

Manipulation-Proof Oblivious Audits against Deceptive Model Providers

Augustin Godinot¹, Sofiane Azogagh², Julien Ferry³, Sébastien Gambs⁴

¹INRIA Centre de l'Université de Rennes, France

²EURECOM, Sophia Antipolis, France

³CIRRELT & SCALE-AI Chair in Data-Driven Supply Chains, Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Canada

⁴Université du Québec à Montréal, Canada

augustin.godinot@inria.fr, sofiane.azogagh@eurecom.fr, julien.ferry@polymtl.ca, gambs.sebastien@uqam.ca

Abstract

Audits have emerged as a critical instrument for algorithmic governance, providing a mechanism for external scrutiny and governance of machine learning models. However, ensuring the integrity of such assessments remains a challenging issue. For instance in regulatory contexts, audits are typically declared or easily detected, thus enabling model providers to manipulate the process, whether intentionally or inadvertently. This vulnerability is particularly acute in the context of fairness evaluations, in which providers can often infer sensitive attributes and strategically equalize allocation rates between groups to satisfy fairness metrics. In this paper, we introduce a novel audit protocol designed to significantly increase the post-audit detectability of such manipulations by enabling the auditor to query the model in an oblivious manner. Our approach leverages a Private Information Retrieval mechanism to require the provider to label a large set of instances, while preventing it from knowing which subset will ultimately be used for the audit. The protocol is efficient, imposes minimal overhead on the auditor, and requires no modification to the audited model, its training procedure, or its inference pipeline. We provide theoretical guarantees showing that, under this protocol, a provider attempting to hide unfairness must falsify a significantly larger number of responses, thereby increasing both the difficulty and the likelihood of detection of manipulation. Experimental results across representative audit scenarios confirm the effectiveness and practicality of our approach.

Code — github.com/sofianeazogagh/oblivious_audit

Paper with appendices — arxiv.org/abs/2608.04365

1 Introduction

As machine learning (ML) systems increasingly mediate access to essential resources and public discourse, robust ML governance has become necessary to ensure these systems operate fairly and comply with legal and ethical standards (Chandrasekaran et al. 2021). A central pillar of this governance is algorithmic auditing, which allows independent evaluators to scrutinize model behavior (Metaxa et al. 2021; Costanza-Chock, Raji, and Buolamwini 2022). However, effective external auditing is currently hindered by a severe power asymmetry: model providers maintain strict

control over system access, training data and usage statistics (Birhane et al. 2024).

While several mechanisms attempt to bridge this transparency gap—such as dedicated research APIs (Entrena-Serrano et al. 2025), academic partnerships (Social Science One 2019) and legally mandated data access—they often face practical limitations and remain vulnerable to provider interference (Bourrée et al. 2023). A recent example highlights this fragility: the Meta Ad Library, a tool intended to help regulators and researchers track fraudulent advertisements via keyword searches. A 2025 Reuters’ report revealed that Meta actively monitored the specific keywords and celebrity names used by Japanese regulators investigating scam ads (Horwitz 2025). The company then repeatedly ran those exact searches to preemptively delete the targeted ads in Japan. By selectively scrubbing the specific data surfaced by the regulators’ queries rather than deploying systematic scam filtering, the provider was able to avoid a regulatory crackdown while avoiding the cost of systematic ad verification. This reactive curation creates a false impression of compliance, demonstrating how easily current transparency mechanisms can be manipulated by providers when audits queries are observable by the auditee (Fukuchi, Hara, and Maehara 2020; Garcia Bourrée et al. 2025).

In this work, we consider a generic regulation setting that captures the dynamics illustrated by the Ad Library example. A regulator wishes to audit a company providing a prediction service, whether a prediction API or a model used as part of a larger system. We assume that the regulator can impose an audit protocol on the provider, provided a reasonable computational cost. The provider is modeled as a *malicious* adversary: rather than incurring the cost of genuinely improving its system, it seeks to pass the audit by adversarially flipping a number of predictions, without modifying its deployed model. Indeed, the central challenge in this setup (regulator-led audits, requiring provider cooperation) is that the audited provider may strategically modify its outputs when answering audit queries, deviating from the outputs it would provide to genuine users. Not only is this type of manipulation difficult to detect (Garcia Bourrée et al. 2025), but a few modifications of the outputs on the audit set can have a dramatic impact on the audit conclusions (see Proposition 1). To address this, we propose a rigorous auditing framework that offers formal theoretical guarantees on the

audit’s conclusions and limits the extent of strategic manipulation the provider can perform. While our proposed framework is model- and metric-agnostic, we restrict our presentation to auditing demographic parity in binary classifiers for simplicity, and discuss in Section 6.1 how it extends to other settings. Our main contributions are as follows:

- We introduce $R^2\text{esPIR}$, an audit protocol designed to mitigate audit manipulations. The core idea is to hide the actual audit set S within a larger candidate set C , provably making manipulation harder for a deceptive provider. The only prerequisite for this protocol is that the audited provider must label a candidate set C larger than the actual audit set S .
- We prove that $R^2\text{esPIR}$ forces a deceptive provider to modify its answers on a substantially larger number of points in the candidate set C in order to fake the same level of demographic parity (Theorem 1). This results in an increased probability of manipulation detection by the auditor.
- We characterize how large the candidate set C must be, relative to the audit set S , to achieve a desired level of manipulation detection while demonstrating that the protocol does not incur prohibitive computational overhead, making it practical for real-world deployment.

Outline. First, in Section 2 we review the background on audits, attempts at making them robust, and Private Information Retrieval, which serves as a core building block of our approach. Next, in Section 3 we describe the auditing framework on which our work relies, and we explain why standard black-box fairness audits can be vulnerable to manipulation. To address this issue, we introduce in Section 4 an oblivious auditing protocol based on PIR, which we call $R^2\text{esPIR}$. We then establish the protocol’s theoretical guarantees in terms of manipulation effort and detectability, and evaluate it experimentally in Section 5. Finally, we conclude in Section 6 with a discussion regarding our framework’s applicability, the main design principles for hidden-audit-set protocols, and directions for future work.

2 Background & Related Work

2.1 Audits and Manipulations

Audits and regulation. Independent audits are a key mechanism for holding model providers accountable and fostering healthier algorithmic ecosystems (Metaxa et al. 2021; Vecchione, Levy, and Barocas 2021; Costanza-Chock, Raji, and Buolamwini 2022). This view has been adopted by recent regulations: the EU AI Act (Reg 2024), DSA (Reg 2022b), DMA (Reg 2022a) and related frameworks increasingly mandate external assessments of high-risk AI systems, making algorithmic auditing not only a research concern but a legal requirement.

Practical challenges to auditing. Several incidents highlight the practical difficulties of conducting trustworthy audits. In some cases, the studied algorithm changed without notice during the study, as in the 2020 Facebook polarization study (Ribeiro 2024). In others, promised (or required) data access was repeatedly delayed (Social Science

One 2019) or simply not provided (European Commission 2025). Finally, when data or models are eventually shared, they may contain errors that undermine published academic work (Timberg 2021) or fail to reflect the reality experienced by users (Entrena-Serrano et al. 2025; Pearson et al. 2024).

Manipulability of audit targets. Beyond practical obstruction, a growing body of work demonstrates that many components of ML systems can be strategically manipulated. Training data and sampling procedures can be biased to fake fairness (Fukuchi, Hara, and Maehara 2020). Model explanations can be fairwashed to conceal discriminatory behavior (Aïvodji et al. 2019; Anders et al. 2020; Shamsabadi et al. 2022; Fokkema, de Heide, and van Erven 2023). Model outputs themselves can be selectively altered to deceive black-box auditors (Bourrée et al. 2023; Garcia Bourrée et al. 2025). And more broadly, fairness metrics are susceptible to gaming when providers can anticipate the evaluation procedure (Hutchinson et al. 2022; Thomas and Uminsky 2022).

2.2 Related Work on Robust Auditing

Existing work on making audits more robust to manipulation can be organized into three broad categories.

(1) *Data-driven and prior-based approaches.* Several works propose auditing protocols that leverage prior knowledge or careful query design to limit manipulation. Yan and Zhang (2022) study active fairness auditing, where the auditor adaptively selects queries to efficiently estimate fairness violations. Yadav, Moshkovitz, and Chaudhuri (2024) provide a theoretical framework for auditing with explanations. Most closely related to our work, Garcia Bourrée et al. (2025) formally study manipulation-proof auditing and show that auditors with prior knowledge about the ground truth can detect manipulations, provided the provider does not know the exact prior they use. Complementary to these approaches, our method allows to relax the private prior assumption by cryptographically hiding the auditor’s queries.

(2) *Cryptographic approaches.* A second line of work uses cryptographic tools to provide verifiable guarantees. Shamsabadi et al. (2023) propose zero-knowledge proofs of fair training for decision trees. Waiwitlikhit et al. (2024) design a protocol based on zero-knowledge proofs that enables trustless audits without revealing models or data. Franzese et al. (2025) introduce online fairness certificates via a scalable zero-knowledge proof protocol that verifies fairness with respect to data received during deployment. Pentyala et al. (2022) develop privacy-preserving fairness auditing using secure computation. These approaches offer strong guarantees but typically require the provider’s active cooperation in running the cryptographic protocol and may impose significant computational overhead. In contrast, our proposed framework relies on lightweight PIR primitives and incurs only negligible overhead.

(3) *Secure hardware.* A third category relies on trusted execution environments or secure hardware to ensure audit integrity (Park, Kim, and Lim 2022). While promising, these approaches require trust in the hardware manufacturer and physical deployment constraints, which limit their applicability in many regulatory contexts.

2.3 Private Information Retrieval

Private Information Retrieval (PIR) is a cryptographic primitive that enables a client to retrieve an item from a database without revealing to the server which item is being accessed (Chor et al. 1998). A large body of work has proposed different PIR constructions depending on the trust and deployment assumptions. One approach distributes the database across multiple non-colluding servers to obtain information-theoretic privacy guarantees (Olumofin and Goldberg 2011). Other approaches rely on trusted hardware (Smith and Safford 2001) but the most promising ones are homomorphic encryption schemes based on lattice problems such as Learning With Errors (LWE) (Melchor et al. 2016; Angel et al. 2018; Ali et al. 2021; Ahmad et al. 2021; Mughees, Chen, and Ren 2021; Menon and Wu 2022; Davidson, Pestana, and Celi 2023; Henzinger et al. 2023). In these computational PIR based on homomorphic encryption, the client typically encodes its query as an encrypted selection vector, conceptually corresponding to a one-hot vector indicating the desired record index. Afterwards, the server homomorphically combines this encrypted query with the database contents, which are generally represented as a matrix, which means that answering a query essentially reduces to an encrypted matrix-vector multiplication. The resulting ciphertext decrypts to the requested record while the server learns nothing about the queried index. However, this approach generally incurs a computational cost linear in the database size for each query, which severely limits its practicality for large-scale databases. To overcome this limitation, a recent line of work has introduced so-called *stateful PIR* schemes, in which a significant portion of the computation and communication is shifted to an offline preprocessing phase (Corrigan-Gibbs and Kogan 2020; Kogan and Corrigan-Gibbs 2021; Shi et al. 2021; Mughees, Chen, and Ren 2021; Corrigan-Gibbs, Henzinger, and Kogan 2022). In this stateful PIR, the server first computes and sends to the client a public or semi-public hint (or *digest*) that depends solely on the database and can be reused across multiple queries. Once this preprocessing step is completed, each individual query can be answered with substantially reduced communication complexity and the server becomes sublinear in the database size.

Beyond achieving sublinear asymptotic costs, stateful PIR techniques have also been leveraged, not necessarily to achieve sublinear costs, but to make the linear cost per query significantly more efficient (Henzinger et al. 2023; Davidson, Pestana, and Celi 2023). A prominent example is SimplePIR (Henzinger et al. 2023), which relies on Regev’s LWE-based encryption scheme (Regev 2009). The key insight of SimplePIR is to exploit the linear structure of Regev encryption to precompute the vast majority of the expensive matrix operations associated with the database during the offline phase. As a result, the online query phase only requires lightweight computations that depend on the encrypted query, while the bulk of the matrix-vector multiplication is carried out in advance and depends exclusively on the database. This property makes SimplePIR particularly well suited to our setting, in which the provider is informed in advance of an upcoming audit and jointly agrees with

the auditor on the audit set. Once this agreement is established, the provider can label the audit set accordingly and transmit a corresponding digest to the auditor (see details in Section 4). Furthermore, De Castro and Lee (de Castro and Lee 2024) introduced VeriSimplePIR, an extension of SimplePIR that ensures verifiability. More precisely, it allows the client (*i.e.*, the auditor in our case) to verify that the database queried is exactly the one used to generate the digest sent by the server (*i.e.*, the provider in our setting) without disclosing the content of the database. This verifiability is achieved at no additional online cost and can be used to prevent the provider from changing the labels during the audit.

3 The Black-Box Auditing Game

In this section, we first describe the typical black-box fairness auditing setup considered in this paper. Then, we theoretically quantify its vulnerability to manipulation.

3.1 Black-Box Fairness Auditing

Let $\mathcal{X}$ be the input space, and let $h : \mathcal{X} \rightarrow \{0, 1\}$ denote the provider’s (binary) prediction function. Each query $x \in \mathcal{X}$ is associated with a binary *protected* attribute $A(x) \in \{0, 1\}$, which partitions the population into two *protected groups* (those with $A = 1$ and those with $A = 0$).

The *demographic parity* gap of h quantifies how differently h assigns positive outcomes across these two groups. More precisely, it is defined as the difference between the probability of predicting the positive class for individuals in group $A = 1$ and the corresponding probability for individuals in group $A = 0$:

$$d_{\mathcal{D}}(h) := \mathbb{P}_{X \sim \mathcal{D}}(h(X) = 1 | A(X) = 1) - \mathbb{P}_{X \sim \mathcal{D}}(h(X) = 1 | A(X) = 0),$$

in which $\mathcal{D}$ denotes the underlying data distribution. In practice, since $\mathcal{D}$ is typically unknown, this quantity is estimated on a finite sample set $S \subset \mathcal{X}$. For $a \in \{0, 1\}$, let $n_a = |\{x \in S : A(x) = a\}|$ and define the empirical positive prediction rate over protected group a as

$$p_{a,S}(h) := \frac{1}{n_a} \sum_{x \in S_a} h(x).$$

The demographic parity gap of h on S is then

$$d_S(h) := p_{1,S}(h) - p_{0,S}(h).$$

When the classifier h is clear from the context, we omit it from the notation and simply write $p_{a,S}$ and d_S . Given a tolerance $\varepsilon \geq 0$, we say that h satisfies the demographic parity constraint on S if

$$|d_S| \leq \varepsilon, \quad (1)$$

in which ε is a chosen tolerance threshold. The objective of the auditor is to verify whether Constraint (1) holds for a given *audit set* S . More precisely, if $|d_S| \leq \varepsilon$, the ML provider passes the test while otherwise, if $|d_S| > \varepsilon$, the provider is flagged as unfair by the auditor. However, a key challenge is that the auditor often does not have access to

the model h itself, nor to its training data, code or documentation. To estimate the demographic parity gap on an audit set S , the auditor must therefore *resort to black-box access* and query h on a set of inputs $S \subset \mathcal{X}$ that they collect. The resulting predictions are then used to compute d_S . This black-box audit procedure is described in Algorithm 1. In the next subsection, we show how this framework is vulnerable to provider manipulation.

3.2 Manipulating Black-box Audit

One important limitation of typical demographic parity audits conducted as described in Section 3.1 is that the auditor only has access to query answers (through the provider’s audit API) and no knowledge of the underlying model. It is therefore possible for a malicious provider to adversarially manipulate the audit by returning a label $h(x)$ that differs from the underlying model’s true output. To quantify how easy such manipulation is, Proposition 1 characterizes the minimum number of predictions that the model provider must flip in order to fall below the detection threshold.

Proposition 1 (Black-box audit manipulation). *Let $S \subset \mathcal{X}$ be the audit set (with $|S| = n$), let d_{true} denote the demographic parity gap of the provider’s model on S (in the absence of manipulation), let $n_{\min} = \min(n_0, n_1)$ be the size of the smallest protected group in S , and let ε be the audit threshold. Assuming the model would fail the audit, i.e., $|d_{\text{true}}| > \varepsilon$, the minimum number m_{vanilla} of predictions that the provider must flip in order to pass the audit satisfies*

$$m_{\text{vanilla}} \geq \lceil (|d_{\text{true}}| - \varepsilon) n_{\min} \rceil. \quad (2)$$

Proof. Assume without loss of generality that $d_{\text{true}} > 0$. Flipping a prediction $h(x)$ from 1 to 0 for an example with $A(x) = 1$ decreases $p_{1,S}$, and hence the gap on S , by at most $1/n_1$. Similarly, flipping a prediction from 0 to 1 for an example with $A(x) = 0$ increases $p_{0,S}$ and decreases the gap by at most $1/n_0$. Since a single flip affects only one group rate, the gap can decrease by at most

$$\max\left(\frac{1}{n_0}, \frac{1}{n_1}\right) = \frac{1}{n_{\min}}$$

per modification. Therefore, reducing the gap from d_{true} to at most ε requires at least $(d_{\text{true}} - \varepsilon) n_{\min}$ modifications, hence at least $\lceil (d_{\text{true}} - \varepsilon) n_{\min} \rceil$ since m_{vanilla} is an integer. $\square$

Proposition 1 provides a lower bound on the number of output modifications that a provider must perform in order to pass the audit. This bound is linear in the amount of unfairness to be hidden but also in the size of the smallest protected group in the audit set, which means that unbalanced audit sets therefore greatly facilitate manipulation. However, even for a balanced audit set, manipulating the audited fairness remains easy. For instance, to hide a disparity of 0.01 (i.e., $d_{\text{true}} - \varepsilon = 0.01$) with a balanced audit set of $n = 400$ queries, the provider only has to modify two predictions.

The manipulation model considered in this work therefore consists of arbitrary output flipping. By allowing any (worst-case) modification to the prediction vector returned by the platform, this model generalizes more constrained threat models and yields model-agnostic guarantees.

Algorithm 1: Black-Box audit procedure

Require: $\mathcal{X}, h$

- 1: Auditor gathers or generate an audit dataset $S \subset \mathcal{X}$
 - 2: For each example $x \in S$, the auditor queries the provider for $h(x)$
 - 3: The auditor computes the parity gap d_S
 - 4: **if** $d_S < \varepsilon$ **then**
 - 5: | **return** Pass
 - 6: **else**
 - 7: | **return** Fail
-

Assume that the auditor possesses a small *verification set* of k queries (also called *canaries*). These queries are externally labeled (e.g., via crowdsourcing or by directly querying the provider’s public API), and manipulation is detected as soon as at least one verification query is modified by the provider. Since the auditor controls the audit set S , we assume that the canaries are included in S by construction. Therefore, the canary effectiveness can be modeled through a single *catch-all* parameter $q \in [0, 1]$, interpreted as the probability that a canary included in the audit set remains indistinguishable to the provider (so that the provider cannot selectively avoid modifying it). Let $n := |S|$ denote the size of the audit set, and let m denote the number of audit set queries whose outputs are adversarially modified by the provider. Then, conditional on being effectively present, a canary is modified with probability m/n , and the probability that manipulation is detected satisfies

$$\mathbb{P}(\text{manipulation detected}) = 1 - (1 - qm/n)^k. \quad (3)$$

Overall, this highlights that typical black-box fairness audits can be highly brittle, as a small number of targeted output modifications may suffice to hide unfairness. Moreover, our detection analysis shows that the probability of catching such manipulation depends critically on both the number of adversarial manipulations m and the size k of the verification set: small k provides limited guarantees when only a few outputs are modified, whereas detection quickly becomes likely as either m or k increases.

While increasing the number of canaries k is a simple countermeasure, it is often impractical or too costly in practice. Instead, in the next section, we propose an oblivious auditing protocol that prevents the provider from tailoring its modifications to the specific audit set. As a result, to hide unfairness with high probability, the provider is forced to perform a larger number m of output modifications, which increases the cost of manipulation and its detectability.

4 Oblivious Auditing

4.1 R²esPIR Protocol

Based on the observation that hiding unfairness over a known finite set is easily achieved, we propose a new audit protocol coined Robust Research audit APIs enabled by Private Information Retrieval (R²esPIR). To mitigate audit manipulations, R²esPIR forces the provider to find a strategy to optimize its manipulations on a significantly larger *candidate set* C rather than on the particular audit set S

drawn by the auditor. The cryptographic primitive that prevents the provider from knowing which sample is used in the audit is the Private Information Retrieval (PIR) (cf. Section 2.3). The audit protocol consists of three steps, described in Algorithm 2:

- Step 1 (Algorithm 2, Line 1). The provider and the auditor agree on a (large) *candidate set* C . In practice the candidate set can be proposed by the auditor (e.g., derived from a large public dataset) or by the provider (e.g., the company is required to share internal information to prepare for the audit). The selection process for C can even be iterative: after one party proposes a first candidate set, the other can ask for additions or deletions to reach a more representative distribution.
- Step 2 (Algorithm 2, Lines 2 and 3) The auditor samples an *audit set* $S \subset C$ among the candidate set and runs it through the PIR mechanism. The PIR mechanism guarantees that 1) the provider cannot infer the exact audit queries used in S and 2) the auditor cannot use the interaction to gather more information about the output of h other than what is given by $h(S)$. Thus, the PIR properties effectively exclude any manipulation during the interaction between the auditor and the provider at step 2.
- Step 3 (Algorithm 2, Lines 4 to 7) Using the answers of the provider $h(S)$ gathered during the PIR phase, the auditor produces an estimate d_S of the audit metric and decides whether the provider passes the audit or not. The goal of a malicious provider is to make the auditor accept its model (i.e., $d_S \leq \varepsilon$) when it should not have passed (i.e., $d_{\text{true}} > \varepsilon$).

We further highlight the following design properties. First, the candidate set C is constructed to be representative of the (unknown) data distribution $\mathcal{D}$. Consequently, when the audit set $S \subseteq C$ is sampled uniformly at random and is sufficiently large, it is expected to closely reflect $\mathcal{D}$. While S could be sampled from C in other ways, uniform sampling is used because it (i) yields representativeness via standard concentration arguments and is easier for the provider to accept, whereas tailored sampling can appear arbitrary to target specific fairness values, (ii) requires no prediction-dependent design since the auditor sees features in C but not the model’s outputs, and (iii) enables a clean probabilistic derivation of manipulation lower bounds (Theorem 1).

Moreover, planting a small number k of verification queries (*canaries*) into S does not materially affect this representativeness when $k \ll |S|$. As a result, it is reasonable to require the provider to pass the fairness verification (i.e., to exhibit unfairness of at most ε) on S .

Second, although the provider is required to label a potentially large candidate set C , the amount of information revealed to the auditor is the same as in the traditional black-box auditing setup described in Section 3.1, since the sampled audit set S need not be larger. Because the provider’s underlying model may be proprietary or constitute a business secret, this property is important in practice.

Third, the use of a PIR scheme prevents the provider from tailoring its responses to a known audit set. Instead, to pass the audit with high probability, the provider must ensure that

Algorithm 2: The $R^2\text{esPIR}$ audit procedure

Require: $\mathcal{X}, h$

- 1: Provider and auditor agree on a large candidate set $C \subset \mathcal{X}$
 - 2: Provider privately labels all examples in C , creating $D = \{(x, h(x)) : x \in C\}$
 - 3: Auditor interacts with D to retrieve outputs on an audit set $S \subset D$ using the PIR primitive
 - 4: **if** $d_S < \varepsilon$ **then**
 - 5: **return** Pass
 - 6: **else**
 - 7: **return** Fail
-

the fairness constraint holds for a random audit set drawn from the candidate set, implying that fairness must generalize across the candidate set rather than being satisfied only on a specific subset. In Section 4.3, we provide a probabilistic analysis showing that this substantially increases both the cost and detectability of manipulation.

4.2 VeriSimplePIR Applied to an Audit Set

Let S denote the audit set and h the provider’s prediction function. The primary advantage of using a PIR protocol for auditing, as opposed to sending an encrypted version of the audit set S , lies in its efficiency as PIR is considerably faster than evaluating the model on S via homomorphic encryption. For example, if h is a neural network, running $x \in S$ through h with homomorphic encryption would require as many matrix-vector multiplications as there are network layers. In contrast, the PIR protocol reduces this to a single, lightweight matrix-vector multiplication per $x \in S$.

While a detailed description of the interaction during the PIR between the auditor and the provider is provided in Figure 3 in Appendix A, we summarize here the main steps of the protocol at a high level. VeriSimplePIR proceeds in two phases: an offline phase for setup and commitment and an online phase for querying. The process starts when the auditor requests an audit from the provider. After both parties agree on the audit set S , the provider applies its prediction function h to label the samples, resulting in a database $h(S)$. These labels are then packed into a small matrix $\mathbf{D} \in \mathbb{Z}_p^{\ell \times \mu}$, with the parameters ℓ , p , and μ specified by the auditor to satisfy security requirements. Additionally, the provider provides an indexing table to assist the auditor in mapping labels to the corresponding samples in $\mathbf{D}$.

The protocol then proceeds through four main functions: **Commitment**, **Query**, **Answer** and **Recovery**. During the offline phase, the provider performs the **Commitment** function by generating a cryptographic digest of the matrix $\mathbf{D}$ and a “hint” sent to the auditor to verify that $\mathbf{D}$ is well formed. The auditor cannot learn the content of the database from the digest or the hint. Optionally, the provider can also precompute a proof π that will later be used to verify that queries are performed on the committed database (see Construction 5.1 in (de Castro and Lee 2024) for more details about this proof).

In the online phase, the auditor executes the **Query** function by selecting an index $i = (i_r, i_c)$ and sending an

	Database size				
	128 KiB	128 MiB	1 GiB	4 GiB	8 GiB
Query size	0.375 KiB	13.16 KiB	37.76 KiB	77.85 KiB	110.1 KiB
Answer size	7.812 KiB	276.6 KiB	816.8 KiB	1.644 MiB	2.325 MiB
Bandwidth usage	8.187 KiB	289.8 KiB	854.6 KiB	1.722 MiB	2.435 MiB
Query time	0.5024 ms	1.9587 ms	5.3174 ms	10.5765 ms	15.3573 ms
Answer time	0.0221 ms	12.9588 ms	69.1425 ms	1447.86 ms	2816.19 ms
Recovery time	0.574 ms	13.247 ms	42.825 ms	169.721 ms	807.623 ms
Total online time	1.09 ms	28.16 ms	117.28 ms	1628.15 ms	3639.17 ms

Table 1: Performance of VeriSimplePIR (de Castro and Lee 2024) on the online phase for different database sizes.

encryption of the one-hot encoding of i_c to the provider using the Regev’s LWE encryption scheme (Regev 2009). The provider then performs the **Answer** function by computing a matrix-vector multiplication between the **D** and the encrypted query. Finally, the auditor executes the **Recovery** function to decrypt and extract the queried value from the response. The auditor can perform multiple queries and verify them all at once using the precomputed proof π (see (de Castro and Lee 2024) for more details about this proof). To demonstrate how the protocol performs in practice, Table 1 reports timing measurements for the online phase (query, answer and recovery steps) along the bandwidth usage, across various sizes of $h(S)$, in which h is a binary classifier. The given results are an average of 10 runs. When $\forall y \in h(S), \log_2(y) < \log_2(p)$, multiple labels y can be encoded within a single element in $\mathbb{Z}_p$, so each entry in **D** can store several labels. As a result, the matrix **D** can be considerably smaller than the raw set of labels $h(S)$. The results reported shows that the PIR protocol maintains practical efficiency even when handling large audit sets.

4.3 Manipulation Difficulty Analysis

We now establish probabilistic guarantees that hiding the audit set through $R^2\text{esPIR}$ increases the difficulty of targeted manipulations. The results formalize a core intuition: if the provider cannot identify the audited queries, then passing the audit with high probability requires improving behavior on a nontrivial fraction of the entire candidate set, rather than selectively patching a known evaluation subset.

Theorem 1 ($R^2\text{esPIR}$ audit manipulation). *Let $C \subset \mathcal{X}$ be the candidate set (with $|C| = N$) and let $S \subset C$ be the audit set (with $|S| = n$), sampled uniformly at random without replacement from C . Let $d_{C,\text{true}}$ denote the demographic parity gap of the provider’s model on C (in the absence of manipulation), let $n_{\min} = \min(n_0, n_1)$ be the size of the smallest protected group in S , let $N_{\min} = \min(N_0, N_1)$ be the size of the smallest protected group in C , and let ε be the audit threshold. For $\delta \in (0, \frac{1}{2}]$, assuming the model would fail the audit, i.e., $|d_{C,\text{true}}| > \varepsilon$, it suffices for the provider to flip at least*

$$m_{R^2\text{esPIR}} \geq \left\lceil \left(|d_{C,\text{true}}| - \varepsilon + \sqrt{\frac{2 \ln(4/\delta)}{n_{\min}}} \right) N_{\min} \right\rceil \quad (4)$$

predictions in C in order to pass the audit with probability at least $1 - \delta$.

Proof. Let $\gamma := \sqrt{\frac{2 \ln(4/\delta)}{n_{\min}}}$. By Theorem 2 (provided hereafter in Section 4.3), a sufficient condition for passing the audit with probability at least $1 - \delta$ (conditional on n_0, n_1) is that the (post-manipulation) candidate-set gap satisfies

$$|d_C| \leq \varepsilon - \gamma.$$

Thus, it suffices for the provider to modify predictions on C so as to reduce the initial gap $|d_{C,\text{true}}|$ down to at most $\varepsilon - \gamma$.

We now bound the number of prediction flips required to achieve such a reduction on C . Assume without loss of generality that $d_{C,\text{true}} > 0$ (otherwise apply the same argument to $-d_{C,\text{true}}$). Consider a single flip of one prediction on C . If the flipped example belongs to group $A = 1$, then $p_{1,C}$ changes by at most $1/N_1$, hence d_C changes by at most $1/N_1$. If it belongs to group $A = 0$, then $p_{0,C}$ changes by at most $1/N_0$, hence d_C changes by at most $1/N_0$. Therefore, a single flip can change d_C by at most

$$\max\left(\frac{1}{N_0}, \frac{1}{N_1}\right) = \frac{1}{N_{\min}}.$$

To reach $d_C \leq \varepsilon - \gamma$, the provider must decrease the gap by at least

$$d_{C,\text{true}} - (\varepsilon - \gamma) = |d_{C,\text{true}}| - \varepsilon + \gamma.$$

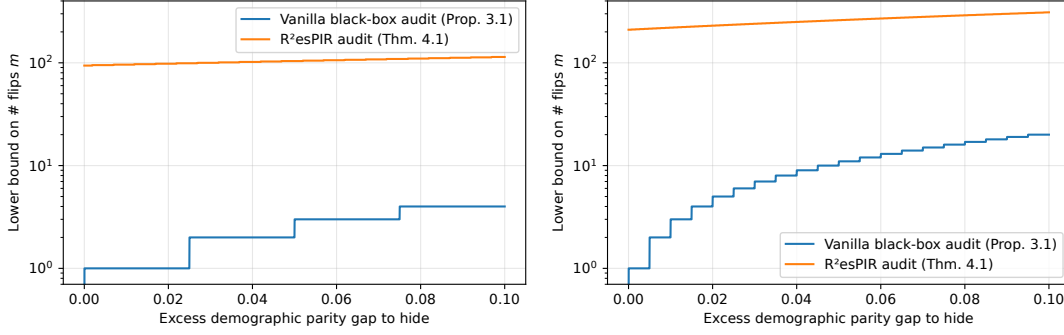
Since each flip can reduce d_C by at most $1/N_{\min}$, it suffices to flip at least

$$\lceil (|d_{C,\text{true}}| - \varepsilon + \gamma) N_{\min} \rceil$$

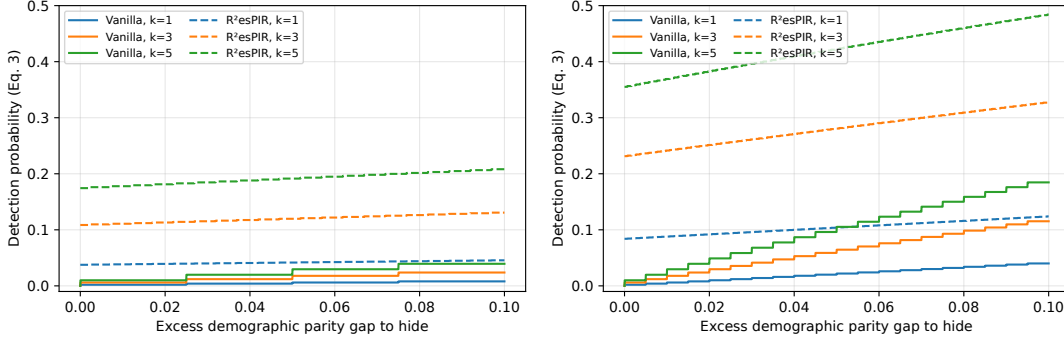
predictions in C , as claimed. $\square$

The manipulation resistance of $R^2\text{esPIR}$ (Theorem 1) improves on that of a “vanilla” black-box audit (Proposition 1) in two complementary ways. First, $R^2\text{esPIR}$ enforces a stricter effective target on the provider’s underlying disparity because of the sampling uncertainty term $\sqrt{2 \ln(4/\delta)/n_{\min}}$. This term materializes the fact that under uniform random sampling from C , audits sets S might exhibit unfairness d_S slightly lower or higher due to the sampling process. Nevertheless, as long as they are representative of the underlying distribution (which is the case unless $n = |S|$ is pathologically small), the provider’s model is expected to behave fairly on them as well. In other words, to ensure unfairness does not exceed ε on the sampled S with high probability, the provider must over-correct unfairness over the candidate set. Second, $R^2\text{esPIR}$ increases the scale at which modifications must be performed. In a vanilla black-box audit, the provider may concentrate changes on the audited set S , so the best-case manipulation complexity scales with $n_{\min}$. Under $R^2\text{esPIR}$, the provider must instead improve fairness across the entire candidate set C to pass for an unknown random S , implying a manipulation complexity that scales with $N_{\min}$, the size of the smaller protected group in C . Since typically $N_{\min} \gg n_{\min}$, this yields a substantially larger manipulation cost.

Taken together, these two effects force the provider to modify a substantially larger number of outputs to pass reliably, which in turn increases the probability that manipulation is detected. Indeed, given a verification set of k canaries



(a) Lower bounds on the number of output flips m required to hide unfairness, for the unbalanced setting (left) and the balanced setting (right).



(b) Probability that the auditor detects manipulation, for the unbalanced setting (left) and the balanced setting (right).

Figure 1: Illustration of our theoretical bounds on manipulation difficulty for a vanilla black-box audit and for $R^2\text{esPIR}$. We consider audit sets of size $|S| = 400$ and $k \in \{1, 3, 5\}$ canaries, each detected with effectiveness probability $q = 0.8$. For $R^2\text{esPIR}$, we use a candidate set of size $|C| = 2,000$ and target passing probability $1 - \delta = 0.95$. We compare two protected group-composition regimes: an unbalanced regime where the minority group represents 10% of S (and of C for $R^2\text{esPIR}$) and a balanced regime with equal group sizes. The horizontal axis varies the amount of unfairness above the tolerance ε .

(drawn from C and externally labeled), the probability of detection increases with the manipulated fraction m/N (according to (3), in which we substitute n by N to model the probability that a canary's output is modified). Moreover, when k is small compared to the audit size n , the auditor can include these canaries alongside the sampled audit set S without materially affecting the fairness estimation guarantees derived above.

Illustrative Example The manipulation difficulty of both the vanilla black-box audit and our proposed $R^2\text{esPIR}$ audit is illustrated in Figure 1 for a representative parameter setting. In particular, Figure 1a reports the number of output flips the provider must perform as a function of the amount of unfairness to hide, while Figure 1b shows the corresponding probability that the auditor detects such manipulation for different numbers of canaries included in the audit set S . As expected, protected group imbalance (left panels) substantially facilitates audit manipulation compared to the balanced case (right panels), as fewer output flips are needed to drive the observed disparity below the tolerance ε . In both regimes, $R^2\text{esPIR}$ notably increases the required number of flips m and, consequently, the probability of manipulation detection.

Intermediate Concentration Result Theorem 1 follows from Theorem 2, which formalizes the fact that the audit only observes the disparity through the sample-based estimate d_S . When S is sampled uniformly at random from C , d_S concentrates around d_C , with deviations controlled at scale $\sqrt{\ln(1/\delta)/n_{\min}}$. Consequently, to pass with high probability, the provider must keep its candidate-set gap at least this far *below* the fairness threshold.

Theorem 2 ($R^2\text{esPIR}$ audit: Sufficient candidate set condition for high-probability passing). *Let $n_{\min} := \min(n_0, n_1)$ and fix $\delta \in (0, 1)$. If the provider's candidate set gap satisfies*

$$|d_C| \leq \varepsilon - \sqrt{\frac{2 \ln(4/\delta)}{n_{\min}}},$$

then the provider passes the audit with probability at least $1 - \delta$:

$$\Pr(|d_S| \leq \varepsilon \mid n_0, n_1) \geq 1 - \delta.$$

Proof. Conditioning on (n_0, n_1) , we apply finite-population Hoeffding/Serfling bounds (Hoeffding 1963; Serfling 1974) to each group rate $p_{a,S}$, and then use the triangle inequality

and a union bound to obtain

$$\Pr(|d_S - d_C| \geq \gamma \mid n_0, n_1) \leq 4 \exp\left(-\frac{\gamma^2}{2} n_{\min}\right).$$

With $\gamma = \sqrt{2 \ln(4/\delta)/n_{\min}}$, this probability is at most δ . Hence, if $|d_C| \leq \varepsilon - \gamma$, then $|d_S| \leq \varepsilon$ with probability at least $1 - \delta$. See Appendix B for the detailed proof. $\square$

5 Experimental Evaluation

To illustrate and empirically validate our results, we selected three datasets spanning tabular and text domains: Default of Credit Card Clients (CCD) (Yeh and hui Lien 2009), COMPAS (Larson et al. 2016), and HateDay (Tonneau et al. 2025). We extracted numerical features from the HateDay text data using an off-the-shelf sentence transformer¹. For a fixed protected attribute, each dataset was randomly split 70/30 into a training set and a candidate set, using stratification with respect to the sensitive attribute. The model and data pre-processing is a histogram gradient boosted tree baseline from the `skrub` library². We tuned the learning rate, number of leaf nodes and number of estimators via random search with 5-fold cross-validation on the training set. After hyperparameter selection, we retrained the model on the full training set. Our source code is available in our online repository³, which includes all scripts and datasets required to reproduce our experiments and results, as well as our PIR-based auditing framework implemented as a user-friendly module.

Table 2 summarizes the main characteristics of each dataset. Specifically, for each dataset we report the model accuracy (Acc) and the demographic parity gap ($|d_{C,\text{true}}|$) before any manipulation, both measured on the candidate set C . We also report the candidate set size N , the audit set size n as well as the size of the smallest protected group in each set (respectively $N_{\min}$ and $n_{\min}$). As shown in the last column, some setups are highly imbalanced. For example, in HateDay the smallest protected group represents only about 8% of the candidate set.

We then simulate audit manipulation, in which a malicious provider attempts to hide half of its unfairness. That is, for each dataset and protected attribute, we set $\varepsilon = |d_{C,\text{true}}|/2$. We leverage Proposition 1 and Theorem 1 to compute the number of output modifications required to hide this amount of unfairness against a traditional (“vanilla”) black-box audit and against a $R^2\text{esPIR}$ -based audit. For $R^2\text{esPIR}$, the provider targets passing probability $1 - \delta = 80\%$. Finally, we plug these modification counts into Equation (3) to obtain the corresponding probability of manipulation detection in a setting where the auditor has a verification set of $k \in \{5, 10, 20, 50\}$ canaries (see Section 4.3). Note that Theorem 1 should be interpreted as an analytical characterization of manipulation difficulty, rather than as an

¹huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v2

²skrub-data.org/stable/reference/generated/skrub.tabular_pipeline.html

³github.com/sofianeazogagh/oblivious_audit

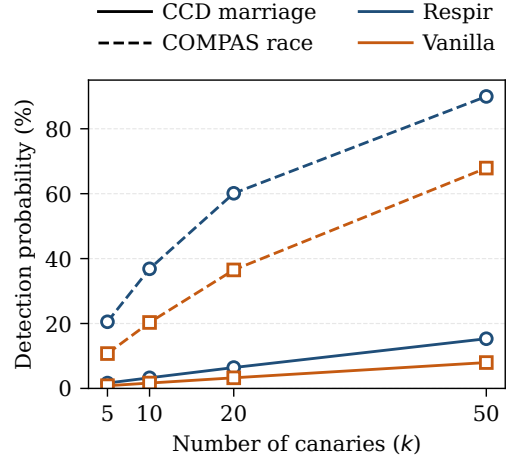


Figure 2: Probability of manipulation detection for a subset of our experiments, as a function of the number of canaries k , for vanilla audits and for our proposed $R^2\text{esPIR}$ audit framework.

operational recipe. Indeed, the bound depends on quantities that are not all known to the auditor, such as the platform’s true candidate-set disparity, and, under random sampling, on quantities not known in advance to the platform, such as the realized value of $n_{\min}$.

The results are summarized in Table 3, and displayed in Figure 2 for a subset of our experiments. As expected, the number of output manipulations required to hide unfairness increases substantially under $R^2\text{esPIR}$ compared to a traditional (“vanilla”) black-box audit. This, in turn, leads to systematically higher probabilities of detecting such manipulation, often by roughly a factor of two, which is particularly noteworthy since the final audit set size $n = |S|$ is identical for both audit frameworks. Comparing results across datasets and protected attributes and relating them to the dataset characteristics summarized in Table 2, two main factors drive detectability. First, protected group imbalance greatly facilitates manipulation. For instance, for the highly imbalanced HateDay setting, the probability of detecting manipulation remains very small even with $k = 50$ canaries. Second, larger unfairness levels to be hidden require more output flips, which increases detectability. For example, the highest detection probabilities for both audit frameworks occur for COMPAS with race as the protected attribute, which is also the configuration with the largest initial unfairness. As shown in Figure 2, the number of canaries owned by the auditor remains a crucial parameter, strongly influencing manipulation detectability for both methods.

It is also worth noting that when the tolerance ε is close to zero, and in particular when it is smaller than $\gamma = \sqrt{\frac{2 \ln(4/\delta)}{n_{\min}}}$ (which can occur, for instance, when $n_{\min}$ is small), the sufficient target

$$|d_C| \leq \varepsilon - \sqrt{\frac{2 \ln(4/\delta)}{n_{\min}}}$$

Dataset	Protected Attribute	Dataset Size	Acc	$ d_{C,\text{true}} $	N	n	$N_{\min}$	$n_{\min}$	$\frac{N_{\min}}{N}$
CCD	gender	30,000	0.823	0.028	13,500	6,750	5,350	2,675	0.396
	education		0.82	0.044			2,417	1,209	0.179
	marriage		0.821	0.009			6,147	3,073	0.455
	age		0.82	0.027			4,956	2,478	0.367
COMPAS	gender	6,172	0.66	0.148	2,778	1,389	529	264	0.19
	race		0.67	0.164			947	473	0.341
hateday	language	81,000	0.95	0.008	36,450	18,225	3,038	1,519	0.083

Table 2: Overview of the characteristics of the considered datasets.

Dataset	Protected Attribute	m_{vanilla}	$m_{\text{R}^2\text{esPIR}}$	k	$P_{\text{detect}}(\text{vanilla})$	$P_{\text{detect}}(\text{R}^2\text{esPIR})$
CCD	gender	38	152	5	2.23%	4.42%
				10	4.41%	8.65%
				20	8.63%	16.55%
				50	20.20%	36.39%
CCD	education	27	107	5	1.59%	3.13%
				10	3.15%	6.16%
				20	6.21%	11.95%
				50	14.81%	27.24%
CCD	marriage	14	56	5	0.83%	1.65%
				10	1.65%	3.27%
				20	3.27%	6.43%
				50	7.97%	15.31%
CCD	age	34	134	5	2.00%	3.91%
				10	3.96%	7.66%
				20	7.76%	14.74%
				50	18.28%	32.88%
COMPAS	gender	20	79	5	5.63%	10.87%
				10	10.94%	20.56%
				20	20.68%	36.89%
				50	43.97%	68.36%
COMPAS	race	39	156	5	10.74%	20.53%
				10	20.32%	36.85%
				20	36.51%	60.12%
				50	67.89%	89.96%
hateday	language	7	26	5	0.15%	0.28%
				10	0.31%	0.57%
				20	0.61%	1.14%
				50	1.52%	2.81%

Table 3: Manipulation cost and manipulation detection probability for each dataset and protected attribute. For each setting, we report the required number of output flips to hide half of the initial unfairness under a vanilla black-box audit and under R^2esPIR , together with the resulting detection probabilities for different numbers of canaries k .

for passing with probability at least $1 - \delta$ might become unattainable, since $|d_C| \geq 0$. In our experiments, such cases arise for some configurations in which $|d_{C,\text{true}}|$ is already very small. In these regimes, the provider may be unable to reach the desired passing confidence level $1 - \delta$ for the R^2esPIR audit. Nevertheless, we still report its best-effort manipulation, namely the number of output modifications required to achieve $|d_C| = 0$, which, as reflected in our results, still tends to make manipulation more detectable by the auditor.

Overall, these findings confirm the purpose of R^2esPIR . By hiding the audited set S through PIR, our protocol forces malicious providers to spread manipulations over a larger fraction of the candidate set, thereby increasing the effort required to hide unfairness and substantially improving the probability of detecting such manipulation.

6 Limitations, Discussion and Conclusion

6.1 Applicability of the Proposed Approach

Audited Model. Our proposed framework, as well as the accompanying theoretical analysis, is agnostic to the type of audited model. Indeed, the provider’s model class matters only if the provider must realize flips by retraining a new classifier within the same hypothesis space. Our adversary model is more powerful as the provider can patch the deployed system by overriding outputs on selected inputs without retraining. This is at the same time realistic but also a worst-case situation for the auditor. If the provider was not able to flip selected answers without a full model retraining, arbitrary flips might be infeasible and the provider could need more changes to hide unfairness, so our lower bounds (Proposition 1 and Theorem 1, holding for any input-output mapping) would remain valid but potentially conservative.

Audited Metric. While in this paper, we focus on the demographic parity metric, our approach naturally extends to auditing other model properties. Indeed, the PIR-based component treats the fairness computation as a modular measurement block, so our proposed auditing framework is metric-agnostic. In particular, Proposition 1 and Theorem 1 can be straightforwardly adapted to other fairness criteria (including separation- and sufficiency-based metrics, which encompasses but is not limited to group fairness metrics based on true positive rates (equal opportunity), false positive rates (predictive equality) or both (equalized odds)), by incorporating the proper sensitivity value suited to the audited measure. Similar guarantees regarding resilience to manipulation would then hold, as illustrated for the equal opportunity metric in Appendix C. On the same line, our framework and guarantees can also be applied to setups with several sensitive attributes or multi-class problems, as long as the sensitivity of the audited function can be analytically expressed or bounded (as in the proofs of Proposition 1 and Theorem 1), key to lower bounding the number of flips required to fool the audit. We therefore expect the same modularity could support regression as well, though fairness for regression has been less explored in the literature.

6.2 Discussion

We studied the problem of auditing fairness claims in adversarial settings, in which an audited provider may adapt its outputs once it can anticipate which inputs will be inspected. Our main contribution is $R^2\text{esPIR}$, an oblivious auditing protocol that leverages PIR to hide the auditor’s queries within a larger candidate set. By construction, this prevents the provider from patching a known audit set. Instead, passing with high probability requires spreading modifications over a nontrivial fraction of the candidate set. We formalized this intuition through concentration-based guarantees that lower bound the manipulation effort needed to pass, and we combined these bounds with a canary-based verification mechanism to translate manipulation effort into a detection probability. Our experiments on tabular and text datasets confirms that for a fixed audit set size, $R^2\text{esPIR}$ increases the number of required output flips and, consequently, improves the detectability of targeted manipulation, with the strongest gains arising when the candidate set is large and sufficiently balanced across protected groups.

Nevertheless, as illustrated by our experiments, strong protected group imbalance and small audit sets remain limiting factors. The manipulation guarantees in Section 4.3 and their empirical evaluation in Section 5 highlight three main design principles:

1. *Large and representative candidate sets.* As the best-case manipulation cost scales with $\min(N_0, N_1)$, the candidate set should be large and should contain nontrivial representation of each protected group.
2. *Balanced auditing.* Since detection power scales with $\min(n_0, n_1)$, the auditor should sample S using a rule that avoids severe group imbalance; stratified sampling from C is a natural option.
3. *Hiding the audit set is necessary but not always sufficient.* PIR prevents the provider from learning *which* queries are in the audit set, blocking the cheapest form of manipulation (patching exactly S , as in Proposition 1). However, our guarantees rely on the candidate set capturing a meaningful portion of the provider’s operational domain. If C is not representative, then passing the audit may not translate into improved fairness outside C .

Limitations. We view $R^2\text{esPIR}$ as a proof-of-concept showing how cryptographic primitives can be leveraged to hide the actual queries used during an algorithmic audit. Its main benefit is to prevent audit set adaptivity: the platform can no longer tailor its responses to the particular audit set S , but must instead commit to outputs over a larger candidate set C . This makes manipulation more costly and, when combined with external verification queries, more detectable. Nonetheless, $R^2\text{esPIR}$ also has inherent limitations.

First, because the candidate set C is known to both the platform and the auditor, the protocol does not prevent all forms of manipulation. A strategic platform can still attempt to modify its outputs over C so as to satisfy the audited fairness criterion with high probability. A stronger alternative would be to require a private-inference API, for instance based on verifiable homomorphic encryption or on hybrid

cryptographic protocols, allowing the auditor to evaluate the committed model on private inputs without revealing the audited queries to the platform. Conceptually, this would remove the need for an explicit shared candidate set, since the auditor’s query domain would no longer have to be enumerated in advance. In practice, however, such a requirement remains restrictive. Encrypted inference typically incurs substantial computational and communication overhead, and often requires model-specific adaptations such as polynomial approximations, quantization, or hybrid protocols for non-linear operations (Gilad-Bachrach et al. 2016; Brutzkus, Gilad-Bachrach, and Elisha 2019; Juvekar, Vaikuntanathan, and Chandrakasan 2018; Mishra et al. 2020; Gong et al. 2024), not to mention the verification step which introduces additional overhead. Therefore, we view the use of a fixed candidate set as a pragmatic compromise between limiting the platform’s computational burden, ensuring deployment feasibility, and preserving audit usefulness.

Second, manipulation detectability could be further enhanced by using an iterative protocol, in which the auditor progressively requires the platform to label the candidate set C , in an order designed to trigger its worst-case manipulation bounds. For instance, the auditor could first require the platform to label the smallest protected group, thereby constraining the platform’s subsequent manipulation strategy and potentially forcing its final flips to be performed on the largest subgroup, resulting in more changes and higher detectability. This would prevent the platform from reaching its lower-cost manipulation bounds, making manipulation more difficult in practice. However, such an extension would need to be designed carefully to preserve the non-adaptivity property of $R^2\text{esPIR}$; otherwise, the iterative structure itself could reveal information that the platform may exploit strategically.

6.3 Conclusion

Overall, $R^2\text{esPIR}$ is a step toward manipulation-resistant audits combining cryptographic query hiding with probabilistic guarantees. We hope this work will motivate further research at the intersection of auditing, cryptography and trustworthy machine learning, and inform the design of regulatory audits robust to adversarial environments.

Several directions remain open. A first avenue is to extend the analysis beyond demographic parity to separation- and sufficiency-based criteria, as well as to settings with continuous outputs or multi-class protected attributes. A second direction is to study candidate-set design in greater depth, including how to construct representative candidate sets under limited data access, how to allocate audit budgets across groups to mitigate imbalance, and how to choose the confidence parameter to balance robustness and audit cost. Finally, from a systems perspective, deploying $R^2\text{esPIR}$ in real auditing pipelines calls for exploring the performance trade-offs of different PIR instantiations, robust engineering under rate limits and latency constraints and protocol mechanisms for logging and accountability that remain compatible with query privacy.

Acknowledgments

This research was enabled in part by funding from the SCALE-AI Chair in Data-Driven Supply Chains as well as by the *Fonds de recherche du Québec – Nature et technologies (FRQNT)* through a Team Research Project (327090). Sébastien Gambs is supported by the Canada Research Chair program (CRC on Privacy-preserving and Ethical Analysis of Big Data) and a Discovery Grant from NSERC. The authors would like to thank the anonymous reviewers for their valuable suggestions.

References

- 2022a. Regulation (EU) 2022/1925 of the European Parliament and of the Council of 14 September 2022 on Contestable and Fair Markets in the Digital Sector and Amending Directives (EU) 2019/1937 and (EU) 2020/1828 (Digital Markets Act) (Text with EEA Relevance).
- 2022b. Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services and Amending Directive 2000/31/EC (Digital Services Act) (Text with EEA Relevance).
2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying down Harmonised Rules on Artificial Intelligence and Amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA Relevance).
- Ahmad, I.; Yang, Y.; Agrawal, D.; El Abbadi, A.; and Gupta, T. 2021. Addra: Metadata-private voice communication over fully untrusted infrastructure. In *Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation, OSDI*, 313–329.
- Aïvodji, U.; Arai, H.; Fortineau, O.; Gambs, S.; Hara, S.; and Tapp, A. 2019. Fairwashing: the risk of rationalization. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 161–170.
- Ali, A.; Lepoint, T.; Patel, S.; Raykova, M.; Schoppmann, P.; Seth, K.; and Yeo, K. 2021. Communication–Computation trade-offs in PIR. In *Proceedings of the 30th USENIX Security Symposium*, 1811–1828.
- Anders, C. J.; Pasliev, P.; Dombrowski, A.; Müller, K.; and Kessel, P. 2020. Fairwashing explanations with off-manifold detergent. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, 314–323.
- Angel, S.; Chen, H.; Laine, K.; and Setty, S. 2018. PIR with compressed queries and amortized query processing. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy, SP*, 962–979.
- Birhane, A.; Steed, R.; Ojewale, V.; Vecchione, B.; and Raji, I. D. 2024. AI auditing: The Broken Bus on the Road to AI Accountability. In *Proceedings of the IEEE Conference on Secure and Trustworthy Machine Learning, SaTML*, 612–643.
- Bourrée, J. G.; Merrer, E. L.; Trédan, G.; and Rottembourg, B. 2023. On the relevance of APIs facing fairwashed audits. *arXiv preprint arXiv:2305.13883*.
- Brutzkus, A.; Gilad-Bachrach, R.; and Elisha, O. 2019. Low Latency Privacy Preserving Inference. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, 812–821.
- Chandrasekaran, V.; Jia, H.; Thudi, A.; Travers, A.; Yaghini, M.; and Papernot, N. 2021. SoK: Machine Learning Governance. *arXiv preprint arXiv:2109.10870*.
- Chor, B.; Kushilevitz, E.; Goldreich, O.; and Sudan, M. 1998. Private Information Retrieval. *Journal of the ACM*, 45(6): 965–981.
- Corrigan-Gibbs, H.; Henzinger, A.; and Kogan, D. 2022. Single-server private information retrieval with sublinear amortized time. In *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 3–33.
- Corrigan-Gibbs, H.; and Kogan, D. 2020. Private information retrieval with sublinear online time. In *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 44–75.
- Costanza-Chock, S.; Raji, I. D.; and Buolamwini, J. 2022. Who Audits the Auditors? Recommendations from a field scan of the algorithmic auditing ecosystem. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT*, 1571–1583.
- Davidson, A.; Pestana, G.; and Celi, S. 2023. FrodoPir: Simple, scalable, single-server private information retrieval. *Proceedings on Privacy Enhancing Technologies*.
- de Castro, L.; and Lee, K. 2024. VeriSimplePIR: Verifiability in SimplePIR at No Online Cost for Honest Servers. In *Proceedings of the 33rd USENIX Security Symposium*, 5931–5948.
- Entrena-Serrano, C.; Degeling, M.; Romano, S.; and Çetin, R. B. 2025. TikTok’s Research API: Problems Without Explanations. *arXiv preprint arXiv:2506.09746*.
- European Commission. 2025. Commission fines X €120 million under the digital services act.
- Fokkema, H.; de Heide, R.; and van Erven, T. 2023. Attribution-based Explanations that Provide Recourse Cannot be Robust. *Journal of Machine Learning Research*, 24: 360:1–360:37.
- Franzese, O.; Shamsabadi, A. S.; Luck, C.; and Haddadi, H. 2025. Secure and Confidential Certificates of Online Fairness. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems, NeurIPS*.
- Fukuchi, K.; Hara, S.; and Maehara, T. 2020. Faking Fairness via Stealthily Biased Sampling. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI*, 412–419.
- Garcia Bourrée, J.; Godinot, A.; Biswas, S.; Kermarrec, A.-M.; Le Merrer, E.; Tredan, G.; Vos, M. D.; and Vujasinovic, M. 2025. Robust ML Auditing Using Prior Knowledge. In *Proceedings of the 42nd International Conference on Machine Learning, ICML*, 18794–18810.

- Gilad-Bachrach, R.; Dowlin, N.; Laine, K.; Lauter, K.; Naehrig, M.; and Wernsing, J. 2016. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In *Proceedings of the 33rd International Conference on Machine Learning, ICML*, 201–210.
- Gong, Y.; Chang, X.; Mišić, J.; Mišić, V. B.; Wang, J.; et al. 2024. Practical solutions in fully homomorphic encryption: a survey analyzing existing acceleration methods. *Cybersecurity*, 7(5).
- Henzinger, A.; Hong, M. M.; Corrigan-Gibbs, H.; Meiklejohn, S.; and Vaikuntanathan, V. 2023. One server for the price of two: Simple and fast Single-Server private information retrieval. In *Proceedings of the 32nd USENIX Security Symposium*, 3889–3905.
- Hoeffding, W. 1963. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301): 13–30.
- Horwitz, J. 2025. Meta Created ‘Playbook’ to Fend off Pressure to Crack down on Scammers, Documents Show. *Reuters*.
- Hutchinson, B.; Rostamzadeh, N.; Greer, C.; Heller, K.; and Prabhakaran, V. 2022. Evaluation gaps in machine learning practice. In *Proceedings of the 2022 ACM conference on fairness, accountability, and transparency, FAccT*, 1859–1876.
- Juvekar, C.; Vaikuntanathan, V.; and Chandrakasan, A. 2018. GAZELLE: A Low Latency Framework for Secure Neural Network Inference. In *Proceedings of the 27th USENIX Security Symposium*, 1651–1669.
- Kogan, D.; and Corrigan-Gibbs, H. 2021. Private blocklist lookups with checklist. In *Proceedings of the 30th USENIX Security Symposium*, 875–892.
- Larson, J.; Mattu, S.; Kirchner, L.; and Angwin, J. 2016. How We Analyzed the COMPAS Recidivism Algorithm. *ProPublica*.
- Melchor, C. A.; Barrier, J.; Fousse, L.; and Killijian, M. 2016. XPIR : Private Information Retrieval for Everyone. *Proceedings on Privacy Enhancing Technologies*, 155–174.
- Menon, S. J.; and Wu, D. J. 2022. Spiral: Fast, high-rate single-server PIR via FHE composition. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy, SP*, 930–947.
- Metaxa, D.; Park, J. S.; Robertson, R. E.; Karahalios, K.; Wilson, C.; Hancock, J. T.; and Sandvig, C. 2021. Auditing Algorithms: Understanding Algorithmic Systems from the Outside In. *Foundations and Trends® in Human-Computer Interaction*, 14(4): 272–344.
- Mishra, P.; Lehmkuhl, R.; Srinivasan, A.; Zheng, W.; and Popa, R. A. 2020. Delphi: A Cryptographic Inference Service for Neural Networks. In *Proceedings of the 29th USENIX Security Symposium*, 2505–2522.
- Mughees, M. H.; Chen, H.; and Ren, L. 2021. OnionPIR: Response efficient single-server PIR. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2292–2306.
- Olumofin, F. G.; and Goldberg, I. 2011. Revisiting the Computational Practicality of Private Information Retrieval. In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security, FC*, 158–172.
- Park, S.; Kim, S.; and Lim, Y.-s. 2022. Fairness audit of machine learning models with confidential computing. In *Proceedings of the ACM Web Conference 2022*, 3488–3499.
- Pearson, G. D.; Silver, N. A.; Robinson, J. Y.; Azadi, M.; Schillo, B. A.; and Kreslake, J. M. 2024. Beyond the margin of error: a systematic and replicable audit of the TikTok research API. *Information, Communication & Society*, 28(3): 452–470.
- Pentyala, S.; Melanson, D.; Cock, M. D.; and Farnadi, G. 2022. PrivFair: a Library for Privacy-Preserving Fairness Auditing. *arXiv preprint arXiv:2202.04058*.
- Regev, O. 2009. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6): 1–40.
- Ribeiro, M. H. 2024. Is Facebook “Standard” Algorithm Polarizing? <https://doomscrollingbabel.manoel.xyz/p/is-facebook-standard-algorithm-polarizing>.
- Serfling, R. J. 1974. Probability Inequalities for the Sum in Sampling without Replacement. *The Annals of Statistics*, 2(1): 39–48.
- Shamsabadi, A. S.; Wyllie, S. C.; Franzese, N.; Dullerud, N.; Gambs, S.; Papernot, N.; Wang, X.; and Weller, A. 2023. Confidential-PROFIT: Confidential PROof of FaIr Training of Trees. In *Proceedings of the 11th International Conference on Learning Representations, ICLR*.
- Shamsabadi, A. S.; Yaghini, M.; Dullerud, N.; Wyllie, S. C.; Aivodji, U.; Alaagib, A.; Gambs, S.; and Papernot, N. 2022. Washing The Unwashable : On The (Im)possibility of Fairwashing Detection. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems, NeurIPS*.
- Shi, E.; Aqeel, W.; Chandrasekaran, B.; and Maggs, B. 2021. Puncturable pseudorandom sets and private information retrieval with near-optimal online bandwidth and time. In *Proceedings of the Annual International Cryptology Conference*, 641–669.
- Smith, S. W.; and Safford, D. 2001. Practical server privacy with secure coprocessors. *IBM Systems Journal*, 40(3): 683–695.
- The European Advisory Committee Social Science One. 2019. Public Statement from the Co-Chairs and European Advisory Committee of Social Science One.
- Thomas, R. L.; and Uminsky, D. 2022. Reliance on metrics is a fundamental challenge for AI. *Patterns*, 3(5).
- Timberg, C. 2021. Facebook made big mistake in data it provided to researchers, undermining academic work.
- Tonneau, M.; Liu, D.; Malhotra, N.; Hale, S. A.; Fraiberger, S.; Orozco-Olvera, V.; and Röttger, P. 2025. HateDay: Insights from a Global Hate Speech Dataset Representative of a Day on Twitter. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2297–2321.

Vecchione, B.; Levy, K.; and Barocas, S. 2021. Algorithmic auditing and social justice: Lessons from the history of audit studies. In *Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, 1–9.

Waiwitlikhit, S.; Stoica, I.; Sun, Y.; Hashimoto, T.; and Kang, D. 2024. Trustless Audits without Revealing Data or Models. In *Proceedings of the 41st International Conference on Machine Learning, ICML*, 49808–49821.

Yadav, C.; Moshkovitz, M.; and Chaudhuri, K. 2024. XAudit : A Learning-Theoretic Look at Auditing with Explanations. *Transactions on Machine Learning Research*.

Yan, T.; and Zhang, C. 2022. Active fairness auditing. In *Proceedings of the 39th International Conference on Machine Learning, ICML*, 24929–24962.

Yeh, I.-C.; and hui Lien, C. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36: 2473–2480.