| = 1$, then the server is stateless and responds to each round of queries independently. If $|\Sigma| = 1$ and $n = R$, then we say the server is *memoryless* and it responds to each query independently.

Now we formalize the two competing requirements—utility for honest users and security against (malicious) adversaries—in the distillation game. Note that the accuracy guarantee is required only for honest (i.i.d.) queries and does not promise anything for adversarial queries. To state accuracy formally for stateful and randomized servers, let $\pi_{r,i} := (X_{\leq r}, Y_{<r}, y_{r,<i})$ denote the history of queries up to round $r$ and all server responses prior to the $i$-th query of round $r$, and let $P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}) \in \Delta(\mathcal{Y})$ denote the conditional distribution of the server's response $y_{r,i}$ given $\pi_{r,i}$.

**Definition 3.2** ($\alpha$-Accuracy for Honest Queries)**.** *An* honest client $\mathcal{A}_{honest}$ *is a client participating in the Distill$(\mathcal{A}_{honest}, \mathcal{S}, h, \mathcal{D}, n, R)$ game such that, in every round $r \in [R]$, $\mathcal{A}_{honest}$ chooses its queries $X_r = (x_{r,1}, \ldots, x_{r,n_r})$ by sampling each $x_{r,j}$ independently from $\mathcal{D}$, irrespective of prior rounds.*

*A server $\mathcal{S}$ is* $\alpha$-accurate with loss L for $n$ queries in $R$ rounds with respect to model $h$ and distribution $\mathcal{D}$ *if, for any honest client $\mathcal{A}_{honest}$, we have*

$$\mathop{\mathbb{E}}_{(X,Y,\hat{h})\leftarrow \mathit{Distill}(\mathcal{A}_{honest},\mathcal{S},h,\mathcal{D},n,R)} \left[\sum_{r=1}^{R}\sum_{i=1}^{n_r} \mathrm{L}\left(P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}), h(x_{r,i})\right)\right] \le n\alpha. \tag{3.2}$$

*We say that $\mathcal{S}$ is* $\alpha$-accurate with respect to $(\mathcal{H},\mathcal{D})$ *if* (3.2) *holds for every $h \in \mathcal{H}$, and we denote by $\mathfrak{S}(\alpha,\mathcal{D})$ (resp. $\mathfrak{S}_{mem}(\alpha,\mathcal{D})$) the class of all servers (resp. memoryless servers) that are $\alpha$-accurate with respect to $(\mathcal{H},\mathcal{D})$.*

When $h : \mathcal{X} \to \mathcal{Y}$ is a deterministic model and $\mathrm{L} = \mathrm{TV}$, we have $\mathrm{TV}\left(P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}), h(x_{r,i})\right) = \mathbb{P}\left[y_{r,i} \neq h(x_{r,i}) \mid \pi_{r,i}\right]$; (3.2) then simplifies to $\mathbb{E}\left[\sum_{r=1}^{R}\sum_{i=1}^{n_r} \mathbb{I}[y_{r,i} \neq h(x_{r,i})]\right] \le n\alpha$, which is the standard expected 0-1 error in PAC learning.

Our accuracy guarantee bounds the expected error; one may alternatively demand a high-probability bound (see Appendix B), which we omit in the main text for brevity.

For any adversary $\mathcal{A}$, server $\mathcal{S}$, target $h \in \mathcal{H}$, and distribution $\mathcal{D}$, we denote the expected population loss of the distilled hypothesis $\hat{h}$ by

$$\mathrm{Risk}(\mathcal{A},\mathcal{S},h,\mathcal{D},n,R) := \mathop{\mathbb{E}}_{(X,Y,\hat{h})\leftarrow \mathrm{Distill}(\mathcal{A},\mathcal{S},h,\mathcal{D},n,R)} \left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{L}\left(\hat{h}(x), h(x)\right)\right]\right]. \tag{3.3}$$

When $n$ and $R$ are clear from context, we write $\mathrm{Risk}(\mathcal{A},\mathcal{S},h,\mathcal{D})$. We now define anti-distillation security (and insecurity) first for a *fixed* server $\mathcal{S}$ and distribution $\mathcal{D}$ (Definition 3.3), and then in the *distribution-free / universal* setting where one quantifies over all distributions $\mathcal{D}$ and $\alpha$-accurate servers $\mathcal{S} \in \mathfrak{S}(\alpha,\mathcal{D})$ (Definition 3.4).

**Definition 3.3** (Fixed-Server $\gamma$-Anti-Distillation Security and Insecurity)**.** *Fix a model class $\mathcal{H}$, distribution $\mathcal{D}$ over $\mathcal{X}$, query parameters $(n,R)$, and adversary computational bound $\mathcal{T}_A$.*

1. ***Worst-Case ($\gamma$-WC) Anti-Distillation Security:*** *A server $\mathcal{S}$ provides* worst-case $\gamma$-anti-distillation security *(or simply* $\gamma$-anti-distillation security*) against $\mathcal{T}_A$ with respect to $(\mathcal{H},\mathcal{D})$ if, for every adversary $\mathcal{A} \in \mathcal{T}_A$, there exists a model $h \in \mathcal{H}$ on which $\mathcal{A}$ incurs expected risk at least $\gamma$. Equivalently,*

$$\forall \mathcal{A} \in \mathcal{T}_A \; \exists h \in \mathcal{H}, \quad \mathrm{Risk}(\mathcal{A},\mathcal{S},h,\mathcal{D},n,R) \ge \gamma. \tag{3.4}$$

   *Conversely, $\mathcal{S}$ is* $\gamma$-distillation insecure *on $(\mathcal{H},\mathcal{D})$ if $\exists \mathcal{A} \in \mathcal{T}_A$ such that*

$$\forall h \in \mathcal{H}, \quad \mathrm{Risk}(\mathcal{A},\mathcal{S},h,\mathcal{D},n,R) \le \gamma.$$

2. ***Average-Case ($\gamma$-AC) Anti-Distillation Security:*** *Given a prior distribution $\mu \in \Delta(\mathcal{H})$ over $\mathcal{H}$, a server $\mathcal{S}$ provides* average-case $\gamma$-anti-distillation security *against $\mathcal{T}_A$ with respect to $(\mu,\mathcal{D})$ if:*

$$\forall \mathcal{A} \in \mathcal{T}_A, \quad \mathop{\mathbb{E}}_{h\leftarrow\mu}\left[\mathrm{Risk}(\mathcal{A},\mathcal{S},h,\mathcal{D},n,R)\right] \ge \gamma. \tag{3.5}$$

**Quantifiers over the Model Class $\mathcal{H}$ ($\forall\mathcal{A}\forall h$ vs. $\forall\mathcal{A}\mathbb{E}_{h\leftarrow\mu}$ vs. $\forall\mathcal{A}\exists h$).** Ideally, a defender holding a specific model $h \in \mathcal{H}$ would want $\forall \mathcal{A} \in \mathcal{T}_A \; \forall h \in \mathcal{H}, \mathrm{Risk}(\mathcal{A},\mathcal{S},h,\mathcal{D}) \ge \gamma$. However, this $\forall\mathcal{A}\forall h$ guarantee is impossible for any non-trivial class $\mathcal{H}$: for any $h_0 \in \mathcal{H}$, an adversary $\mathcal{A}_{h_0}$ that hardcodes and outputs $\hat{h} = h_0$ achieves zero risk whenever the true model is $h_0$. Worst-case security (3.4) ($\forall\mathcal{A}\exists h$) is a minimal relaxation that avoids this triviality. Average-case security (3.5) ($\forall\mathcal{A}\mathbb{E}_{h\leftarrow\mu}[\mathrm{Risk}] \ge \gamma$) is even stronger and protects a typical model drawn from $\mu$.

**Quantifiers over the Distribution $\mathcal{D}$ and Server $\mathcal{S}$ ($\exists(\mathcal{D},\mathcal{S})\forall\mathcal{A}$ vs. $\forall\mathcal{A}\exists(\mathcal{D},\mathcal{S})$).** We also distinguish whether a *single fixed* pair $(\mathcal{D},\mathcal{S})$ defeats all adversaries (Definition 3.3: $\exists\mathcal{D},\mathcal{S}\;\forall\mathcal{A}$), or whether $\mathcal{D}$ and $\mathcal{S}$ may depend on $\mathcal{A}$ ($\forall\mathcal{A}\;\exists\mathcal{D},\mathcal{S}$). In distribution-free PAC learning (Valiant, 1984; Kearns & Vazirani, 1994), a learner must succeed across *all* distributions $P$. Thus, allowing $\mathcal{D},\mathcal{S}$ to depend on $\mathcal{A}$ (as formalized below) naturally corresponds to a distribution-free learning setting.

**Definition 3.4** (Distribution-Free $\gamma$-Anti-Distillation Security and Universal $\gamma$-Insecurity)**.** *Fix a model class $\mathcal{H}$, accuracy parameter $\alpha \geq 0$, query parameters $(n, R)$, adversary class $\mathcal{T}_A$, and a family of servers $\mathfrak{S}$. We say that $\mathfrak{S}$ provides* distribution-free (adversary-dependent) $\gamma$-anti-distillation security *against $\mathcal{T}_A$ if, for every adversary $\mathcal{A} \in \mathcal{T}_A$, there exist a distribution $\mathcal{D}$, an $\alpha$-accurate server $\mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D})$, and a model $h \in \mathcal{H}$ such that the expected risk is at least $\gamma$. That is,*

$$\forall \mathcal{A} \in \mathcal{T}_A \; \exists \mathcal{D} \; \exists \mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D}) \; \exists h \in \mathcal{H}, \quad \mathrm{Risk}(\mathcal{A}, \mathcal{S}, h, \mathcal{D}, n, R) \geq \gamma. \tag{3.6}$$

*Conversely, $\mathfrak{S}$ is* universally $\gamma$-distillation insecure *against $\mathcal{T}_A$ if $\exists \mathcal{A} \in \mathcal{T}_A \; \forall \mathcal{D} \; \forall \mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D}) \; \forall h \in \mathcal{H}$ such that* $\mathrm{Risk}(\mathcal{A}, \mathcal{S}, h, \mathcal{D}, n, R) \leq \gamma$.

Fixed-Server $\gamma$-Security ($\exists \mathcal{D}, \mathcal{S} \; \forall \mathcal{A} \; \exists h$, Definition 3.3) immediately implies Distribution-Free $\gamma$-Security ($\forall \mathcal{A} \; \exists \mathcal{D}, \mathcal{S}, h$, Definition 3.4), whereas the converse does not hold in general. Table 3.1 summarizes our core notations and definitions.

Table 3.1: Summary of security/insecurity definitions in the Distillation Game.

| Notation | Mathematical Expression | Meaning and Where Used |
|---|---|---|
| **Fixed-$(\mathcal{S}, \mathcal{D})$ $\gamma$-WC Security** | $\forall \mathcal{A} \in \mathcal{T}_A \; \exists h \in \mathcal{H} : \mathrm{Risk}(\mathcal{A}, \mathcal{S}, h, \mathcal{D}) \geq \gamma$ | Fixed $\mathcal{S}$ on $\mathcal{D}$ protects at least one $h \in \mathcal{H}$ per $\mathcal{A}$ (Def. 3.3(1)) |
| **Fixed-$(\mathcal{S}, \mathcal{D})$ $\gamma$-AC Security** | $\forall \mathcal{A} \in \mathcal{T}_A : \mathop{\mathbb{E}}_{h \leftarrow \mu} [\mathrm{Risk}(\mathcal{A}, \mathcal{S}, h, \mathcal{D})] \geq \gamma$ | Fixed $\mathcal{S}$ on $\mathcal{D}$ protects a random $h \leftarrow \mu$ against all $\mathcal{A}$ (Def. 3.3(2), Thm. 6.3) |
| **Fixed-$(\mathcal{S}, \mathcal{D})$ $\gamma$-Insecurity** | $\exists \mathcal{A} \in \mathcal{T}_A \; \forall h \in \mathcal{H} : \mathrm{Risk}(\mathcal{A}, \mathcal{S}, h, \mathcal{D}) \leq \gamma$ | Negation of Def. 3.3(1): $\mathcal{A}$ distills fixed $(\mathcal{S}, \mathcal{D})$ for all $h \in \mathcal{H}$ (Thm. 4.1, Thm. 7.2) |
| **Distribution-Free $\gamma$-Security** | $\forall \mathcal{A} \in \mathcal{T}_A \; \exists \mathcal{D} \; \exists \mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D}) \; \exists h \in \mathcal{H} : \mathrm{Risk} \geq \gamma$ | Each $\mathcal{A}$ fails on some hard $(\mathcal{D}, \mathcal{S}, h)$; weaker than $\exists(\mathcal{D}, \mathcal{S})$ (Def. 3.4, Cor. 5.4) |
| **Universal $\gamma$-Insecurity** | $\exists \mathcal{A} \in \mathcal{T}_A \; \forall \mathcal{D} \; \forall \mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D}) \; \forall h \in \mathcal{H} : \mathrm{Risk} \leq \gamma$ | Negation of Def. 3.4: one $\mathcal{A}$ distills all $\mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D})$ on all $\mathcal{D}$ (Def. 3.4, Thm. 4.1, 5.2, 5.3) |

**Efficiency.** Following standard convention in learning theory (Kearns & Vazirani, 1994), we assume $\mathcal{H} = \bigcup_{d \in \mathbb{N}} \mathcal{H}_d$, $\mathcal{X} = \bigcup_{d \in \mathbb{N}} \mathcal{X}_d$, and $\mathcal{Y} = \bigcup_{d \in \mathbb{N}} \mathcal{Y}_d$, where $\mathcal{H}_d$ contains hypotheses $h : \mathcal{X}_d \to \Delta(\mathcal{Y}_d)$ parameterized by complexity $d$. An algorithm is *efficient* (denoted $\mathcal{T}_{\text{eff}}$) if it runs in $d^{O(1)}$ time.

## 4 Information-Theoretic Distillation

As a baseline, we establish the information-theoretic possibility of distillation in our setup. However, the attack is not computationally efficient. Thus the takeaway from this result is that anti-distillation defenses *must* exploit the fact that the attacker is computationally bounded.

**Theorem 4.1.** *For any finite model class $\mathcal{H}$, query parameters $(n, R)$, and accuracy parameter $\alpha \geq 0$, the family of all servers $\mathfrak{S}$ (with* TV *loss) is* universally $\gamma$-distillation insecure *(Definition 3.4) against computationally unbounded adversaries for $\gamma = 2\left(\alpha + \sqrt{\frac{1}{n} \log |\mathcal{H}|}\right)$.*

To paraphrase the theorem: If the server is $\alpha$-accurate for honest clients, then a computationally unbounded adversary can recover the model up to accuracy $\gamma = O(\alpha)$ using $n = O(\log |\mathcal{H}|/\alpha^2)$ queries. As a hypothetical comparison point, if the sample complexity were $n = \log_2 |\mathcal{H}|$ with $\gamma = 0$, that would mean the adversary can effectively download one bit of the model per query. The catch is that implementing this adversary may not be computationally tractable.

The attacker in Theorem 4.1 simply acts like an honest client and queries honestly. For deterministic models, the attacker then employs a standard empirical risk minimization (ERM) strategy. For randomized models, the analysis is more involved and thus we defer it to Appendix C. Rather than ERM, we use a minimum Scheffé distance estimate (Devroye & Lugosi, 2001); note that this proof is specific to the total variation loss, although analogous results hold for other losses.

## 5 Equivalence of Memoryless Distillation and Agnostic PAC Learning

Our information-theoretic attack in the previous section ignores computational complexity. In this section, we consider when such an attack can be performed efficiently against a *memoryless server*

(which answers each query independently). We show that in the distribution-free setting, universal distillation against memoryless servers is computationally *equivalent* to agnostic PAC learning.

The strategy described for Theorem 4.1 above essentially shows that agnostic learning is a form of distillation attack. Here we show that computational efficiency carries through in both directions.

**Preliminaries and Definitions.** For simplicity, we restrict our attention to deterministic models $h : \mathcal{X} \to \mathcal{Y}$ and any bounded loss function $\mathrm{L} : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ that is a metric; in particular, it is symmetric ($\mathrm{L}(y_1, y_2) = \mathrm{L}(y_2, y_1)$) and satisfies the triangle inequality $\mathrm{L}(y_1, y_3) \leq \mathrm{L}(y_1, y_2) + \mathrm{L}(y_2, y_3)$ for all $y_1, y_2, y_3 \in \mathcal{Y}$ (such as the 0-1 loss). For a distribution $P$ over $\mathcal{X} \times \mathcal{Y}$ and a model $h : \mathcal{X} \to \mathcal{Y}$, the expected population loss is $L_P(h) := \mathop{\mathbb{E}}_{(x,y)\leftarrow P}[\mathrm{L}(h(x), y)]$.

We formulate distribution-free agnostic PAC learning as a gap problem[1]:

**Definition 5.1** (Gap PAC Learner)**.** *An algorithm $\mathcal{A}_{PAC}$ is an $(\alpha, \gamma)$-Gap PAC learner with sample complexity $m$ for $\mathcal{H}$ with respect to loss* L *if the following holds for every distribution $P$ over $\mathcal{X} \times \mathcal{Y}$: if there exists $h^* \in \mathcal{H}$ with $L_P(h^*) \leq \alpha$, then given $m$ i.i.d. samples from $P$, $\mathcal{A}_{PAC}$ outputs $\hat{h}$ satisfying $\mathbb{E}\left[L_P(\hat{h})\right] \leq \gamma$, where the expectation is over both the samples and the algorithm's randomness.*

For the distillation game, we restrict our attention to the class $\mathfrak{S}_{\text{mem}}$ of *memoryless servers* (i.e., singleton state space $|\Sigma| = 1$ and one query per round $n = R$), which respond to each query $x \in \mathcal{X}$ independently according to a conditional distribution $\mathcal{S}(x) \in \Delta(\mathcal{Y})$. Extending L linearly to distributions $Q \in \Delta(\mathcal{Y})$ via $\mathrm{L}(Q, y_0) := \mathop{\mathbb{E}}_{y\leftarrow Q}[\mathrm{L}(y, y_0)]$ for $y_0 \in \mathcal{Y}$, the $\alpha$-accuracy condition (3.2) with respect to target $h \in \mathcal{H}$ and distribution $\mathcal{D}$ simplifies to $\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D},\, y\leftarrow\mathcal{S}(x)}[\mathrm{L}(y, h(x))] \leq \alpha$.

Throughout this section, we fix the (parameterized) concept class $\mathcal{H} = \bigcup_{d\in\mathbb{N}} \mathcal{H}_d$ and henceforth do not state it explicitly for brevity. Recall from Section 3 that an algorithm is efficient (denoted $\mathcal{T}_{\text{eff}}$) if it runs in $d^{O(1)}$ time, which implies that its sample/query complexity is also $d^{O(1)}$.

**Agnostic Learning Algorithm Implies Distillation Attack.** If we possess an efficient algorithm for Gap PAC learning, we can convert it into a universal distillation adversary that successfully extracts a model from any $\alpha$-accurate memoryless server across any distribution $\mathcal{D}$, simply by submitting honest i.i.d. queries from $\mathcal{D}$ and running the learner.

**Theorem 5.2.** *If there is an efficient $(\alpha, \gamma)$-Gap PAC learner $\mathcal{A}_{PAC}$ for $\mathcal{H}$, then the class of $\alpha$-accurate memoryless servers $\mathfrak{S}_{mem}$ is* universally $(\gamma + \alpha)$-distillation insecure *against efficient adversaries (Definition 3.4).*

**Universal Distillation Attack Implies Agnostic Learning Algorithm.** The converse reduction requires more care. Nevertheless, we show that, if an adversary efficiently distills a model over any distribution, we can leverage it to agnostically learn $\mathcal{H}$.

**Theorem 5.3.** *If the class of $\alpha$-accurate memoryless servers $\mathfrak{S}_{mem}$ is* universally $\gamma$-distillation insecure *against efficient adversaries (Def. 3.4), then, for any $\varepsilon \geq \frac{1}{d^{O(1)}}$, there exists an efficient $(\alpha - \varepsilon, \gamma + \alpha + \varepsilon)$-Gap PAC learner $\mathcal{A}_{PAC}$ for $\mathcal{H}$.*

The contrapositive of Theorem 5.3 translates the computational hardness of distribution-free agnostic learning into *Distribution-Free Anti-Distillation Security* for numerous concept classes known to be hard to agnostically learn under standard complexity assumptions (e.g., halfspaces (Daniely, 2016), DNFs, and intersections of halfspaces (Daniely & Vardi, 2021; Klivans & Sherstov, 2009)):

**Corollary 5.4** (Hardness of Gap PAC Implies Distribution-Free Anti-Distillation Security)**.** *If the distribution-free $(\alpha, \gamma)$-Gap PAC learning problem for $\mathcal{H}$ is computationally hard, then for any $\varepsilon \geq \frac{1}{d^{O(1)}}$, the class of $(\alpha + \varepsilon)$-accurate memoryless servers $\mathfrak{S}_{mem}$ achieves* Distribution-Free $(\gamma - \alpha - \varepsilon)$-Anti-Distillation Security *(Definition 3.4) against all efficient adversaries.*

Notably, the proof of Theorem 5.3 (and hence the corresponding defense in Corollary 5.4) uses an "allowlist" technique: the simulated server $\mathcal{S}_{\text{sim}}$ only provides accurate responses on a polynomial-size allowlist $X_m = \{x_1, \ldots, x_m\}$ supporting $\mathcal{D}$ and returns a default output on all other queries. This mirrors practical API defenses that restrict informative answers to a curated set of allowed inputs.

[1]For the 0-1 loss, due to boosting (e.g. (Brukhim et al., 2023)), the standard notion for efficient agnostic PAC learning is equivalent to having an efficient *weak learner*, i.e. an $(\alpha, \gamma)$-Gap PAC learner for any $\gamma < 1 - 1/|\mathcal{Y}|$.

## 6 FIXED-SERVER STATEFUL DEFENSE ON THE UNIFORM DISTRIBUTION

While Corollary 5.4 yields a memoryless defense from the hardness of agnostic PAC learning, it only guarantees *distribution-free security* ($\forall\mathcal{A}\exists\mathcal{D},\mathcal{S}$), where both the "allowlist" distribution $\mathcal{D}$ and the server $\mathcal{S}$ depend on the adversary $\mathcal{A}$. In this section, we construct a stronger *fixed-server* defense ($\exists\mathcal{D},\mathcal{S}\ \forall\mathcal{A}\ \mathbb{E}_{h\leftarrow\mathcal{H}}$, Definition 3.3(2)): under the natural *uniform distribution* $\mathcal{D}$ over $\{0,1\}^d$, a single stateful server defeats all polynomial-time adversaries in the average case over $\mathcal{H}$.

The intuition for our defense is that many classes of classifiers (e.g., parities (Blum et al., 2003), halfspaces (Guruswami & Raghavendra, 2009), DNFs (Khot & Saket, 2008)) are computationally easy to learn in the noiseless PAC model, but are computationally hard to learn in the presence of label noise. Thus, one could hope that the server can carefully inject noise to prevent the adversary from learning the model. However, in the distillation game, the adversary can adaptively choose its queries, whereas standard learning hardness (such as Learning Parity with Noise) requires the samples to be uniformly random. Indeed, parities are easy to learn with chosen queries, even in the presence of random noise, by using repetition to denoise the answers and then applying Gaussian elimination. We overcome this challenge via two complementary ingredients:

1. **Cryptographic Hash Preprocessing:** We pass inputs through a cryptographic hash function $H$ (modeled as a random oracle) before applying the parity function: $c_a(x) = \langle H(x), a\rangle \bmod 2$. Even though the adversary can choose the queries $x_i$ arbitrarily (and has full oracle access to $H$), the hashed vectors $H(x_i)$ are independent and uniformly distributed in $\{0,1\}^d$, preventing the adversary from crafting algebraic linear dependencies among the inputs.
2. **Stateful Query Caching:** Without state, an adversary could still query the same point $x$ multiple times and take a majority vote to denoise the answers. By maintaining a stateful cache $T$ that records and replays the noisy response $T[x]$ whenever $x$ is queried again, the server ensures that repeated queries yield zero additional information.

We now formalize this construction, beginning with the Learning Parity with Noise (LPN) assumption.

**Definition 6.1** (Learning Parity with Noise (LPN))**.** *The $\mathrm{LPN}_{d,p}$ assumption with dimension $d$ and noise rate $p \in (0, 1/2)$ states that, for $n = \mathrm{poly}(d)$, the following two distributions over $(\{0,1\}^d \times \{0,1\})^n$ are computationally indistinguishable:*

- ***LPN distribution:*** *$(z_i, \langle z_i, a\rangle \oplus b_i)_{i=1}^n$, where $a \leftarrow \{0,1\}^d$ is chosen uniformly at random once, each $z_i \leftarrow \{0,1\}^d$ is drawn uniformly and independently, and each $b_i \leftarrow \mathrm{Bernoulli}(p)$ independently.*
- ***Uniform distribution:*** *$(z_i, u_i)_{i=1}^n$, where each $z_i \leftarrow \{0,1\}^d$ and each $u_i \leftarrow \{0,1\}$ are drawn uniformly and independently.*

*All arithmetic is modulo 2.*

The LPN assumption is a standard assumption in cryptography (Blum et al., 2003) and is believed to hold for noise rates as low as $p = \Theta(1/d^{1-o(1)})$ (Brakerski et al., 2018; 2019; Bitansky & Freizeit, 2022).

**Construction.** Let $H : \{0,1\}^d \to \{0,1\}^d$ be a hash function modeled as a random oracle. Both the server and the adversary have oracle access to $H$. Define the concept class $\mathcal{H} = \{c_a : a \in \{0,1\}^d\}$, where $c_a : \{0,1\}^d \to \{0,1\}$ is given by $c_a(x) = \langle H(x), a\rangle \bmod 2$, and let $\mathcal{D}$ be the uniform distribution over $\{0,1\}^d$.

The server $\mathcal{S}_p$, holding secret $a \in \{0,1\}^d$, is *stateful* and maintains a cache $T$ (initially empty). On query $x$:

- If $x \in T$: return $T[x]$.
- Else: sample $b \leftarrow \mathrm{Bernoulli}(p)$, set $T[x] \leftarrow c_a(x) \oplus b$, and return $T[x]$.

The cache ensures that repeated queries receive consistent responses, preventing the adversary from averaging out the noise. More formally, we prove the following for our construction.

**Proposition 6.2** (Non-Triviality: Learnability without Noise)**.** *When $p = 0$, the concept class $\mathcal{H}$ is efficiently learnable: there exists a polynomial-time adversary using $d$ server queries in one round that recovers $c_a$ exactly (with overwhelming probability over the random oracle).*

**Theorem 6.3** (Fixed-Server Average-Case Anti-Distillation Security via LPN)**.** *Under the* $\mathrm{LPN}_{d,p}$ *assumption (Definition 6.1), in the random oracle model, the stateful server* $\mathcal{S}_p$ *satisfies:*

1. ***Accuracy:*** *For any* $n = \mathrm{poly}(d)$*,* $\mathcal{S}_p$ *is* $p$*-accurate with loss* TV *for* $n$ *queries in* $R$ *rounds with respect to every* $c_a \in \mathcal{H}$ *and the uniform distribution* $\mathcal{D}$*.*
2. ***Fixed-Server Average-Case Anti-Distillation Security:*** $\mathcal{S}_p$ *provides* average-case $(1/2 - \mathrm{negl}(d))$-anti-distillation security *(Definition 3.3(2)) over* $\mu = \mathrm{Uniform}(\mathcal{H})$ *with loss* TV *against any polynomial-time adversary making* $n = \mathrm{poly}(d)$ *server queries and* $q = \mathrm{poly}(d)$ *random oracle queries with respect to* $\mathcal{H}$ *and* $\mathcal{D}$*.*

Together, Proposition 6.2 and Theorem 6.3 show that the defense is non-trivial, accurate, and secure in the strongest fixed-server average-case sense: without noise ($p = 0$), the adversary can efficiently learn $c_a$ (Proposition 6.2); with noise ($p > 0$) and stateful caching, any $\mathrm{poly}(d)$-time adversary is reduced to essentially random guessing on a random target $c_a \leftarrow \mathcal{H}$ under LPN (Theorem 6.3).

## 7 Example of an Undetectable Attack

In Sections 5 and 6, we showed that computational hardness (combined with stateful caching and hash preprocessing in Section 6) can serve as the basis of an accurate and secure anti-distillation defense. A natural question is whether such a defense is always possible without altering the input structure of the concept class.

In this section, we show that the answer is no: for the standard concept class of Boolean parities over $\{0,1\}^d$ under the uniform distribution $\mathcal{D}$, there exists an efficient distillation attack that succeeds against *any* computationally bounded server providing sufficiently accurate answers (including stateful, adaptive servers), and moreover, the attack is *undetectable*—the server cannot computationally distinguish the adversary's queries from honest i.i.d. queries drawn from $\mathcal{D}$.

**Setup.** Let $\mathcal{H} = \{h_a : a \in \{0,1\}^d\}$ where $h_a(x) = \langle x, a\rangle \bmod 2$, and let $\mathcal{D}$ be the uniform distribution over $\{0,1\}^d$. We rely on the following computational assumption (Agrawal et al., 2024; Christ & Gunn, 2024).

**Definition 7.1** (Planted XOR Assumption (Christ & Gunn, 2024))**.** *The* $(n, m, t)$ *planted XOR assumption (*$\mathrm{XOR}_{m,t}$*) states that the following two distributions over* $\{0,1\}^{n\times m}$ *are computationally indistinguishable:*

- ***Null distribution*** $\mathcal{D}_0(n,m)$***:*** *A uniformly random matrix* $G \in \{0,1\}^{n\times m}$*.*
- ***Planted distribution*** $\mathcal{D}_1(n,m,t)$***:*** *Sample a random* $t$*-sparse vector* $s \in \{0,1\}^n$ *(i.e.,* $s$ *has exactly* $t$ *nonzero entries), then sample* $G \in \{0,1\}^{n\times m}$ *uniformly subject to* $s^\top G = 0$*.*

Our main result of this section is stated formally below.

**Theorem 7.2** (Undetectable Distillation Attack)**.** *For any constant* $\varepsilon \in (0, 1/2)$*, assume the* $\mathrm{XOR}_{n^{1-\Theta(\varepsilon)}, n^{\Theta(\varepsilon)}}$ *assumption (Definition 7.1). There exists an efficient adversary* $\mathcal{A} \in \mathcal{T}_{\mathit{eff}}$ *making* $n = 2d^{1+1.7\varepsilon}$ *queries in* $1$ *round such that, for any (stateful, randomized) efficient server* $\mathcal{S} \in \mathfrak{S}(d^{-1/2-\varepsilon}, \mathcal{D})$ *that is* $\left(d^{-1/2-\varepsilon}\right)$*-accurate for honest queries under the uniform distribution* $\mathcal{D}$*:*

1. ***Distillation (Fixed-*$(\mathcal{S},\mathcal{D})$ ***Insecurity):*** *For every* $a \in \{0,1\}^d$*,* $\mathcal{A}$ *recovers* $h_a$ *(i.e., the exact coefficient vector* $a$*) with probability* $1 - o(1)$*, achieving* $\mathrm{Risk}(\mathcal{A}, \mathcal{S}, h_a, \mathcal{D}, n, 1) = o(1)$*.*
2. ***Undetectability:*** *The query batch* $X \in (\{0,1\}^d)^n$ *submitted by* $\mathcal{A}$ *is computationally indistinguishable from* $n$ *honest queries drawn i.i.d. from* $\mathcal{D}$*.*

We remark that the accuracy parameter $\alpha = d^{-1/2-\varepsilon}$, while small, is interesting: As mentioned earlier, LPN (Definition 6.1) is believed to be hard for this regime of noise rate $p = \alpha$. This means that, even for a memoryless server that injects independent random noise with noise rate $\alpha$, an *honest* client *cannot* efficiently learn $\mathcal{H}$. This necessitates a dishonest client, which also needs to be carefully constructed to be undetectable. We will next describe high-level ideas of the construction below.

**Construction Overview.** All arithmetic below is modulo 2. For a matrix $A$, we write $A_i$ for its $i$-th row. Recall that the target function is $h_a(x) = \langle x, a\rangle$ for some unknown $a \in \{0,1\}^d$. The adversary's goal is to recover $a$. For certain $n, m$, the adversary constructs queries as follows:

1. Sample a random matrix $P \in \{0,1\}^{m \times n}$ where each row has sparsity $t$.
2. Sample $G \in \{0,1\}^{n \times d}$ uniformly at random subject to $PG = 0$.
3. Sample $B \in \{0,1\}^{n \times d}$ where each entry is independently set to 1 with probability $1/dt$.
4. Submit the rows of $X = G + B$ as queries: $x_1, \ldots, x_n$, and receive labels $y_1, \ldots, y_n$.

**Undetectability.** By Lemma 8 of Christ & Gunn (2024), under the $\mathrm{XOR}_{m+d,t}$ assumption (where in our parameter regime $m = d^{1+\varepsilon} \gg d$ and $n = d^{1+1.7\varepsilon}$, so $m+d = n^{1-\Theta(\varepsilon)}$ and $t = d^{0.1\varepsilon} = n^{\Theta(\varepsilon)}$), the matrix $G$ is computationally indistinguishable from a uniformly random matrix. Since $B$ is sampled independently of $G$, $X = G + B$ is also computationally indistinguishable from a uniformly random matrix. Hence, no polynomial-time server can distinguish the adversary's queries from i.i.d. draws from $\mathcal{D}$.

**Recovering the model coefficients.** Let us fix $j \in [d]$. For each $i \in [m]$, $(PB)_{i,j}$ is the XOR of $t$ independent $\mathrm{Bernoulli}(1/dt)$ random variables. It can be seen that, with high probability there exists a row $i_j \in [m]$ such that $(PB)_{i_j}$ is the indicator vector $e_j$ (i.e., all zeros except a 1 in position $j$). The adversary identifies such an $i_j$ for each $j$ and sets $\hat{a}_j = (Py)_{i_j}$, where $y = (y_1, \ldots, y_n)^\top$.

**Analysis: noiseless case ($\alpha = 0$).** When the server returns exact labels $y = Xa = (G + B)a$,

$$(Py)_{i_j} = (P(G+B)a)_{i_j} = (PGa)_{i_j} + (PBa)_{i_j} = 0 + (PB)_{i_j} \cdot a = e_j^\top a = a_j. \tag{7.1}$$

**Analysis: noisy case ($\alpha > 0$).** When there is some noise, we can only show that the above procedure recovers $1 - \tilde{O}(\alpha t)$ fraction of the coordinates $j$ correctly. Unfortunately, this is insufficient for us since we wish to recover all coordinates. Thus, in our full construction (Appendix F), we construct another "block" of $P, G, B$ but with a higher-density $B$ to recover all coordinates.

## 8 Conclusion & Discussion

In this paper, we developed a foundational theoretical framework to answer when provable anti-distillation security is mathematically achievable and when distillation is unavoidable. By explicitly separating the quantifiers over the adversary $\mathcal{A}$, input distribution $\mathcal{D}$, server $\mathcal{S}$, and target hypothesis $h \in \mathcal{H}$ (summarized in Table 3.1), our results provide several insights for anti-distillation:

1. *Information-theoretic inevitability ($\mathcal{T}_{all}$):* Theorem 4.1 proves that without computational restrictions on the adversary, **Universal $\gamma$-Distillation Insecurity** holds with $\gamma = 2\big(\alpha + \sqrt{\frac{1}{n}\log|\mathcal{H}|}\big)$ against *all* $\alpha$-accurate servers $\mathfrak{S}(\alpha, \mathcal{D})$, including stateful and adaptive ones. Any effective anti-distillation defense must therefore rely on computational hardness.
2. *Exact equivalence for memoryless defenses ($\mathfrak{S}_{mem}$):* Theorems 5.2 and 5.3 establish that for memoryless servers $\mathfrak{S}_{\mathrm{mem}}(\alpha, \mathcal{D})$, **Universal Distillation Insecurity** ($\exists\mathcal{A}\forall\mathcal{D}\forall\mathcal{S}\forall h$) is computationally equivalent to distribution-free $(\alpha, \gamma)$-Gap PAC learning. Contrapositively, **Distribution-Free Anti-Distillation Security** ($\forall\mathcal{A}\exists\mathcal{D}\exists\mathcal{S}\exists h$) is equivalent to the computational hardness of distribution-free agnostic PAC learning (Corollary 5.4).
3. *Statefulness achieves Fixed-Server Average-Case Security:* While distribution-free PAC hardness only yields an adversary-dependent hard distribution ($\forall\mathcal{A}\exists\mathcal{D}$), Theorem 6.3 constructs a single stateful server $\mathcal{S}_p$ for hashed parities under the uniform distribution $\mathcal{D}$ that achieves **Fixed-$(\mathcal{S}_p, \mathcal{D})$ Average-Case Security** ($\forall\mathcal{A} \in \mathcal{T}_{\text{eff}}, \mathop{\mathbb{E}}_{h\leftarrow\mu}[\mathrm{Risk}] \geq \frac{1}{2} - \mathrm{negl}(d)$) under $\mathrm{LPN}_{d,p}$.
4. *Unhashed linear structure enables undetectable extraction:* Conversely, Theorem 7.2 shows that for standard Boolean parities over $\mathbb{F}_2$ under the uniform distribution $\mathcal{D}$, no efficient server—even a stateful, adaptive one—can prevent distillation: under the Planted XOR assumption, an efficient attacker achieves **Fixed-$(\mathcal{S}, \mathcal{D})$ Insecurity** ($\exists\mathcal{A}\forall\mathcal{S}\forall h$) while keeping its query batches computationally indistinguishable from honest i.i.d. draws from $\mathcal{D}$.

While our definitional framework (Section 3) and generic information-theoretic result (Section 4) apply to arbitrary input domains and randomized models (including probabilistic language models), our concrete cryptographic separations in Sections 6 and 7 focus on Boolean parity and hashed-parity concept classes. Extending both provable stateful defenses and undetectable extraction attacks to richer hierarchical model classes—such as two-layer neural networks, decision trees, and low-rank sequence models (Liu & Moitra, 2025)—and removing the random oracle in Theorem 6.3 via standard-model cryptographic primitives remain exciting open problems.

### AI USE STATEMENT

The main ideas of the definitions and all the proofs are conceptualized by the authors. We used generative AI tools to assist with drafting and editing exposition, and drafting mathematical proofs. All theoretical definitions, theorem statements, proofs, and citations have been reviewed and verified for correctness by the authors. We take full responsibility for the final content of this work, including all text and claims.

### REPRODUCIBILITY STATEMENT

This is a theoretical paper. All formal definitions, model assumptions (such as the Learning Parity with Noise assumption in Definition 6.1 and the Planted XOR assumption in Definition 7.1), and theorem statements are stated in the main text, and complete, self-contained mathematical proofs of all results are provided in Appendices B–F.

## REFERENCES


Shweta Agrawal, Sagnik Saha, Nikolaj I. Schwartzbach, Akhil Vanukuri, and Prashant Nalini Vasudevan. k-sum in the sparse regime: Complexity and applications. In *CRYPTO*, pp. 315–351, 2024. doi: 10.1007/978-3-031-68379-4_10. URL https://doi.org/10.1007/978-3-031-68379-4_10.

Ibrahim M Alabdulmohsin, Xin Gao, and Xiangliang Zhang. Adding robustness to support vector machines against adversarial reverse engineering. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 231–240, 2014. URL https://dl.acm.org/doi/abs/10.1145/2661829.2662047.

Dana Angluin and Michael Kharitonov. When won't membership queries help? *J. Comput. Syst. Sci.*, 50(2):336–355, 1995. doi: 10.1006/JCSS.1995.1026. URL https://doi.org/10.1006/jcss.1995.1026.

Anthropic. Consumer terms of service, 2025. https://www.anthropic.com/legal/consumer-terms.

Anthropic. Detecting and preventing distillation attacks, 2026. https://www.anthropic.com/news/detecting-and-preventing-distillation-attacks.

Nir Bitansky and Sapir Freizeit. Statistically sender-private OT from LPN and derandomization. In *CRYPTO*, pp. 625–653, 2022. doi: 10.1007/978-3-031-15982-4_21. URL https://doi.org/10.1007/978-3-031-15982-4_21.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

Olivier Bousquet, Daniel Kane, and Shay Moran. The optimal approximation factor in density estimation. In *Conference on Learning Theory*, pp. 318–341. PMLR, 2019. URL https://arxiv.org/abs/1902.05876.

Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous ibe, leakage resilience and circular security from new assumptions. In *EUROCRYPT*, pp. 535–564, 2018. doi: 10.1007/978-3-319-78381-9_20. URL https://doi.org/10.1007/978-3-319-78381-9_20.

Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In *EUROCRYPT*, pp. 619–635, 2019. doi: 10.1007/978-3-030-17659-4_21. URL https://doi.org/10.1007/978-3-030-17659-4_21.

Nataly Brukhim, Amit Daniely, Yishay Mansour, and Shay Moran. Multiclass boosting: Simple and intuitive weak learning criteria. In *NeurIPS*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/050f8591be3874b52fdac4e1060eeb29-Abstract-Conference.html.

Mark Bun, Gautam Kamath, Thomas Steinke, and Steven Z Wu. Private hypothesis selection. *Advances in Neural Information Processing Systems*, 32, 2019. URL https://arxiv.org/abs/1905.13229.

Ran Canetti and Ari Karchmer. Covert learning: How to learn with an untrusted intermediary. In *Theory of Cryptography Conference*, pp. 1–31. Springer, 2021. URL https://eprint.iacr.org/2021/764.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, et al. Stealing part of a production language model. In *International Conference on Machine Learning*, pp. 5680–5705. PMLR, 2024. URL https://arxiv.org/abs/2403.06634.

Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1309–1326, 2020. URL https://arxiv.org/abs/1811.02054.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

M. Christ and S. Gunn. Pseudorandom error-correcting codes. In *International Cryptology Conference*, pp. 325–347, 2024.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pp. 1125–1139. PMLR, 2024. URL https://arxiv.org/abs/2306.09194.

Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *STOC*, pp. 105–117, 2016. doi: 10.1145/2897518.2897520. URL https://doi.org/10.1145/2897518.2897520.

Amit Daniely and Gal Vardi. From local pseudorandom generators to hardness of learning. In *COLT*, pp. 1358–1394, 2021. URL http://proceedings.mlr.press/v134/daniely21a.html.

Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A Servedio. Learning poisson binomial distributions. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 709–728, 2012. URL https://arxiv.org/abs/1107.2702.

Luc Devroye and Gábor Lugosi. *Combinatorial methods in density estimation*, volume 1. Springer, 2001. URL https://doi.org/10.1007/978-1-4613-0125-7.

Adam Dziedzic, Muhammad Ahmad Kaleem, Yu Shen Lu, and Nicolas Papernot. Increasing the cost of model extraction with calibrated proof of work. In *ICLR*. OpenReview.net, 2022. URL https://openreview.net/forum?id=EAy7C1cgE1L.

Hao Fang, Tianyi Zhang, Tianqu Zhuang, Jiawei Kong, Kuofeng Gao, Bin Chen, Leqi Liang, Shu-Tao Xia, and Ke Xu. Towards distillation-resistant large language models: An information-theoretic perspective. *arXiv preprint arXiv:2602.03396*, 2026.

Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models : [extended abstract]. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pp. 931–942. IEEE, 2022. doi: 10.1109/FOCS54457.2022.00092. URL https://doi.org/10.1109/FOCS54457.2022.00092.

Google. Gemini api additional terms of service, 2026. https://ai.google.dev/gemini-api/terms.

Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765, 2009.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. URL https://arxiv.org/abs/1503.02531.

Svante Janson. Large deviations for sums of partly dependent random variables. *Random Struct. Algorithms*, 24(3):234–248, 2004. doi: 10.1002/RSA.20008. URL https://doi.org/10.1002/rsa.20008.

Nikola Jovanovic, Robin Staab, and Martin T. Vechev. Watermark stealing in large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22570–22593. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/jovanovic24a.html.

Ari Karchmer. Theoretical limits of provable security against model extraction by efficient observational defenses. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 605–621. IEEE, 2023. URL https://eprint.iacr.org/2022/1039.

Ari Karchmer. Agnostic membership query learning with nontrivial savings: New results and techniques. In Claire Vernade and Daniel Hsu (eds.), *International Conference on Algorithmic Learning Theory, 25-28 February 2024, La Jolla, California, USA*, volume 237 of *Proceedings of Machine Learning Research*, pp. 654–682. PMLR, 2024a. URL https://proceedings.mlr.press/v237/karchmer24a.html.

Ari Karchmer. Distributional pac-learning from nisan's natural proofs. In Venkatesan Guruswami (ed.), *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, Berkeley, CA, USA, January 30 - February 2, 2024*, volume 287 of *LIPIcs*, pp. 68:1–68:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024b. doi: 10.4230/LIPICS.ITCS.2024.68. URL https://doi.org/10.4230/LIPIcs.ITCS.2024.68.

Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994. ISBN 978-0-262-11193-5. URL https://mitpress.mit.edu/books/introduction-computational-learning-theory.

Subhash Khot and Rishi Saket. Hardness of minimizing and learning dnf expressions. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pp. 231–240. IEEE, 2008.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International conference on machine learning*, pp. 17061–17084. PMLR, 2023.

Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. Syst. Sci.*, 75(1):2–12, 2009. doi: 10.1016/J.JCSS.2008.07.008. URL https://doi.org/10.1016/j.jcss.2008.07.008.

Michael J. Kratsios. Adversarial distillation of american ai models. https://whitehouse.gov/wp-content/uploads/2026/04/NSTM-4.pdf, 2026.

Pingzhi Li, Zhen Tan, Mohan Zhang, Huaizhi Qu, Huan Liu, and Tianlong Chen. Doge: Defensive output generation for llm protection against knowledge distillation. *arXiv preprint arXiv:2505.19504*, 2025. URL https://arxiv.org/abs/2505.19504.

Allen Liu and Ankur Moitra. Model stealing for any low-rank language model. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pp. 1755–1761, 2025. URL https://arxiv.org/abs/2411.07536.

Satyaki Mahalanabis and Daniel Stefankovic. Density estimation in linear time. In *COLT*, 2008. URL https://arxiv.org/abs/0712.2869.

OpenAI. Terms of use, 2026. https://openai.com/policies/row-terms-of-use.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against DNN model stealing attacks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SyevYxHtDB.

Leyi Pan, Aiwei Liu, Shiyu Huang, Yijian Lu, Xuming Hu, Lijie Wen, Irwin King, and Philip S Yu. Can llm watermarks robustly prevent unauthorized knowledge distillation? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13228–13251, 2025. URL https://arxiv.org/abs/2502.11598.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. *Advances in Neural Information Processing Systems*, 37: 21079–21113, 2024.

Yash Savani, Asher Trockman, Zhili Feng, Yixuan Even Xu, Avi Schwarzschild, Alexander Robey, Marc Finzi, and J Zico Kolter. Antidistillation sampling. *arXiv preprint arXiv:2504.13146*, 2025.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pp. 601–618, 2016. URL https://arxiv.org/abs/1609.02943.

Leslie G. Valiant. A theory of the learnable. In *STOC*, pp. 436–445, 1984. doi: 10.1145/800057.808710. URL https://doi.org/10.1145/800057.808710.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Yixuan Even Xu, John Kirchenbauer, Yash Savani, Asher Trockman, Alexander Robey, Tom Goldstein, Fei Fang, and J Zico Kolter. Antidistillation fingerprinting. *arXiv preprint arXiv:2602.03812*, 2026. URL https://arxiv.org/abs/2602.03812.

Yannis G Yatracos. Rates of convergence of minimum distance estimators and kolmogorov's entropy. *The Annals of Statistics*, 13(2):768–774, 1985.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58851–58880. PMLR / OpenReview.net, 2024a. URL https://proceedings.mlr.press/v235/zhang24o.html.

Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. Membership inference attacks cannot prove that a model was trained on your data. *arXiv preprint arXiv:2409.19798*, 2024b. URL https://arxiv.org/abs/2409.19798.

Tingwei Zhang, John X Morris, and Vitaly Shmatikov. How to steal reasoning without reasoning traces. *arXiv preprint arXiv:2603.07267*, 2026. URL https://arxiv.org/abs/2603.07267.

## A Related Work

**Theoretical Foundations of Model Extraction, Covert Learning, and Observational Defenses.** Our work builds on and extends the theoretical line of research studying the complexity of model extraction and learning from untrusted or defensive intermediaries (Angluin & Kharitonov, 1995; Canetti & Karchmer, 2021; Karchmer, 2023; Liu & Moitra, 2025). In classical learning theory, Angluin & Kharitonov (1995) initiated the study of cryptographic hardness for learning with membership queries, showing that under standard cryptographic assumptions, certain concept classes cannot be predicted in polynomial time even with noiseless query access. However, in the setting of Angluin & Kharitonov (1995), the anti-distillation defense is passive: the server returns exact labels ($\alpha = 0$) and relies entirely on the intrinsic cryptographic hardness of evaluating or inverting the concept class.

Closest to our work are the frameworks of *Covert Learning* by Canetti & Karchmer (2021) and *Observational Model Extraction Defense (OMED)* by Karchmer (2023). Canetti & Karchmer (2021) define covert learning as extracting a target function while keeping the query distribution computationally indistinguishable from honest draws from $\mathcal{D}$, and construct covert learners for decision trees and linear functions modulo a large integer $q$ when the server adds independent random noise. Karchmer (2023) studies OMED servers that passively inspect a batch of $n$ queries using a statistical detector $D : \mathcal{X}^n \to \{0, 1\}$ and return exact, uncorrupted labels whenever $D$ accepts. As detailed below in Section A.1, our framework advances beyond Canetti & Karchmer (2021); Karchmer (2023) in three fundamental respects: (i) we model *active output-perturbation defenses* $\mathfrak{S}(\alpha, \mathcal{D})$ that inject bounded noise $\alpha > 0$ on honest queries, and prove that for memoryless servers $\mathfrak{S}_{\text{mem}}(\alpha, \mathcal{D})$, Universal Distillation Insecurity ($\exists\mathcal{A}\forall\mathcal{D}\forall\mathcal{S}\forall h$) is equivalent to distribution-free *agnostic* PAC learning rather than noiseless PAC learning (Section 5); (ii) we show that combining active LPN noise with *stateful query caching* circumvents the stateless impossibility result of Karchmer (2023), yielding Fixed-Server Average-Case Anti-Distillation Security for hashed parities even though the class is efficiently learnable without noise (Section 6); and (iii) we construct an undetectable distillation attack for binary parities over $\mathbb{F}_2$ under the Planted XOR assumption (Christ & Gunn, 2024) that defeats *any* efficient stateful server injecting arbitrary adaptive noise (Section 7). More recently, Liu & Moitra (2025) gave polynomial-time model stealing algorithms for any low-rank language model from purely next-token sample access, further underscoring the vulnerability of structured generative models when no active or stateful defenses are deployed. Our framework also connects to broader complexity-theoretic studies of agnostic and distributional learning (Karchmer, 2024a;b) and cryptographic undetectability in machine learning (Goldwasser et al., 2022) (which studies planting undetectable backdoors in a learned hypothesis, dual to our setting where an untrusted client plants an undetectable algebraic trapdoor in a query batch).

**Model Distillation and Extraction Attacks in Practice.** Knowledge distillation was originally introduced by Hinton et al. (2015) to compress a large "teacher" network into a smaller "student" network using soft output probabilities. In the security domain, Tramèr et al. (2016) demonstrated that black-box prediction APIs returning confidence scores enable rapid parameter extraction, while subsequent works showed that extraction remains feasible using hard labels or sample generation alone (Alabdulmohsin et al., 2014; Chandrasekaran et al., 2020). For modern Large Language Models (LLMs), Carlini et al. (2024) extracted the final embedding projection layer of production models (such as GPT-3.5) via log-probability APIs, and numerous works have demonstrated that synthetic instruction-following data queried from frontier APIs can cheaply replicate capabilities in open-weight student models (Taori et al., 2023; Chiang et al., 2023; Xu et al., 2023). Even when providers conceal intermediate "chains of thought" to protect reasoning capabilities, recent attacks such as Trace Inversion (Zhang et al., 2026) synthesize high-fidelity reasoning traces solely from inputs, final answers, and brief summaries.

**Proactive Anti-Distillation and Output Perturbation Defenses.** To counter unauthorized distillation at inference time, proactive defenses perturb or post-process API outputs while preserving utility for legitimate users. Early defenses against DNN model stealing include Prediction Poisoning (Orekondy et al., 2020), which perturbs output posteriors to corrupt the student's training gradient, and calibrated proof-of-work schemes (Dziedzic et al., 2022) that raise the cost of large-scale querying. For LLMs, recent proactive defenses include Antidistillation Sampling (Savani et al., 2025), which perturbs next-token sampling distributions to poison reasoning traces for downstream student training; Defensive Output Generation (DOGe) (Li et al., 2025), which fine-tunes the teacher's output layer

with an adversarial anti-distillation objective; and information-theoretic logit post-processing (Fang et al., 2026), which minimizes the conditional mutual information between teacher logits and inputs given ground-truth labels. In our formalism, any inference-time perturbation scheme that processes each query independently without maintaining cross-query state corresponds to a memoryless server $\mathcal{S} \in \mathfrak{S}_{\text{mem}}(\alpha, \mathcal{D})$. Our equivalence in Section 5 provides a sharp theoretical lens on all such defenses: a stateless perturbation scheme can prevent distillation if and only if the induced corrupted distribution triggers a worst-case computational-to-statistical gap in agnostic PAC learning, and any stateless random perturbation can be denoised if an adversary can query the same inputs repeatedly.

**Reactive Defenses, Watermarking, and Radioactivity.** Reactive defenses aim to detect unauthorized distillation post hoc rather than prevent it during querying. While direct membership inference attacks are statistically unsound as standalone proofs of training data lineage (Zhang et al., 2024b), LLM watermarking schemes (Kirchenbauer et al., 2023; Christ et al., 2024) embed cryptographic or statistical signals into generated text. Sander et al. (2024) showed that watermarked API outputs exhibit "radioactivity," leaving detectable statistical traces in student models fine-tuned on them, and Xu et al. (2026) designed watermarks specifically optimized to survive distillation. However, recent theoretical and empirical attacks have shown that watermarks can be erased via random-walk perturbations (Zhang et al., 2024a) or stolen via query probing (Jovanovic et al., 2024), and that watermark radioactivity can be neutralized via pre-distillation paraphrasing or inference-time neutralization without degrading the distilled student's accuracy (Pan et al., 2025), motivating the study of provable proactive defenses that prevent model extraction outright.

### A.1 Comparison with Observational Defense Framework (Karchmer, 2023)

Our game-based formulation builds upon and substantially generalizes the *Observational Model Extraction Defense (OMED)* framework of Karchmer (2023) (and the Covert Learning framework of Canetti & Karchmer (2021)). In Karchmer (2023), a model is a deterministic Boolean function $f : \{0,1\}^d \to \{0,1\}$. An OMED is a stateless binary detector $D : (\{0,1\}^d)^n \to \{0,1\}$ that inspects a single batch of $n$ queries $X$: if $D(X) = 0$ (benign), the server returns the exact labels $f(X)$; if $D(X) = 1$ (adverse), it rejects the batch and returns $\perp$.

Our Distillation Game (Definitions 3.1–3.4) extends the OMED framework in three modeling directions and resolves two technical issues in Karchmer (2023):

1. **Active Output Perturbation (Noisy Servers) vs. Passive Observation:** Karchmer (2023) only considers passive defenses that either return exact labels ($\alpha = 0$) or reject the batch. However, many practical defenses (e.g., prediction poisoning and antidistillation sampling (Orekondy et al., 2020; Savani et al., 2025; Li et al., 2025; Fang et al., 2026)) actively perturb outputs while maintaining $\alpha$-accuracy. Theoretically, active label noise shifts the attacker's task from *realizable PAC learning* to the much harder regime of *agnostic PAC learning* (§5).
2. **Stateful Multi-Round Interaction vs. Single-Batch Game:** Whereas Karchmer (2023) studies a 1-round stateless game, Definition 3.1 allows $R$ adaptive rounds and a persistent server state $st_r \in \Sigma$.
3. **General Domains and Randomized Models:** Our formalism supports arbitrary spaces $\mathcal{X}, \mathcal{Y}$, general losses L, and randomized models $h : \mathcal{X} \to \Delta(\mathcal{Y})$ (such as autoregressive LLMs), rather than only deterministic Boolean functions.
4. **Resolving Quantifier Flaws and Redundant Constraints in Karchmer (2023):**
   - *Quantifier inversion and proof flaw:* Karchmer (2023, Lemma 3.6, Definition 3.8, and Theorem 3.9) place an existential quantifier over a hard subset $F \subseteq \mathcal{F}_n$ (of size $|F| \geq \eta|\mathcal{F}_n|$) *before* the universal quantifier over the client $\mathbf{C}$ ($\exists F \subseteq \mathcal{F}_n \; \forall \mathbf{C} \; \forall f \in F$). Taken literally, this statement is false whenever $F$ contains any efficiently computable function $f_0$, since a trivial client $\mathbf{C}_{f_0}$ that outputs $f_0$ extracts $f_0$ with probability 1. Specifically, the proof of Karchmer (2023, Lemma 3.6) errs in its first step when taking the contrapositive: negating $\exists F \; (|F| \geq \eta|\mathcal{F}_n|) \; \forall \mathbf{C} \; \forall f \in F$ only yields that for a $(1-\eta)$-fraction of $f \in \mathcal{F}_n$, there exists a *model-dependent* client $\mathbf{C}_f$ that extracts $f$. Instead, the proof illicitly swaps $\forall \mathbf{C}$ and $\exists F$, assuming a *single* client $\mathbf{C}$ that succeeds on a $(1-\eta)$-fraction of $f \in \mathcal{F}_n$ (which is the negation of $\forall \mathbf{C} \; \exists F_{\mathbf{C}}$, where $F_{\mathbf{C}}$ depends on $\mathbf{C}$).[2]

[2]Additionally, the proof of Karchmer (2023, Theorem 3.9) conflates the soundness guarantee $\mathbb{P}[\mathcal{M}(S) = \text{accept} \mid \text{extracted}] < \gamma$ with the reverse conditional probability $\mathbb{P}[\text{extracted} \mid \mathcal{M}(S) = \text{accept}] \leq \text{negl}(n)$ required by Karchmer (2023, Definition 3.8).

Our worst-case ($\forall\mathcal{A}\exists h$) and average-case ($\forall\mathcal{A}\mathbb{E}_{h\leftarrow\mu}$) definitions (Definitions 3.3 and 3.4) place the quantifiers in their mathematically sound order.

- *Redundant restriction to "$\mathcal{P}_n$-adverse" clients:* Karchmer (2023, Definition 3.7) restricts soundness to "$\mathcal{P}_n$-adverse" clients (whose query distribution is not $\mathcal{D}^{\otimes n}$). This restriction is redundant: any $\mathcal{P}_n$-benign client $\mathcal{A}$ can be perturbed into a $\mathcal{P}_n$-adverse client $\mathcal{A}_\delta$ that runs $\mathcal{A}$ with probability $1-\delta$ and queries a non-product distribution with tiny probability $\delta$, changing its extraction risk by at most $O(\delta)$. Moreover, real-world APIs must also defend against passive clients who distill models using honest i.i.d. queries. Our definitions therefore quantify over *all* computationally bounded adversaries $\mathcal{A}\in\mathcal{T}_A$.

Finally, we stress that no provably secure anti-distillation defense was previously known. Both Canetti & Karchmer (2021) and Karchmer (2023) only construct *attacks* (covert learning algorithms that bypass observational defenses) and prove impossibility results for passive OMEDs. In contrast, we establish the first positive anti-distillation defenses in Sections 5 and 6.

**Comparison with Prior Covert Learning Attacks (Canetti & Karchmer, 2021; Karchmer, 2023).** Our attack in Section 7 is an instance of "covert learning" (Canetti & Karchmer, 2021) (see also (Karchmer, 2023, Section 1.4)), but it strengthens prior constructions because the covert learner in Canetti & Karchmer (2021) assumes the server adds independent random noise (with fixed rate) to each query. By contrast, Theorem 7.2 breaks *any* computationally bounded server $\mathcal{S}\in\mathfrak{S}(\alpha,\mathcal{D})$ that is $\alpha$-accurate on honest queries, even if $\mathcal{S}$ is stateful and injects arbitrarily correlated, adaptive noise across the query batch.

## B High-Probability Definitions

The definitions in Section 3 bound the expected error. We could instead require high probability bounds. This requires adding extra parameters. We state these alternate definitions here for completeness.

**Definition B.1** ($(\alpha,\beta)$-Accuracy for Honest Queries)**.** *An* honest client $\mathcal{A}_{honest}$ *is a client participating in the Distill*$(\mathcal{A}_{honest},\mathcal{S},h,\mathcal{D},n,R)$ *game such that, in every round $r\in[R]$, $\mathcal{A}_{honest}$ constructs its query batch $X_r=(x_{r,1},\dots,x_{r,n_r})$ by sampling each $x_{r,j}$ independently from $\mathcal{D}$, irrespective of any prior rounds.*

*A server $\mathcal{S}$ is* $(\alpha,\beta)$-accurate with loss $\mathrm{L}$ for $n$ queries in $R$ rounds with respect to model $h$ and distribution $\mathcal{D}$ *if, for any honest client $\mathcal{A}_{honest}$, we have*

$$\mathop{\mathbb{P}}_{(X,Y,\hat{h})\leftarrow \textit{Distill}(\mathcal{A}_{honest},\mathcal{S},h,\mathcal{D},n,R)}\left[\sum_{r=1}^{R}\sum_{i=1}^{n_r}\mathrm{L}\left(P_{r,i}^{\mathcal{S},h}(\cdot\mid\pi_{r,i}),h(x_{r,i})\right)\le n\alpha\right]\ge 1-\beta, \tag{B.1}$$

*where $P_{r,i}^{\mathcal{S},h}(\cdot\mid\pi_{r,i})\in\Delta(\mathcal{Y})$ is the conditional distribution of $y_{r,i}$ given $\pi_{r,i}=(X_{\le r},Y_{<r},y_{r,<i})$.*

**Definition B.2** ($(\gamma,\delta)$-Anti-Distillation Security)**.** *A server $\mathcal{S}$ provides* $(\gamma,\delta)$-anti-distillation security against computational bound $\mathcal{T}_A$ with loss $\mathrm{L}$ for $n$ queries in $R$ rounds with respect to model class $\mathcal{H}$ and distribution $\mathcal{D}$ *if, for all adversaries $\mathcal{A}$ operating within computational bound $\mathcal{T}_A$, the loss of the model $\hat{h}$ produced by $\mathcal{A}$ in the distillation game is at least $\gamma$ with probability at least $1-\delta$. Formally,*

$$\forall\mathcal{A}\in\mathcal{T}_A\ \exists h\in\mathcal{H}\quad\mathop{\mathbb{P}}_{(X,Y,\hat{h})\leftarrow \textit{Distill}(\mathcal{A},\mathcal{S},h,\mathcal{D},n,R)}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{L}\left(\hat{h}(x),h(x)\right)\right]\ge\gamma\right]\ge 1-\delta. \tag{B.2}$$

Our information-theoretic distillation result (Theorem 4.1) can be translated to these definitions:

**Theorem B.3.** *Let $\mathcal{S}$ be a server that is $(\alpha,\beta)$-accurate with loss* TV *for $n$ queries in $R$ rounds with respect to model $h$ and distribution $\mathcal{D}$ for any $h\in\mathcal{H}$. Further suppose $\mathcal{S}$ provides $(\gamma,\delta)$-anti-distillation security against computational bound $\mathcal{T}_A$ with loss* TV *for $n$ queries in $R$ rounds with respect to model class $\mathcal{H}$ and distribution $\mathcal{D}$. If $\mathcal{T}_A$ represents all adversaries (i.e., computationally unbounded), then either $\gamma\le 2\left(\alpha+\sqrt{\frac{2}{n}\log(|\mathcal{H}|^2/\beta)}\right)$ or $\delta\ge 1-2\beta$ (or both).*

*Proof.* Consider the honest-query Scheffé estimator $\mathcal{A}$ from the proof of Theorem 4.1 (Appendix C) over the family $\mathcal{T}=\{T_{h_1,h_2}:h_1,h_2\in\mathcal{H},h_1\ne h_2\}$ of at most $|\mathcal{H}|(|\mathcal{H}|-1)/2\le|\mathcal{H}|^2/2$ Scheffé

sets. For each step $(r, i)$ and any $T \in \mathcal{T}$, decompose the empirical deviation as

$$\begin{aligned}
&\mathbb{I}[(x_{r,i}, y_{r,i}) \in T] - \mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes h}[(x,y) \in T] \\
&= \Big(\mathbb{I}[(x_{r,i}, y_{r,i}) \in T] - \mathop{\mathbb{P}}_{y\leftarrow P_{r,i}^{\mathcal{S},h}(\cdot|\pi_{r,i})}[(x_{r,i}, y) \in T]\Big) \\
&\quad + \Big(\mathop{\mathbb{P}}_{y\leftarrow P_{r,i}^{\mathcal{S},h}(\cdot|\pi_{r,i})}[(x_{r,i}, y) \in T] - \mathop{\mathbb{P}}_{y\leftarrow h(x_{r,i})}[(x_{r,i}, y) \in T]\Big) \\
&\quad + \Big(\mathop{\mathbb{P}}_{y\leftarrow h(x_{r,i})}[(x_{r,i}, y) \in T] - \mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes h}[(x,y) \in T]\Big).
\end{aligned}$$

The first term is a martingale difference sequence in $[-1, 1]$ of conditional range 1 given $\pi_{r,i}$; the third term is an i.i.d. sequence in $[-1, 1]$ of range 1; and the second term has absolute value bounded by $\mathrm{TV}\left(P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}), h(x_{r,i})\right)$. Applying the Azuma–Hoeffding inequality to the first and third sums (each with failure probability $\beta/2$ via a union bound over $\mathcal{T}$) shows that with probability at least $1 - \beta$,

$$\begin{aligned}
&\max_{T\in\mathcal{T}} \left| \frac{1}{n} \sum_{r=1}^{R} \sum_{i=1}^{n_r} \mathbb{I}[(x_{r,i}, y_{r,i}) \in T] - \mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes h}[(x,y) \in T] \right| \\
&\quad \le \frac{1}{n} \sum_{r=1}^{R} \sum_{i=1}^{n_r} \mathrm{TV}\left(P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}), h(x_{r,i})\right) + \sqrt{\frac{2}{n} \log \frac{|\mathcal{H}|^2}{\beta}}.
\end{aligned}$$

Intersecting this event with the $(\alpha, \beta)$-accuracy event $\sum_{r,i} \mathrm{TV}\left(P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}), h(x_{r,i})\right) \le n\alpha$ (which holds with probability at least $1 - \beta$) via a union bound, both hold simultaneously with probability at least $1 - 2\beta$. On this intersection event, Lemma C.2 guarantees $\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{TV}\left(\hat{h}(x), h(x)\right)\right] = \mathrm{TV}\left(\mathcal{D}\otimes\hat{h}, \mathcal{D}\otimes h\right) \le 2\big(\alpha + \sqrt{\tfrac{2}{n}\log(|\mathcal{H}|^2/\beta)}\big)$. Thus, if $\gamma > 2\big(\alpha + \sqrt{\tfrac{2}{n}\log(|\mathcal{H}|^2/\beta)}\big)$, then $\mathbb{P}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{TV}\left(\hat{h}(x), h(x)\right)\right] \ge \gamma\right] \le 2\beta$, which implies $1 - \delta \le 2\beta$, i.e., $\delta \ge 1 - 2\beta$. □

## C Proof of the Information-Theoretic Distillation Result

**Theorem C.1** (Restatement of Theorem 4.1)**.** *For any finite model class $\mathcal{H}$, query parameters $(n, R)$, and accuracy parameter $\alpha \ge 0$, the family of all servers $\mathfrak{S}$ (with loss* TV*) is* universally $\gamma$-distillation insecure *(Definition 3.4) against computationally unbounded adversaries for*

$$\gamma = 2\left(\alpha + \sqrt{\frac{1}{n}\log|\mathcal{H}|}\right). \tag{C.1}$$

*Proof of Theorem 4.1.* Let $\mathcal{D}$ be any distribution on $\mathcal{X}$ and let $\mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D})$ be any server that is $\alpha$-accurate with loss TV for $n$ queries in $R$ rounds with respect to every $h \in \mathcal{H}$ and $\mathcal{D}$. Let $\mathcal{A}$ be an honest-query client that generates $n$ queries $X \in \mathcal{X}^n$ drawn i.i.d. from $\mathcal{D}$ across the $R$ rounds, receives the corresponding server responses $Y \in \mathcal{Y}^n$ in $(X, Y, \hat{h}) \leftarrow \mathrm{Distill}(\mathcal{A}, \mathcal{S}, h, \mathcal{D}, n, R)$, and then applies the minimum-distance Scheffé estimator defined below. Let $X \times Y \in (\mathcal{X}\times\mathcal{Y})^n$ denote the observed query/response pairs, and let $X \otimes h \in (\mathcal{X}\times\mathcal{Y})^n$ denote an idealized transcript sharing the same queries $X$ where each clean response $\tilde{y}_{r,i} \leftarrow h(x_{r,i})$ is drawn independently according to $h$.

The $\alpha$-accuracy guarantee of $\mathcal{S}$ (Definition 3.2) ensures that we can sequentially couple $X \times Y$ with $X \otimes h$ such that $\mathbb{E}\left[\|X \times Y - X \otimes h\|_0\right] \le \alpha n$. Specifically, for each step $(r, i)$ in lexicographical order, condition on the transcript prefix $\pi_{r,i} = (X_{\le r}, Y_{<r}, y_{r,<i})$ (together with the prior clean responses $\tilde{Y}_{<r}, \tilde{y}_{r,<i}$). Under this conditioning, the server's next output $y_{r,i}$ has conditional distribution $P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}) \in \Delta(\mathcal{Y})$, while the clean response $\tilde{y}_{r,i}$ has distribution $h(x_{r,i}) \in \Delta(\mathcal{Y})$. By the optimal coupling theorem for total variation distance, there exists a conditional coupling of $(y_{r,i}, \tilde{y}_{r,i})$ given $\pi_{r,i}$ such that $\tilde{y}_{r,i} \leftarrow h(x_{r,i})$ is conditionally independent of the past given $x_{r,i}$ and

$$\mathbb{P}\left[y_{r,i} \ne \tilde{y}_{r,i} \mid \pi_{r,i}\right] = \mathrm{TV}\left(P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}), h(x_{r,i})\right). \tag{C.2}$$

Composing these conditional couplings coordinate-by-coordinate via the chain rule and taking expectations over the full transcript yields

$$\mathbb{E}\left[\|X \times Y - X \otimes h\|_0\right] = \mathbb{E}\left[\sum_{r=1}^{R}\sum_{i=1}^{n_r} \mathrm{TV}\left(P_{r,i}^{\mathcal{S},h}(\cdot \mid \pi_{r,i}), h(x_{r,i})\right)\right] \le \alpha n. \tag{C.3}$$

Note that this coupling is purely analytical—the attacker $\mathcal{A}$ only observes $X \times Y$ and does not need to construct the coupling.

Now our goal is to prove that, given $X \times Y$, $\mathcal{A}$ can recover $\hat{h} \approx h$ for any $h \in \mathcal{H}$. In the case where $h$ is a deterministic function, this can be proved using standard concentration inequalities by letting $\hat{h} \in \mathcal{H}$ be the empirical risk minimizer i.e. $\hat{h} = \arg\min_{h'} \|X \times Y - X \otimes h'\|_0$. We consider the possibility that $h$ is randomized, which adds some complexity, but does not fundamentally alter the result.

This task boils down to what is known as *density estimation* or *hypothesis selection*, with the added requirement that we must be robust to $\alpha n$ corruptions. Fortunately, this is a well-studied problem (Devroye & Lugosi, 2001; Mahalanabis & Stefankovic, 2008; Bousquet et al., 2019; Daskalakis et al., 2012; Bun et al., 2019). The key algorithm, which we describe next, is known as a minimum distance estimate over Scheffé sets.

We begin by combining the accuracy guarantee $\mathbb{E}\left[\|X \times Y - X \otimes h\|_0\right] \le \alpha n$ with a concentration bound and the triangle inequality: Let $\mathcal{T}$ be a finite collection of subsets of $\mathcal{X} \times \mathcal{Y}$, and index the $n$ coupled query-response pairs as $(x_i, y_i)$ and $(x_i, \tilde{y}_i)$ for $i \in [n]$ (where $\tilde{y}_i \leftarrow h(x_i)$ is the coupled clean response). Then

$$\begin{aligned}
&\mathbb{E}\left[\max_{T\in\mathcal{T}} \left| \mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes h}[(x,y)\in T] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[(x_i,y_i)\in T]\right|\right] && \text{(C.4)}\\
&\le \mathbb{E}\left[\begin{array}{l}\max_{T\in\mathcal{T}}\left|\mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes h}[(x,y)\in T] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[(x_i,\tilde{y}_i)\in T]\right| \\ \quad + \max_{\bar{T}\subset\mathcal{X}\times\mathcal{Y}}\left|\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[(x_i,\tilde{y}_i)\in \bar{T}] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[(x_i,y_i)\in \bar{T}]\right|\end{array}\right] \\
&\qquad\qquad \text{(triangle inequality)}\\
&= \mathbb{E}\left[\max_{T\in\mathcal{T}} \left| \mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes h}[(x,y)\in T] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[(x_i,\tilde{y}_i)\in T]\right|\right] + \mathbb{E}\left[\frac{1}{n}\|X\times Y - X\otimes h\|_0\right]\\
&\le \sqrt{\frac{\log(2|\mathcal{T}|)}{2n}} + \alpha. && \text{(Hoeffding's inequality; Lemma C.3)}
\end{aligned}$$

This is the underlying guarantee that we will use to establish our result. It only remains to determine the appropriate sets $\mathcal{T}$ and then to use this information to select a single model.

Fix an arbitrary total ordering $\prec$ on $\mathcal{H}$. For any distinct hypotheses $h_1 \prec h_2$ in $\mathcal{H}$, define $T_{h_1,h_2} := \{(x,y) \in \mathcal{X}\times\mathcal{Y} : \mathbb{P}[h_1(x) = y] > \mathbb{P}[h_2(x) = y]\}$ and define $T_{h_2,h_1} := (\mathcal{X}\times\mathcal{Y}) \setminus T_{h_1,h_2} = \{(x,y) \in \mathcal{X}\times\mathcal{Y} : \mathbb{P}[h_2(x) = y] \ge \mathbb{P}[h_1(x) = y]\}$ (with $T_{h_1,h_1} := \emptyset$). By construction, $T_{h_2,h_1}$ is the exact complement of $T_{h_1,h_2}$, and $\mathrm{TV}(\mathcal{D}\otimes h_1, \mathcal{D}\otimes h_2) = |(\mathcal{D}\otimes h_1)(T_{h_1,h_2}) - (\mathcal{D}\otimes h_2)(T_{h_1,h_2})|$ for all $h_1, h_2 \in \mathcal{H}$. Since $|(\mathcal{D}\otimes h)(T_{h_2,h_1}) - S(T_{h_2,h_1})| = |(\mathcal{D}\otimes h)(T_{h_1,h_2}) - S(T_{h_1,h_2})|$, it suffices to control the $|\mathcal{H}|(|\mathcal{H}|-1)/2$ sets $\mathcal{T} = \{T_{h_1,h_2} : h_1 \prec h_2 \in \mathcal{H}\}$ (the Scheffé sets or Yatracos system (Yatracos, 1985)). Denoting the random variable inside the expectation of (C.4) by $A$, we have

$$\forall h_1, h_2 \in \mathcal{H} \quad \left| \mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes h}[(x,y)\in T_{h_1,h_2}] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[(x_i,y_i)\in T_{h_1,h_2}]\right| \le A, \tag{C.5}$$

where $\mathbb{E}[A] \le \alpha + \sqrt{\frac{\log(2|\mathcal{T}|)}{2n}} \le \alpha + \sqrt{\frac{1}{n}\log|\mathcal{H}|}$.

Now we ask: which hypothesis in $\mathcal{H}$ is closest to $h$? Let $\hat{h} \in \mathcal{H}$ be the minimizer of $f(\hat{h})$, where

$$f(\hat{h}) := \max_{h'\in\mathcal{H}} \left| \mathop{\mathbb{P}}_{(x,y)\leftarrow\mathcal{D}\otimes\hat{h}}\left[(x,y)\in T_{\hat{h},h'}\right] - \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[(x_i,y_i)\in T_{\hat{h},h'}]\right| \le A. \tag{C.6}$$

By Lemma C.2 with $\{P_1, \cdots, P_m\} = \{\mathcal{D} \otimes h' : h' \in \mathcal{H}\}$, $R = \mathcal{D} \otimes h$, and $S = X \times Y$, we have pointwise for every transcript:

$$\mathop{\mathbb{E}}_{x \leftarrow \mathcal{D}} \left[ \mathrm{TV}\left( \hat{h}(x), h(x) \right) \right] = \mathrm{TV}\left( \mathcal{D} \otimes \hat{h}, \mathcal{D} \otimes h \right) \leq 3 \min_{h' \in \mathcal{H}} \mathrm{TV}\left( \mathcal{D} \otimes h', \mathcal{D} \otimes h \right) + 2A = 2A, \tag{C.7}$$

as $h \in \mathcal{H}$. Taking expectations over the transcript yields $\mathrm{Risk}(\mathcal{A}, \mathcal{S}, h, \mathcal{D}, n, R) \leq 2\mathbb{E}[A] \leq 2\left(\alpha + \sqrt{\frac{1}{n} \log |\mathcal{H}|}\right)$, so $\mathcal{A}$ achieves the claimed accuracy.

We remark that finding $\hat{h}$ requires enumerating $\mathcal{H}$ and evaluating $f$ (C.6) which also requires enumerating $\mathcal{H}$, so the total runtime is $O(|\mathcal{H}|^2)$. (This could be improved to $O(|\mathcal{H}|)$ (Mahalanabis & Stefankovic, 2008).) This is impractical for any reasonable model class $\mathcal{H}$, since $|\mathcal{H}|$ is exponential in the description length of the models. □

**Lemma C.2** ((Devroye & Lugosi, 2001, Theorem 6.3) (Mahalanabis & Stefankovic, 2008, Theorem 4))**.** *Let $P_1, P_2, \cdots, P_m, R, S$ be probability distributions on $\Omega$. For $1 \leq i < j \leq m$, let $T_{i,j} = \{\omega \in \Omega : P_i(\omega) > P_j(\omega)\}$, let $T_{j,i} = \Omega \setminus T_{i,j} = \{\omega \in \Omega : P_j(\omega) \geq P_i(\omega)\}$, and let $T_{i,i} = \emptyset$. Let $k \in [m]$ be the minimizer of $\max_{j \in [m]} |P_k(T_{k,j}) - S(T_{k,j})|$. Then*

$$\mathrm{TV}(P_k, R) \leq 3 \min_{i \in [m]} \mathrm{TV}(P_i, R) + 2 \max_{i,j \in [m]} |R(T_{i,j}) - S(T_{i,j})|. \tag{C.8}$$

*Proof.* Let $a = \max_{i,j \in [m]} |R(T_{i,j}) - S(T_{i,j})|$ and $b = \min_{i \in [m]} \mathrm{TV}(P_i, R)$. Let $\ell \in [m]$ be the minimizer of $\mathrm{TV}(P_\ell, R)$. We have

$$\begin{aligned}
\mathrm{TV}(P_k, R) &\leq \mathrm{TV}(P_k, P_\ell) + \mathrm{TV}(P_\ell, R) && \text{(triangle inequality)} \\
&= |P_k(T_{k,\ell}) - P_\ell(T_{k,\ell})| + b && \text{(by construction of } T_{k,\ell}) \\
&\leq |P_k(T_{k,\ell}) - S(T_{k,\ell})| + |P_\ell(T_{\ell,k}) - S(T_{\ell,k})| + b \\
& && \text{(triangle inequality \& } T_{\ell,k} = \Omega \setminus T_{k,\ell}) \\
&\leq \max_{j \in [m]} |P_k(T_{k,j}) - S(T_{k,j})| + \max_{j \in [m]} |P_\ell(T_{\ell,j}) - S(T_{\ell,j})| + b \\
&\leq 2 \max_{j \in [m]} |P_\ell(T_{\ell,j}) - S(T_{\ell,j})| + b && \text{(minimality of } k) \\
&\leq 2 \max_{j \in [m]} \left( \mathrm{TV}(P_\ell, R) + |R(T_{\ell,j}) - S(T_{\ell,j})| \right) + b && \text{(triangle inequality)} \\
&\leq 2(b + a) + b \\
&= 3 \min_{i \in [m]} \mathrm{TV}(P_i, R) + 2 \max_{i,j \in [m]} |R(T_{i,j}) - S(T_{i,j})|,
\end{aligned}$$

as required. □

**Lemma C.3** (In-expectation version of Hoeffding's inequality with union bound)**.** *For all $i \in [n]$ and $j \in [m]$, let $X_{i,j} \in [0,1]$ be a random variable (independent from all $X_{i',j}$ with $i' \neq i$) and let $\mu_{i,j} = \mathbb{E}[X_{i,j}]$. For all $j \in [m]$, define $X_j = \sum_{i \in [n]} X_{i,j}$ and $\mu_j = \mathbb{E}[X_j] = \sum_{i \in [n]} \mu_{i,j}$. Let $Y = \max_{j \in [m]} \frac{1}{n} |X_j - \mu_j|$. Then $\mathbb{E}[Y] \leq \sqrt{\frac{\log(2m)}{2n}}$.*

*Proof.* Let $t > 0$ be determined later. We have

$$\begin{aligned}
\exp(t\mathbb{E}[Y]) &\le \mathbb{E}[\exp(tY)] && \text{(Jensen's inequality)}\\
&= \mathbb{E}\left[\max_{j\in[m]} \exp\left(\frac{t}{n}|X_j - \mu_j|\right)\right]\\
&\le \mathbb{E}\left[\sum_{j\in[m]} \exp\left(\frac{t}{n}(X_j - \mu_j)\right) + \exp\left(-\frac{t}{n}(X_j - \mu_j)\right)\right] && \text{(max} \le \text{sum)}\\
&= \sum_{j\in[m]} \mathbb{E}\left[\prod_{i\in[n]} \exp\left(\frac{t}{n}(X_{i,j} - \mu_{i,j})\right)\right] + \mathbb{E}\left[\prod_{i\in[n]} \exp\left(-\frac{t}{n}(X_{i,j} - \mu_{i,j})\right)\right]\\
&= \sum_{j\in[m]} \prod_{i\in[n]} \mathbb{E}\left[\exp\left(\frac{t}{n}(X_{i,j} - \mu_{i,j})\right)\right] + \prod_{i\in[n]} \mathbb{E}\left[\exp\left(-\frac{t}{n}(X_{i,j} - \mu_{i,j})\right)\right]\\
& && \text{(independence)}\\
&\le \sum_{j\in[m]} \prod_{i\in[n]} \exp\left(\frac{t^2}{8n^2}\right) + \prod_{i\in[n]} \exp\left(\frac{t^2}{8n^2}\right) && \text{(Hoeffding's lemma)}\\
&= 2m\exp(t^2/8n).
\end{aligned}$$

This rearranges to

$$\mathbb{E}[Y] \le \frac{1}{t}\log(2m\exp(t^2/8n)) = \frac{\log(2m)}{t} + \frac{t}{8n}. \tag{C.9}$$

Setting $t = \sqrt{8n\log(2m)}$ yields the result. □

# D Proofs from Section 5 – Equivalence of Memoryless Distillation and Agnostic PAC Learning

**Theorem D.1** (Restatement of Theorem 5.2)**.** *Let $\mathcal{H}$ be any hypothesis class, and let $\mathrm{L} : \mathcal{Y}\times\mathcal{Y} \to [0,1]$ be a metric. If there is an efficient $(\alpha,\gamma)$-Gap PAC learner $\mathcal{A}_{PAC}$ for $\mathcal{H}$ with respect to loss $\mathrm{L}$, then $\mathfrak{S}_{mem}$ is* universally $(\gamma+\alpha)$-distillation insecure *against efficient adversaries (Definition 3.4): there exists an efficient distillation adversary $\mathcal{A}_{Distill} \in \mathcal{T}_{eff}$ that breaks the $(\gamma+\alpha)$-anti-distillation security of every $\alpha$-accurate memoryless server $\mathcal{S} \in \mathfrak{S}_{mem}(\alpha,\mathcal{D})$ for every distribution $\mathcal{D}$ (achieving $\mathrm{Risk}(\mathcal{A}_{Distill},\mathcal{S},h,\mathcal{D}) \le \gamma+\alpha$ for all $h\in\mathcal{H}$).*

*Proof of Theorem 5.2.* Let $n$ denote the sample complexity of $\mathcal{A}_{\text{PAC}}$. We construct the universal distillation adversary $\mathcal{A}_{\text{Distill}}$ as follows:

1. Sample $n$ queries $x_1,\dots,x_n$ i.i.d. from the marginal distribution $\mathcal{D}$.
2. Send the queries to the memoryless server $\mathcal{S}$ and receive responses $y_1 \leftarrow \mathcal{S}(x_1),\dots,y_n \leftarrow \mathcal{S}(x_n)$.
3. Feed the labeled pairs $S_n = \{(x_1,y_1),\dots,(x_n,y_n)\}$ to $\mathcal{A}_{\text{PAC}}$ and output its hypothesis $\hat{h}$.

Efficiency of $\mathcal{A}_{\text{Distill}}$ follows immediately from that of $\mathcal{A}_{\text{PAC}}$ (and $\mathcal{A}_{\text{Distill}}$ does not depend on $\mathcal{D}$, $\mathcal{S}$, or $h$).

To prove the adversary's distillation guarantee for any distribution $\mathcal{D}$, any server $\mathcal{S} \in \mathfrak{S}_{\text{mem}}(\alpha,\mathcal{D})$, and any target $h\in\mathcal{H}$, let $P$ be the joint distribution over $\mathcal{X}\times\mathcal{Y}$ defined by drawing $x \leftarrow \mathcal{D}$ and conditionally sampling $y \leftarrow \mathcal{S}(x)$. The pairs in $S_n$ are $n$ i.i.d. samples from $P$ by the memorylessness of $\mathcal{S}$.

Because $\mathcal{S} \in \mathfrak{S}_{\text{mem}}(\alpha,\mathcal{D})$ is $\alpha$-accurate with loss L for the true target $h\in\mathcal{H}$, we know by symmetry of L that:

$$L_P(h) = \mathop{\mathbb{E}}_{(x,y)\leftarrow P}[\mathrm{L}(h(x),y)] = \mathop{\mathbb{E}}_{x\leftarrow\mathcal{D},\, y\leftarrow\mathcal{S}(x)}[\mathrm{L}(y,h(x))] \le \alpha. \tag{D.1}$$

Consequently, the promise of the gap problem is satisfied for $P$ with $h^* = h$. By the guarantee of the Gap PAC learner $\mathcal{A}_{\text{PAC}}$, the output hypothesis $\hat{h}$ satisfies:

$$\mathbb{E}\left[L_P(\hat{h})\right] = \mathbb{E}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D},\, y\leftarrow\mathcal{S}(x)}\left[\mathrm{L}\left(\hat{h}(x), y\right)\right]\right] \leq \gamma. \tag{D.2}$$

Using the triangle inequality for L, we bound the expected risk of $\hat{h}$ with respect to the true target $h$:

$$\begin{aligned}\mathrm{Risk}(\mathcal{A}_{\text{Distill}}, \mathcal{S}, h, \mathcal{D}) &= \mathbb{E}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{L}\left(\hat{h}(x), h(x)\right)\right]\right] \\ &\leq \mathbb{E}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D},\, y\leftarrow\mathcal{S}(x)}\left[\mathrm{L}\left(\hat{h}(x), y\right) + \mathrm{L}\left(y, h(x)\right)\right]\right] \leq \gamma + \alpha.\end{aligned}$$

Since this holds for every distribution $\mathcal{D}$, every $\mathcal{S} \in \mathfrak{S}_{\text{mem}}(\alpha, \mathcal{D})$, and every $h \in \mathcal{H}$, $\mathfrak{S}_{\text{mem}}$ is universally $(\gamma+\alpha)$-distillation insecure. □

**Theorem D.2** (Restatement of Theorem 5.3)**.** *Suppose $\mathfrak{S}_{mem}$ is* universally $\gamma$-distillation insecure *against efficient adversaries (Definition 3.4)—i.e., there exists an efficient distillation adversary $\mathcal{A}_{Distill} \in \mathcal{T}_{eff}$ that achieves $\mathrm{Risk}(\mathcal{A}_{Distill}, \mathcal{S}, h, \mathcal{D}) \leq \gamma$ for every distribution $\mathcal{D}$, every $\alpha$-accurate memoryless server $\mathcal{S} \in \mathfrak{S}_{mem}(\alpha, \mathcal{D})$, and every $h \in \mathcal{H}$. Then, for any $\varepsilon \geq \frac{1}{d^{O(1)}}$, there exists an efficient $(\alpha-\varepsilon, \gamma+\alpha+\varepsilon)$-Gap PAC learner $\mathcal{A}_{PAC}$ for $\mathcal{H}$.*

*Proof of Theorem 5.3.* First, observe that since $\mathcal{A}_{\text{Distill}} \in \mathcal{T}_{\text{eff}}$ runs in time $d^{O(1)}$ (independent of the dataset $S_m$), the total number of bits $M$ in its output representation (circuit/description of $\hat{h}$) across all possible inputs, server transcripts, and random tapes is at most $d^{O(1)}$. Thus, the global set $\mathcal{H}_{\text{Distill}}$ of all possible distinct output hypotheses that $\mathcal{A}_{\text{Distill}}$ can ever output is fixed independently of $S_m$ and has cardinality $|\mathcal{H}_{\text{Distill}}| \leq 2^M$. Let $m = \left\lceil \frac{9}{\varepsilon^2}\left(M \ln 2 + \ln(3/\varepsilon)\right)\right\rceil \leq d^{O(1)}$.

Let $P$ be an arbitrary distribution over $\mathcal{X} \times \mathcal{Y}$, and suppose there exists $h^* \in \mathcal{H}$ such that $L_P(h^*) \leq \alpha - \varepsilon$. We construct the Gap PAC learner $\mathcal{A}_{\text{PAC}}$ as follows:

1. Receive an i.i.d. sample dataset $S_m = \{(x_1, y_1), \ldots, (x_m, y_m)\} \leftarrow P^m$.
2. Define $\mathcal{D}$ to be the empirical uniform distribution over the multiset $X_m = \{x_1, \ldots, x_m\}$ (i.i.d. sampling $x \leftarrow \mathcal{D}$ draws $i \leftarrow \mathrm{Uniform}([m])$ and outputs $x_i$).
3. Define a randomized memoryless server $\mathcal{S}_{\text{sim}} : \mathcal{X} \to \Delta(\mathcal{Y})$ which, on a query $x \in \mathcal{X}$, returns:
   - If $x \notin X_m$, output an arbitrary fixed value $y_0 \in \mathcal{Y}$.
   - If $x \in X_m$, pick an index $i \in \{j \in [m] : x_j = x\}$ uniformly at random and output $y_i$.
4. Run $\mathcal{A}_{\text{Distill}}$ on the distribution $\mathcal{D}$ with all queries answered by $\mathcal{S}_{\text{sim}}$, and output the resulting hypothesis $\hat{h}$.

Since $m \leq d^{O(1)}$ and $\mathcal{A}_{\text{Distill}}$ is efficient, $\mathcal{A}_{\text{PAC}}$ runs in $d^{O(1)}$ time and is therefore efficient.

To analyze $\mathcal{A}_{\text{PAC}}$'s learning guarantee, let $\widehat{L}(h) := \frac{1}{m}\sum_{i=1}^{m} \mathrm{L}\left(h(x_i), y_i\right)$ denote the empirical risk on $S_m$. Note that drawing $x \leftarrow \mathcal{D}$ and $y \leftarrow \mathcal{S}_{\text{sim}}(x)$ is identically distributed to drawing an index $i \leftarrow \mathrm{Uniform}([m])$ and outputting $(x_i, y_i)$. Consequently, for every hypothesis $h$, the expected loss of $\mathcal{S}_{\text{sim}}$ relative to $h$ on $\mathcal{D}$ is exactly the empirical risk on $S_m$:

$$\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D},\, y\leftarrow\mathcal{S}_{\text{sim}}(x)}\left[\mathrm{L}\left(y, h(x)\right)\right] = \frac{1}{m}\sum_{i=1}^{m} \mathrm{L}\left(y_i, h(x_i)\right) = \widehat{L}(h). \tag{D.3}$$

Because $L_P(h^*) \leq \alpha - \varepsilon \leq \alpha - \varepsilon/3$ and $\mathrm{L} \in [0,1]$, Hoeffding's inequality guarantees that with probability at least $1 - \exp(-2m(\varepsilon/3)^2) \geq 1 - \varepsilon/3$ over $S_m \leftarrow P^m$, the event $\mathcal{E} := \{\widehat{L}(h^*) \leq \alpha\}$ holds.

For any realization of $S_m$ for which $\mathcal{E}$ holds, (D.3) shows that $\mathcal{S}_{\text{sim}}$ is $\alpha$-accurate on $\mathcal{D}$ with respect to the target $h^*$. (Formally, if one requires a server in $\mathfrak{S}_{\text{mem}}(\alpha, \mathcal{D})$ to be $\alpha$-accurate for all $h \in \mathcal{H}$, we

may view $\mathcal{S}_{\text{sim}}$ in the Distillation Game as the server $\mathcal{S}(x,h)$ that executes $\mathcal{S}_{\text{sim}}(x)$ when $h = h^*$ and outputs $h(x)$ when $h \neq h^*$; then $\mathcal{S} \in \mathfrak{S}_{\text{mem}}(\alpha, \mathcal{D})$ is $\alpha$-accurate for every $h \in \mathcal{H}$, and $\mathcal{A}_{\text{Distill}}$'s view on target $h^*$ is identical to interacting with $\mathcal{S}_{\text{sim}}$.) Therefore, by the universal $\gamma$-distillation insecurity guarantee of $\mathcal{A}_{\text{Distill}}$, for every dataset $S_m \in \mathcal{E}$, the output $\hat{h}$ of $\mathcal{A}_{\text{Distill}}$ satisfies

$$\mathop{\mathbb{E}}_{\mathcal{A}_{\text{Distill}}}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{L}\left(\hat{h}(x), h^*(x)\right)\right] \,\middle|\, S_m\right] \leq \gamma,$$
$$\text{and hence} \quad \mathbb{E}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{L}\left(\hat{h}(x), h^*(x)\right)\right] \,\middle|\, \mathcal{E}\right] \leq \gamma.$$

By the triangle inequality and (D.3), conditioned on $\mathcal{E}$, the expected empirical risk of $\hat{h}$ satisfies:

$$\begin{aligned}
\mathbb{E}\left[\widehat{L}(\hat{h}) \mid \mathcal{E}\right] &= \mathbb{E}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D},\, y\leftarrow\mathcal{S}_{\text{sim}}(x)}\left[\mathrm{L}\left(\hat{h}(x), y\right)\right] \,\middle|\, \mathcal{E}\right] \\
&\leq \mathbb{E}\left[\mathop{\mathbb{E}}_{x\leftarrow\mathcal{D}}\left[\mathrm{L}\left(\hat{h}(x), h^*(x)\right)\right] + \mathop{\mathbb{E}}_{x\leftarrow\mathcal{D},\, y\leftarrow\mathcal{S}_{\text{sim}}(x)}\left[\mathrm{L}\left(h^*(x), y\right)\right] \,\middle|\, \mathcal{E}\right] \\
&\leq \gamma + \alpha.
\end{aligned}$$

Finally, we relate $\widehat{L}(\hat{h})$ to the population risk $L_P(\hat{h})$. Since $\mathcal{H}_{\text{Distill}}$ is fixed prior to drawing $S_m$, Hoeffding's inequality and a union bound over the at most $2^M$ hypotheses in $\mathcal{H}_{\text{Distill}}$ ensure that with probability at least $1 - 2^M \exp(-2m(\varepsilon/3)^2) \geq 1 - \varepsilon/3$ over $S_m \leftarrow P^m$, the uniform convergence event $\mathcal{U} := \left\{\forall h \in \mathcal{H}_{\text{Distill}},\ L_P(h) \leq \widehat{L}(h) + \varepsilon/3\right\}$ holds. Using indicator random variables and the non-negativity of $\widehat{L}(\hat{h}) \geq 0$ (so that $\widehat{L}(\hat{h})\mathbb{I}[\mathcal{E} \cap \mathcal{U}] \leq \widehat{L}(\hat{h})\mathbb{I}[\mathcal{E}]$), we bound the overall expected population loss:

$$\begin{aligned}
\mathbb{E}\left[L_P(\hat{h})\right] &= \mathbb{E}\left[L_P(\hat{h}) \cdot \mathbb{I}[\mathcal{E} \cap \mathcal{U}]\right] + \mathbb{E}\left[L_P(\hat{h}) \cdot \mathbb{I}[\neg\mathcal{E} \cup \neg\mathcal{U}]\right] \\
&\leq \mathbb{E}\left[(\widehat{L}(\hat{h}) + \varepsilon/3) \cdot \mathbb{I}[\mathcal{E} \cap \mathcal{U}]\right] + 1 \cdot \mathbb{P}[\neg\mathcal{E} \cup \neg\mathcal{U}] \\
&\leq \mathbb{E}\left[\widehat{L}(\hat{h}) \cdot \mathbb{I}[\mathcal{E}]\right] + \frac{\varepsilon}{3} + \mathbb{P}[\neg\mathcal{E}] + \mathbb{P}[\neg\mathcal{U}] \\
&\leq \mathbb{E}\left[\widehat{L}(\hat{h}) \mid \mathcal{E}\right]\mathbb{P}[\mathcal{E}] + \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + \frac{\varepsilon}{3} \\
&\leq \gamma + \alpha + \varepsilon.
\end{aligned}$$

This establishes that $\mathcal{A}_{\text{PAC}}$ is an efficient $(\alpha - \varepsilon, \gamma + \alpha + \varepsilon)$-Gap PAC learner. □

# E Proofs from Section 6 – Fixed-Server Stateful Defense on the Uniform Distribution

**Proposition E.1** (Non-Triviality: Learnability without Noise – Restatement of Proposition 6.2)**.** *When $p = 0$, the concept class $\mathcal{H}$ is efficiently learnable: there exists a polynomial-time adversary using $d$ server queries in one round that recovers $c_a$ exactly (with overwhelming probability over the random oracle).*

*Proof.* Using its oracle access to $H$, the adversary locally samples random points and evaluates $H$ until it identifies $d$ points $x_1, \ldots, x_d \in \{0,1\}^d$ such that $H(x_1), \ldots, H(x_d)$ are linearly independent over $\mathrm{GF}(2)$. Since $H$ is a random oracle, each evaluation returns an independent uniform vector in $\{0,1\}^d$, and in expectation $d + O(1)$ random oracle queries suffice to find such a set (as each new random vector extends the current subspace with probability $\geq 1/2$).

The adversary then makes $d$ server queries on $x_1, \ldots, x_d$, receiving exact labels $y_i = c_a(x_i) = \langle H(x_i), a\rangle$ (since $p = 0$). Forming the invertible matrix $M \in \{0,1\}^{d\times d}$ with rows $H(x_1), \ldots, H(x_d)$, the adversary solves $Ma = y$ over $\mathrm{GF}(2)$ via Gaussian elimination in $O(d^3)$ time and outputs $\hat{h} = c_{M^{-1}y}$. □

**Theorem E.2** (Anti-Distillation Security via LPN – Restatement of Theorem 6.3)**.** *Under the $\mathrm{LPN}_{d,p}$ assumption (Definition 6.1), in the random oracle model, the stateful server $\mathcal{S}_p$ satisfies for $\mathcal{D} = \mathrm{Uniform}(\{0,1\}^d)$ and any query budget $n = \mathrm{poly}(d)$:*

1. ***Accuracy:*** *$\mathcal{S}_p \in \mathfrak{S}(p, \mathcal{D})$ is p-accurate with loss* TV *for $n$ queries across $R$ rounds with respect to every $c_a \in \mathcal{H}$ and $\mathcal{D}$.*

2. ***Fixed-Server Average-Case (and Worst-Case) Anti-Distillation Security:*** *$\mathcal{S}_p$ provides* ***Fixed-$(\mathcal{S}_p, \mathcal{D})$ Average-Case $\left(\frac{1}{2} - \mathrm{negl}(d)\right)$-Anti-Distillation Security*** *(Definition 3.3(2), and hence also Worst-Case Security under Definition 3.3(1)) over the uniform prior $\mu = \mathrm{Uniform}(\mathcal{H})$ with loss* TV *against any polynomial-time adversary $\mathcal{A} \in \mathcal{T}_{\mathit{eff}}$ making $n = \mathrm{poly}(d)$ server queries and $\mathrm{poly}(d)$ random oracle queries:*

$$\forall \mathcal{A} \in \mathcal{T}_{\mathit{eff}} \quad \mathop{\mathbb{E}}_{a \leftarrow \{0,1\}^d} \left[\mathrm{Risk}(\mathcal{A}, \mathcal{S}_p, c_a, \mathcal{D})\right] \geq \frac{1}{2} - \mathrm{negl}(d). \tag{E.1}$$

*Proof.* **Part 1: Accuracy.** Since $c_a$ is deterministic and $\mathrm{L} = \mathrm{TV}$, the expected total loss of $\mathcal{S}_p$ on an honest client's queries $X = (x_{r,i})$ simplifies to $\sum_{r=1}^{R} \sum_{i=1}^{n_r} \mathbb{P}\left[y_{r,i} \neq c_a(x_{r,i})\right]$ (Section 3). For every query $x_{r,i}$ (whether fresh or a cache hit), the server returns $y_{r,i} = c_a(x_{r,i}) \oplus b$, where $b \leftarrow \mathrm{Bernoulli}(p)$ is the noise bit sampled when $x_{r,i}$ was first queried. Because an honest client samples $X$ independently of the server's randomness, $\mathbb{P}\left[y_{r,i} \neq c_a(x_{r,i})\right] = \mathbb{P}\left[b = 1\right] = p$ for every $(r, i)$. Summing over all $n$ queries gives $np$, confirming that $\mathcal{S}_p \in \mathfrak{S}(p, \mathcal{D})$.

**Part 2: Fixed-Server Average-Case Anti-Distillation Security.** Suppose for contradiction that there exists a polynomial-time adversary $\mathcal{A} \in \mathcal{T}_{\text{eff}}$ making at most $n = \mathrm{poly}(d)$ server queries and $q = \mathrm{poly}(d)$ random oracle queries such that, when $a \leftarrow \mathrm{Uniform}(\{0,1\}^d)$,

$$\mathop{\mathbb{E}}_{a \leftarrow \{0,1\}^d} \left[\mathrm{Risk}(\mathcal{A}, \mathcal{S}_p, c_a, \mathcal{D})\right] \leq \frac{1}{2} - \varepsilon \tag{E.2}$$

for some non-negligible $\varepsilon = \varepsilon(d)$. We construct a polynomial-time distinguisher $\mathcal{B}$ that breaks the $\mathrm{LPN}_{d,p}$ assumption using at most $n + q + 1$ samples.

$\mathcal{B}$ receives an oracle $\mathcal{O}$ that provides samples from either the $\mathrm{LPN}_{d,p}$ distribution (for a single uniform secret $a \leftarrow \{0,1\}^d$) or the uniform distribution on $\{0,1\}^d \times \{0,1\}$. $\mathcal{B}$ simulates $\mathcal{A}$'s interaction by maintaining tables $T_H$ (random oracle) and $T$ (server cache). When $\mathcal{A}$ interacts with a point $x$ for the first time (via either a server query or a random oracle query), $\mathcal{B}$ draws a fresh sample $(z, w) \leftarrow \mathcal{O}$ and stores $T_H[x] \leftarrow z$ and $T[x] \leftarrow w$.

- *Server query on $x$:* Return $T[x]$.
- *Random oracle query $H(x)$:* Return $T_H[x]$.

If $x$ has been seen before, $\mathcal{B}$ returns the stored value from the appropriate table.

After $\mathcal{A}$ outputs $\hat{h}$, $\mathcal{B}$ draws one more sample $(z^*, w^*) \leftarrow \mathcal{O}$, picks a fresh test point $x^* \leftarrow \mathcal{D}$ (if $x^* \in T_H$, which occurs with probability at most $(n+q)/2^d = \mathrm{negl}(d)$, $\mathcal{B}$ outputs a random bit; otherwise it programs $T_H[x^*] \leftarrow z^*$), evaluates $\hat{h}(x^*)$ (simulating any additional random oracle queries of $\hat{h}$ using $T_H$), and outputs 1 if $\hat{h}(x^*) = w^*$ and 0 otherwise.

**Analysis.**

*Case 1: $\mathcal{O}$ provides $\mathrm{LPN}_{d,p}$ samples for $a \leftarrow \mathrm{Uniform}(\{0,1\}^d)$.* Each sample has $z \leftarrow \{0,1\}^d$ uniformly and $w = \langle z, a \rangle \oplus b$ with $b \leftarrow \mathrm{Bernoulli}(p)$ for the secret $a$. The programmed values $T_H[x] = z$ are i.i.d. uniform in $\{0,1\}^d$, matching a true random oracle, and the cached server responses $T[x] = w = \langle z, a \rangle \oplus b = c_a(x) \oplus b$ exactly match the stateful server $\mathcal{S}_p$ on $c_a$. Conditioned on $x^* \notin T_H$ (which holds with probability $\geq 1 - (n+q)/2^d$), $c_a(x^*) = \langle z^*, a \rangle$ and $w^* = c_a(x^*) \oplus b^*$ with an independent $b^* \leftarrow \mathrm{Bernoulli}(p)$, so averaging over $a \leftarrow \{0,1\}^d$ gives

$$\begin{aligned} \mathbb{P}\left[\hat{h}(x^*) = w^*\right] &\geq \left(1 - \mathop{\mathbb{E}}_{a} \left[\mathrm{Risk}(\mathcal{A}, \mathcal{S}_p, c_a, \mathcal{D})\right]\right)(1-p) + \mathop{\mathbb{E}}_{a} \left[\mathrm{Risk}(\mathcal{A}, \mathcal{S}_p, c_a, \mathcal{D})\right] \cdot p - \frac{n+q}{2^d} \\ &\geq \left(\tfrac{1}{2} + \varepsilon\right)(1-p) + \left(\tfrac{1}{2} - \varepsilon\right) p - \mathrm{negl}(d) = \tfrac{1}{2} + \varepsilon(1-2p) - \mathrm{negl}(d). \end{aligned}$$

*Case 2: $\mathcal{O}$ provides uniform samples.* Each $w$ (and $w^*$) is a uniform random bit independent of $z$ (and $z^*$), so $\mathbb{P}\left[\hat{h}(x^*) = w^*\right] = 1/2$.

Thus $\mathcal{B}$ achieves distinguishing advantage at least $\varepsilon(1-2p) - \mathrm{negl}(d)$, which is non-negligible since $\varepsilon$ is non-negligible and $p < 1/2$, contradicting $\mathrm{LPN}_{d,p}$. □

# F Proofs from Section 7 – Example of an Undetectable Attack

This section is devoted to the proof of Theorem 7.2. Recall the set up from Section 7. We consider a target parity function $h_a(x) = \langle x, h^* \rangle \bmod 2$ parameterized by its secret coefficient vector $h^* := a \in \{0,1\}^d$ (and similarly denote the recovered coefficient vectors by $\hat{h}^{(1)}, \hat{h} \in \{0,1\}^d$). The server provides responses that are $\alpha$-accurate on the uniform distribution $\mathcal{D}$. Our goal is to recover $h^*$ exactly using an undetectable polynomial-time attack.

## F.1 Parameters and Attack Queries

For an arbitrarily small target constant $\varepsilon \in (0, 1/2)$, we define our bounds and structural parameters as follows:

- Target server accuracy bound: $\alpha = d^{-1/2-\varepsilon}$.
- Sparsity parameter: $t = d^{\varepsilon/10}$.

The attack constructs queries in two independent blocks. For both blocks, the adversary constructs a query matrix $X = G+B$. Under the Planted XOR assumption, $G$ is computationally indistinguishable from a uniformly random matrix, as specified below. Note that the two blocks are constructed in exactly the same manner except for the probability that each entry of $B$ is one ($p_1 = \frac{1}{dt}$ versus $p_2 = d^{-1/2}$).

**Block 1 (Approximate Assigner):**

- $m_1 = d^{1+\varepsilon}$ rows, $n_1 = d^{1+1.7\varepsilon}$ queries.
- $P^1 \in \{0,1\}^{m_1 \times n_1}$ where each row is chosen uniformly at random subject to having exactly $t$ ones.
- $B^1 \in \{0,1\}^{n_1 \times d}$ with i.i.d. entries $\leftarrow \mathrm{Ber}(p_1)$ where $p_1 = \frac{1}{dt}$ (and $\mathrm{Ber}(p)$ denotes $\mathrm{Bernoulli}(p)$).
- $G^1 \in \{0,1\}^{n_1 \times d}$ drawn uniformly subject to $P^1 G^1 = 0$.
- Query matrix $X^1 = G^1 + B^1$.

**Block 2 (Exact Corrector):**

- $m_2 = d^{1+\varepsilon}$ rows, $n_2 = d^{1+1.7\varepsilon}$ queries.
- $P^2 \in \{0,1\}^{m_2 \times n_2}$ where each row is chosen uniformly at random subject to having exactly $t$ ones.
- $B^2 \in \{0,1\}^{n_2 \times d}$ with i.i.d. entries $\leftarrow \mathrm{Ber}(p_2)$ where $p_2 = d^{-1/2}$.
- $G^2 \in \{0,1\}^{n_2 \times d}$ drawn uniformly subject to $P^2 G^2 = 0$.
- Query matrix $X^2 = G^2 + B^2$.

**Queries:**

- Consider each row of $X^1, X^2$ as a query, and send these $n = n_1 + n_2$ queries to the server.
- Let $y^1 \in \{0,1\}^{n_1}$ and $y^2 \in \{0,1\}^{n_2}$ be the server's labels for $X^1$ and $X^2$, respectively.

## F.2 Undetectability

Note that $m_b = n_b^{1-\Theta(\varepsilon)}$ and $t = n_b^{\Theta(\varepsilon)}$ for each block $b \in \{1,2\}$. Thus, by Lemma 8 of (Christ & Gunn, 2024), under the $\mathrm{XOR}_{n^{1-\Theta(\varepsilon)}, n^{\Theta(\varepsilon)}}$ assumption, the matrices $G^1$ and $G^2$ are computationally indistinguishable from uniformly random matrices. Since $B^1, B^2$ are sampled independently of

$G^1, G^2$, we have that $X^1 = G^1 + B^1$ and $X^2 = G^2 + B^2$ are also computationally indistinguishable from uniformly random matrices. Hence, no polynomial-time server can distinguish the adversary's queries from $n$ i.i.d. draws from $\mathcal{D}$.

### F.3 Additional Preliminaries: The Dependent Chernoff Bound

Because the queries generated in our matrix products contain specific overlaps, the corresponding indicator events are not fully independent. To analyze these rigorously, we rely on the Dependent Chernoff Bound (also known as Chernoff bounds with dependency graphs).

**Definition F.1** (Dependency Graph)**.** *Let $Z_1, \dots, Z_m$ be a set of random variables. A graph $G = ([m], E)$ is a dependency graph for $\{Z_i\}$ if for any $i \in [m]$, the variable $Z_i$ is mutually independent of the set of all variables $Z_j$ such that $(i, j) \notin E$. The maximum degree of $G$ is denoted by $\Delta$.*

**Theorem F.2** (Dependent Chernoff Bound (e.g., (Janson, 2004)))**.** *Let $Z_1, \dots, Z_m$ be a set of Boolean random variables with a dependency graph $G$ having maximum degree $\Delta$. Let $X = \sum_{i=1}^{m} Z_i$ and $\mu = \mathbb{E}[X]$. Then for any $0 < \delta < 1$:*

$$\mathbb{P}(X \leq (1-\delta)\mu) \leq \exp\left(-\frac{\delta^2 \mu}{2(\Delta+1)}\right) \tag{F.1}$$

*and for any $\delta > 0$:*

$$\mathbb{P}(X \geq (1+\delta)\mu) \leq \exp\left(-\frac{\min(\delta^2, \delta)\mu}{3(\Delta+1)}\right). \tag{F.2}$$

**Remark F.3.** *When the variables are mutually independent, the dependency graph has no edges ($\Delta = 0$), and Theorem F.2 naturally recovers the standard i.i.d. Chernoff bounds. The term $(\Delta + 1)$ acts as an inflation factor to the variance due to local dependencies.*

### F.4 Probabilistic Foundations

Before we discuss how to decode $h^*$, we establish the core probabilistic events (over the randomness of our construction and the server's answers). We show below that each event holds with high probability.

**Lemma F.4** (Bounded Server Noise Volume)**.** *Let $e^1 \in \{0,1\}^{n_1}$ and $e^2 \in \{0,1\}^{n_2}$ be the server's noise vectors, defined as $e^b = y^b \oplus X^b h^*$. With probability $1 - o(1)$, both $\|e^1\|_0 \leq d^{\varepsilon/10}\alpha \cdot n_1$ and $\|e^2\|_0 \leq d^{\varepsilon/10}\alpha \cdot n_2$ hold simultaneously.*

*Proof.* Fix any target parity model $h^* \in \{0,1\}^d$ and any polynomial-time stateful server $\mathcal{S} \in \mathfrak{S}(\alpha, \mathcal{D}) \cap \mathcal{T}_{\text{eff}}$. Consider the polynomial-time distinguisher $\mathcal{B}_{\text{err}}$ (with $h^*$ hardcoded) that takes a candidate query matrix $Z \in \{0,1\}^{n_b \times d}$ for block $b \in \{1,2\}$, runs the server $\mathcal{S}$ on the two-block batch of total size $n = n_1 + n_2 = 2n_b$ (using $Z$ for block $b$ and an independent block for the other round) to obtain $y^b \leftarrow \mathcal{S}(Z)$, computes the normalized Hamming error $W(Z) := \frac{1}{n_b}\|y^b \oplus Zh^*\|_0 \in [0,1]$, and outputs 1 with probability $W(Z)$. When both blocks consist of honest i.i.d. uniform queries $(U^1, U^2) \leftarrow \text{Uniform}(\{0,1\}^{2n_b \times d})$, the $\alpha$-accuracy of $\mathcal{S}$ across the $n = 2n_b$ queries (Definition 3.2) guarantees that $\mathbb{E}[\|e^1(U^1)\|_0 + \|e^2(U^2)\|_0] \leq \alpha(n_1 + n_2) = 2\alpha n_b$, and in particular $\mathbb{P}[\mathcal{B}_{\text{err}}(U^b) = 1] = \mathbb{E}[W(U^b)] \leq 2\alpha$. By Lemma 8 of (Christ & Gunn, 2024) (as shown in the Undetectability analysis above), the planted XOR query matrix $X^b$ is computationally indistinguishable from $U^b$ to any polynomial-time distinguisher, so $|\mathbb{P}[\mathcal{B}_{\text{err}}(X^b) = 1] - \mathbb{P}[\mathcal{B}_{\text{err}}(U^b) = 1]| \leq \text{negl}(d)$. Consequently, $\mathbb{E}[\|e^b\|_0] = n_b \mathbb{E}[W(X^b)] \leq 2\alpha n_b + \text{negl}(d)$. Applying Markov's inequality, the probability that the actual noise volume exceeds $\alpha n_b$ by a factor of $d^{\varepsilon/10}$ is bounded by:

$$\mathbb{P}\left(\|e^b\|_0 \geq d^{\varepsilon/10}\alpha n_b\right) \leq \frac{\mathbb{E}[\|e^b\|_0]}{d^{\varepsilon/10}\alpha n_b} \leq \frac{2\alpha n_b + \text{negl}(d)}{d^{\varepsilon/10}\alpha n_b} = (2 + o(1))d^{-\varepsilon/10} = o(1). \tag{F.3}$$

A union bound over both blocks $b \in \{1,2\}$ yields the result. □

**Lemma F.5** (Column Weights of Projection Matrices)**.** *With probability $1 - o(1)$, the maximum column weight of both $P^1$ and $P^2$ is bounded by $\log d$.*

*Proof.* By construction, each row of $P^b$ is chosen uniformly from all weight-$t$ vectors. The expected weight of any column in $P^b$ is $m_b t/n_b = d^{1+\varepsilon}d^{0.1\varepsilon}/d^{1+1.7\varepsilon} = d^{-0.6\varepsilon} \ll 1$. Because the rows of $P^b$ are sampled independently, the entries in any specific column are independent Bernoulli trials. By standard Chernoff bounds, the probability a specific column exceeds $\log d$ is bounded by $d^{-\Omega(\log d)}$. Taking a union bound over all $n_b$ columns gives the $1 - o(1)$ probability bound. □

Before we state the remaining events, let us note that the bounded column weights above allow us to model the rows of $P^bB^b$ (conditioned on any fixed realization of $P^b$ satisfying Lemma F.5, over the independent randomness of $B^b$) as a dependency graph with bounded maximum degree.

**Lemma F.6** (Dependency Graph of Query Blocks)**.** *Conditioned on any fixed matrix $P^b$ satisfying the event in Lemma F.5, over the randomness of $B^b$, the dependencies between the rows of the product matrix $P^bB^b$ can be modeled by a dependency graph with maximum degree $\Delta \le d^{0.1\varepsilon}\log d$.*

*Proof.* For a fixed matrix $P^b$, the matrix product $P^bB^b$ introduces dependencies between its rows because different rows of $P^b$ may have 1s in the same columns. Specifically, the $i$-th row of $P^bB^b$ is given by $\sum_{k=1}^{n_b} P^b_{i,k}B^b_{k,\cdot}$. Thus, row $i$ of the product relies strictly on the rows $k$ of $B^b$ where $P^b_{i,k} = 1$.

Since the rows of $B^b$ are mutually independent, for fixed $P^b$, any event concerning row $i$ of $P^bB^b$ is mutually independent of any set of rows $i'$ whose column supports in $P^b$ are disjoint from that of row $i$. We define an edge in our dependency graph between $i$ and $i'$ if they share at least one active column $k$ in $P^b$. Since each row $i$ has exactly $t$ active columns, and by Lemma F.5 each column contains at most $\log d$ ones, row $i$ can intersect with at most $t \cdot \log d = d^{0.1\varepsilon}\log d$ other rows. Therefore, $\Delta \le d^{0.1\varepsilon}\log d$. □

We now proceed to state and bound the probability of the remaining three events (conditioning on any fixed $P^b$ satisfying Lemma F.5 and taking probabilities over the independent randomness of $B^b$).

**Lemma F.7** (Density of Pure Indicators in Block 1)**.** *For each variable $j \in [d]$, let $S_j \subseteq [m_1]$ be the set of "pure indicator" equations in Block 1, defined as the set of all $i \in [m_1]$ such that $(P^1B^1)_{i,j} = 1$ and $(P^1B^1)_{i,l} = 0$ for all $l \neq j$. With probability $1 - o(1)$, $|S_j| \ge \frac{d^\varepsilon}{4e}$ for all $j \in [d]$ simultaneously.*

*Proof.* For a fixed row $i$, the entry $(P^1B^1)_{i,j}$ is the XOR sum of $t$ independent $\mathrm{Ber}(p_1)$ variables. Let $p_1'$ be the probability that this entry is 1. We have $p_1' = \frac{1-(1-2p_1)^t}{2}$. Since $p_1 = \frac{1}{dt}$, standard approximations yield $\frac{1}{2d} \le p_1' \le \frac{1}{d}$ for any sufficiently large $d$.

Because the columns of $B^1$ are generated independently, the entries strictly across a fixed row $i$ of $P^1B^1$ are independent. The probability that row $i$ is a pure indicator for variable $j$ is exactly $p_1'(1-p_1')^{d-1}$, which we bound as:

$$\mathbb{P}(i \in S_j) \ge \frac{1}{2d}\left(1 - \frac{1}{d}\right)^{d-1} \ge \frac{1}{2ed} \tag{F.4}$$

Thus, the expected size is $\mu = \mathbb{E}[|S_j|] \ge m_1 \frac{1}{2ed} = \frac{d^{1+\varepsilon}}{2ed} = \frac{d^\varepsilon}{2e}$.

To bound the lower tail of $|S_j|$, we invoke the Dependent Chernoff Bound (Theorem F.2) using $\Delta \le d^{0.1\varepsilon}\log d$ (Lemma F.6) and $\delta = 1/2$:

$$\mathbb{P}\left(|S_j| \le \frac{d^\varepsilon}{4e}\right) \le \exp\left(-\frac{(1/2)^2(d^\varepsilon/2e)}{2(d^{0.1\varepsilon}\log d + 1)}\right) = \exp\left(-\Omega\left(\frac{d^{0.9\varepsilon}}{\log d}\right)\right) \tag{F.5}$$

A union bound over all $d$ variables $j \in [d]$ concludes the proof. □

**Lemma F.8** (Size of Active Equation Sets in Block 2)**.** *For any variable $j \in [d]$, let $I_j = \{i \in [m_2] \mid (P^2B^2)_{i,j} = 1\}$ be the set of "active" equations in Block 2. With probability $1 - o(1)$, $|I_j| \ge \frac{1}{4}d^{0.5+1.1\varepsilon}$ for all $j \in [d]$ simultaneously.*

*Proof.* The probability that $(P^2B^2)_{i,j} = 1$ is $p_2' = \frac{1-(1-2p_2)^t}{2}$. Using the inequality $(1-x)^t \leq 1 - tx + \frac{t^2}{2}x^2$, we can bound $p_2' \geq \frac{1}{2}(2tp_2 - 2t^2p_2^2) = tp_2(1-tp_2)$. Since $p_2 = d^{-1/2}$ and $t = d^{\varepsilon/10}$, $tp_2 \ll 1/2$ for large $d$, giving $p_2' \geq \frac{1}{2}tp_2 = \frac{1}{2}d^{-0.5+0.1\varepsilon}$.

The expected size is $\mu = \mathbb{E}[|I_j|] \geq m_2 \frac{1}{2} tp_2 = \frac{1}{2}d^{1+\varepsilon}d^{-0.5+0.1\varepsilon} = \frac{1}{2}d^{0.5+1.1\varepsilon}$. Applying Theorem F.2 with $\Delta \leq d^{0.1\varepsilon}\log d$:

$$\mathbb{P}\left(|I_j| \leq \frac{1}{4}d^{0.5+1.1\varepsilon}\right) \leq \exp\left(-\frac{(1/4)(0.5d^{0.5+1.1\varepsilon})}{2(d^{0.1\varepsilon}\log d + 1)}\right) = \exp\left(-\Omega\left(\frac{d^{0.5+\varepsilon}}{\log d}\right)\right) \tag{F.6}$$

A union bound over all $d$ variables $j \in [d]$ concludes the proof. □

**Lemma F.9** (Co-occurrence Bound in Block 2)**.** *For any pair of distinct variables $j, l \in [d]$, let $C_{j,l} = |I_j \cap I_l|$ be the number of equations in Block 2 containing both variables. With probability $1 - o(1)$, for all pairs $j \neq l$, $C_{j,l} \leq 2d^{1.2\varepsilon}$.*

*Proof.* For a fixed row $i \in [m_2]$, the elements in column $j$ and column $l$ of $B^2$ are completely independent. The probability that an equation contains both variables is exactly $(p_2')^2$. Since $(1-2p_2)^t \geq 1 - 2tp_2$, we have $p_2' \leq tp_2$. Thus, $(p_2')^2 \leq (tp_2)^2 = t^2d^{-1} = d^{-1+0.2\varepsilon}$.

The expected co-occurrence is $\mu \leq m_2 d^{-1+0.2\varepsilon} = d^{1+\varepsilon}d^{-1+0.2\varepsilon} = d^{1.2\varepsilon}$. Applying the upper tail of Theorem F.2 with $\delta = 1$ and $\Delta \leq d^{0.1\varepsilon}\log d$:

$$\mathbb{P}\left(C_{j,l} \geq 2d^{1.2\varepsilon}\right) \leq \exp\left(-\frac{1 \cdot d^{1.2\varepsilon}}{3(d^{0.1\varepsilon}\log d + 1)}\right) = \exp\left(-\Omega\left(\frac{d^{1.1\varepsilon}}{\log d}\right)\right) \tag{F.7}$$

A union bound over all $\binom{d}{2}$ pairs $j, l \in [d]$ yields the result. □

Let $\mathcal{E}$ be the intersection of the events in Lemmas F.4, F.5, F.7, F.8 and F.9. Since each holds with probability $1 - o(1)$, we have $\mathbb{P}(\mathcal{E}) = 1 - o(1)$. We condition on $\mathcal{E}$ for the remainder of the analysis.

### F.5 Block 1: Baseline Reconstruction

The adversary uses Block 1 to extract an approximate assignment $\hat{h}^{(1)}$. For each variable $j \in [d]$, the adversary identifies the pure indicator set $S_j$. It then assigns $\hat{h}_j^{(1)}$ as the majority vote of the server's labeled responses $(P^1y^1)_i$ for $i \in S_j$.

**Theorem F.10** (Approximate Assigner Guarantees)**.** *Conditioned on $\mathcal{E}$, the set of erroneously assigned variables $E = \{l \mid \hat{h}_l^{(1)} \neq h_l^*\}$ is bounded by $|E| \leq 8e \cdot d^{0.5-0.2\varepsilon}\log d$.*

*Proof.* In an ideal, noiseless scenario, for any $i \in S_j$, the equation expands to $(P^1y^1)_i = (P^1X^1h^*)_i = (P^1B^1h^*)_i = h_j^*$. Thus, a pure indicator equation $i \in S_j$ yields an incorrect vote for $h_j^*$ *if and only if* the server noise corrupted the equation. Specifically, an error occurs only if $(P^1e^1)_i = 1$.

By Lemmas F.4 and F.5, we deterministically bound the *total* number of corrupted equations $N_{\text{server}}^{(1)}$ across the entire first block. Since every error in $e^1$ propagates to at most the maximum column weight of $P^1$ equations, we have:

$$\begin{aligned} N_{\text{server}}^{(1)} &\leq \|e^1\|_0 \cdot (\text{max column weight of } P^1) \\ &\leq (d^{\varepsilon/10}\alpha n_1) \cdot (\log d) \\ &= (d^{0.1\varepsilon}d^{-0.5-\varepsilon}d^{1+1.7\varepsilon}) \cdot (\log d) \\ &= d^{0.5+0.8\varepsilon}\log d \end{aligned}$$

Let $W_j$ be the number of corrupted equations strictly inside the set $S_j$. Since each pure indicator row $i \in S_j$ has unit Hamming weight in $P^1B^1$ supported uniquely at column $j$, the sets $S_1, \ldots, S_d \subseteq [m_1]$ are pairwise disjoint, and thus $\sum_{j=1}^d W_j \leq N_{\text{server}}^{(1)}$. For a variable $j$ to be decoded incorrectly

(i.e., $j \in E$), the majority vote must fail, requiring $W_j \geq |S_j|/2$. By Lemma F.7, $|S_j| \geq \frac{d^\varepsilon}{4e}$, so failure requires $W_j \geq \frac{d^\varepsilon}{8e}$.

Therefore, the maximum number of failed variables $|E|$ is bounded by:

$$|E| \cdot \frac{d^\varepsilon}{8e} \leq \sum_{j \in E} W_j \leq d^{0.5+0.8\varepsilon} \log d, \tag{F.8}$$

which gives $|E| \leq 8e \cdot d^{0.5-0.2\varepsilon} \log d$ as desired. □

### F.6 Block 2: Exact Recovery

We execute the decoding algorithm on Block 2 to correct the residual errors in $\hat{h}^{(1)}$. For each active equation $i \in I_j$, the adversary isolates the vote for variable $j$ by substituting the remaining variables with estimates from Block 1:

$$V_{i,j} = (P^2 y^2)_i - \sum_{l \neq j} (P^2 B^2)_{i,l} \hat{h}_l^{(1)} \pmod 2 \tag{F.9}$$

The final assignment $\hat{h}_j$ is the majority vote among $V_{i,j}$ for $i \in I_j$.

**Theorem F.11** (Exact Recovery Guarantee)**.** *Conditioned on $\mathcal{E}$, the majority vote $\hat{h}_j$ is correct for all $j \in [d]$ simultaneously for any sufficiently large $d$. That is, $\hat{h} = h^*$.*

*Proof.* Fix an arbitrary variable $j \in [d]$. By Lemma F.8, the total number of votes available is $|I_j| \geq \frac{1}{4} d^{0.5+1.1\varepsilon}$. A vote $V_{i,j}$ evaluates incorrectly if and only if either:

1. **Server Noise:** The equation was corrupted by the server, i.e. $(P^2 e^2)_i = 1$.
2. **Substitution Noise:** A substitution error occurred, i.e. $(P^2 B^2)_{i,l} = 1$ for some variable $l \in E \setminus \{j\}$.

We bound both types of corruptions as follows.

**Server Noise Bound:** Just as in Block 1, the total number of equations in all of Block 2 corrupted by server noise is bounded by the noise weight and maximum column weight:

$$\begin{aligned} N_{\text{server_err}} &\leq \|e^2\|_0 \cdot (\text{max column weight of } P^2) \\ &\leq (d^{0.1\varepsilon} d^{-0.5-\varepsilon} d^{1+1.7\varepsilon}) \cdot (\log d) \\ &= d^{0.5+0.8\varepsilon} \log d, \end{aligned}$$

where we again use Lemmas F.4 and F.5 in the second inequality.

**Substitution Noise Bound:** An equation is corrupted by substitution only if it co-occurs with a variable $l \neq j$ that was assigned incorrectly in Block 1 ($l \in E \setminus \{j\}$). By Lemma F.9, any pair of distinct variables co-occurs in at most $2d^{1.2\varepsilon}$ equations. Consequently, the total number of substitution errors affecting variable $j$ is bounded by:

$$\begin{aligned} N_{\text{sub_err}} &\leq \sum_{l \in E \setminus \{j\}} C_{j,l} \leq |E| \cdot \max_{l \neq j} C_{j,l} \\ &\leq \left(8e \cdot d^{0.5-0.2\varepsilon} \log d\right) \cdot \left(2d^{1.2\varepsilon}\right) \\ &= 16e \cdot d^{0.5+1.0\varepsilon} \log d \end{aligned}$$

**Successful Decoding:** For the majority vote to correctly decode $h_j^*$, the sum of all corrupted votes must be strictly less than half of the total available equations in $I_j$:

$$N_{\text{server_err}} + N_{\text{sub_err}} < \frac{1}{2} |I_j| \tag{F.10}$$

However, from the bounds above, we have $N_{\text{server_err}} = o(|I_j|)$, and $N_{\text{sub_err}} = o(|I_j|)$. Therefore, for all sufficiently large $d$, this condition holds true. In other words, $\hat{h}_j = h_j^*$ as desired. □

### F.7 On Improving the Parameters in Our Construction

As proved above, our attack only works for accuracy parameter $\alpha = 1/d^{1/2+\varepsilon}$. Nevertheless, it seems plausible that this attack can be extended to handle a much larger $\alpha$ such as $\alpha = d^{-\varepsilon}$ or even $1/(\log d)^{O(1)}$ by using $\Theta(1/\varepsilon)$ blocks and $\Theta(\log d)$ blocks, respectively. The construction would geometrically increase the density of $B^b$ in each block $b$, and the decoding goal would then be to geometrically decrease the incorrect coordinates across each block (by a factor of $d^{\Theta(\varepsilon)}$ and a constant factor, respectively). Despite this, we unfortunately do not see any way to handle $\alpha$ all the way to some absolute (positive) constant, and we leave this as an open question.

Regarding the parameter in the XOR assumption (Definition 7.1), currently we require $m = n^{1-\Theta(\varepsilon)}$. Obviously, this allows us to tradeoff $m$ with $\alpha$. We note, however, that it would be even better to push $m = n^{\kappa}$ for any constant $\kappa > 0$. Again, this is another interesting open question.