Δ-AUDITS: ADAPTIVE PERFORMANCE MONITORING

Anonymous authors
Paper under double-blind review

ABSTRACT

Evaluations are the cornerstone of ML governance. Audits of ML-enabled systems are conventionally treated by regulators and academics as one-shot events: a single audit is conducted, its outcome is certified, and the system is presumed compliant until a fresh audit is commissioned. However, deployed systems are updated regularly, often without the auditor's knowledge, which can invalidate the audit's conclusions. To help regulators choose when to commission a fresh audit, we introduce Δ-*Audits*, a setting that pairs the initial white-box audit with a subsequent lower-access black-box monitoring phase. We instantiate monitoring with a fixed-budget disagreement test parameterized by a query distribution. We use the initial white-box inspection phase to optimize this distribution through a bi-level formulation that anticipates a provider maximizing its utility under the monitoring constraint. The goal is to limit deterioration in auditor utility while leaving the provider room to improve its own utility. Experiments on tabular and image classification tasks evaluate manipulation gain under uniform and optimized monitoring constraints. Optimized monitoring detects harmful updates with an order of magnitude fewer queries than uniform or performance-based monitoring, and sharply reduces the utility a provider can gain by manipulating the audit, whereas uniform monitoring leaves this gain almost intact.

1 INTRODUCTION

AI audits are an integral part of ensuring trustworthiness (Raji et al., 2020) and safety (Perez et al., 2022) at release. A thorough inspection of the provider's training pipeline (i.e., a *white-box* audit) can show that the provider is releasing a trustworthy model. But given the frequent model updates, the audit result may soon be invalidated. Providers may update the model with benign or adversarial intentions. Evidence from social platform suggests both happen. During a Meta–academic study of feed algorithms (Guess et al., 2023), Meta deployed emergency changes to the ranking algorithm to curb election misinformation, which confounded the study's results (Bagchi et al., 2024). In contrast, while under regulatory scrutiny over scam ads, Meta reportedly removed only the ads that surfaced in regulators' known queries instead of improving its scam filtering overall (Horwitz, 2025).

Regardless of provider intentions, when models are updated and audits are invalidated, a new audit is required. But given the frequency of such updates, performing an inspection each time is economically infeasible. This motivates *monitoring*, as a cheaper audit but with significantly less access to the model (i.e., black-box access). Our key research question is: *How can an auditor use its white-box access during inspection to design effective black-box monitoring?*

We introduce Δ-Audit, an auditing regime that anticipates post-inspection updates at inspection time. A Δ-Audit pairs an initial white-box inspection phase, in which the auditor enjoys full model access, with a subsequent black-box monitoring phase under strictly limited access. During inspection, the auditor uses its access to the model and information about the provider's objective to construct a query distribution for subsequent monitoring.

We formalize monitoring as as a hypothesis test parametrized by a monitoring distribution $\rho \in \Delta^n$ over the audit dataset D of size n. The baseline monitoring strategy is to uniformly sample from D and check the rate of monitored model disagreements with the originally inspected model. But this strategy does not leave room for provider improvements (e.g. benign updates). It is also not necessarily query efficient for the auditor who may place more importance on certain queries than others (e.g. a fairness auditor cares more about performance degradations for minorities).

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.

We introduce `Twister`, a bi-level optimizer to find the sampling distribution ρ. In the inner level the provider who knows ρ seeks to maximizes its utility. In the outer level, the auditor chooses weights to limit the resulting deterioration in its utility. Experiments on classical tabular and image datasets highlight the query efficiency gain and manipulation deterrence achieved by `Twister`.

Our contributions are:

1. We formulate Δ-Audits as inspection followed by query-limited monitoring, instantiated with a standard disagreement test parameterized by a sampling distribution (Section 2).
2. We introduce a bi-level formulation for optimizing that distribution to protect auditor utility against a rational provider, and a practical optimization procedure, called `Twister` (Section 3).
3. We evaluate the resulting monitoring designs through provider manipulation gain across scenarios with conflicting auditor and provider utilities (Section 4). `Twister` detects harmful updates with an order of magnitude fewer queries than uniform or performance-based monitoring, at the price of more false alarms on large beneficial updates. It also keeps the provider's manipulation gain below 0.01 on tabular data and at strict audit thresholds on images, whereas uniform monitoring leaves it close to that of an unmonitored provider.

Related work. Statistical monitoring methods detect changes in performance or behavior Chugg et al. (2023); Feng et al. (2024); Richter et al. (2024), while model fingerprinting and change detection identify deployed model instances or modifications Richardeau et al. (2026); Chauvin et al. (2026). Other approaches protect audits through proof-of-inference techniques Franzese et al. (2024), comparison of multiple data sources Garcia Bourrée et al. (2027; 2025), or trusted evaluations Beers & Toner (2025). Our focus is the strategic design of monitoring queries using inspection access: we choose a distribution according to auditor utility while anticipating a provider's utility-maximizing response. Detailed related work appears in Section A.

2 Monitoring as a Hypothesis Test

Let h_0, h_m be a pair of models that take a query x (e.g. an ad) and produces a decision y (is scam, or not). h_0 is the model that has received an inspection and is known to be trustworthy. h_m is the deployed, potentially updated, model which we seek to monitor in a black-box fashion.

Let $D = \{(x_i, y_i)\}_{i=1}^n$ be the audit dataset, and $\rho \in \Delta^n = \{\rho \in \mathbb{R}_+^n : \sum_i \rho_i = 1\}$ the *query distribution* defined over the audit points. For the deployed model h_m, its weighted disagreement with the inspected model is

$$\delta_\rho(h_m) = \textstyle\sum_{i=1}^n \rho_i \mathbb{1}[h_m(x_i) \neq h_0(x_i)]. \tag{1}$$

The monitoring procedure tests

$$H_0 : \delta_\rho(h_m) \leq \gamma \qquad \text{against} \qquad H_1 : \delta_\rho(h_m) > \gamma, \tag{2}$$

where $\gamma \in (0, 1)$ is a disagreement tolerance, distinct from the inspection requirement τ.

For a fixed query budget b, draw $I_1, \ldots, I_b$ independently from ρ with replacement and count disagreements. Each indicator $\mathbb{1}[h_m(x_{I_t}) \neq h_0(x_{I_t})]$ is independent and by definition has probability $\delta_\rho(h_m)$, thus

$$S_b = \textstyle\sum_{t=1}^b \mathbb{1}[h_m(x_{I_t}) \neq h_0(x_{I_t})] \sim \mathrm{Binomial}\big(b, \delta_\rho(h_m)\big). \tag{3}$$

This assumes deterministic predictions and a deployed model fixed during the monitoring run; ρ is chosen before the run, and the provider may know it when choosing h_m. The standard one-sided exact binomial test rejects when $\Pr\{S_b > k_b\} \leq \alpha$, where $B \sim \mathrm{Binomial}(b, \gamma)$. Given a desired disagreement tolerance γ and false-alarm constraint α, we present in Section 3.4 how the auditor can calibrate its decision threshold k_b.

Uniform monitoring and its limitation. The simplest baseline uses $\rho_i = 1/n$, so every changed prediction consumes the same share of the tolerance. However, this has multiple limitations. On one hand, when the provider's goals conflict with the auditor's, uniform monitoring can be too lenient. A provider that stops filtering a profitable category of scam ads changes only a small fraction of

predictions; if that fraction is below γ, the update passes, however much harm those ads cause. On the other hand, when the goals align, it can be too strict. A provider that retrains and now catches scams that h_0 missed makes changes the auditor welcomes, or at least does not mind; if they exceed a fraction γ of the audit set, the update is flagged all the same. In both cases, disagreement measures how much the model has changed, not whether the change matters to the auditor.

The query distribution ρ determines which changes consume the tolerance: a change on a low-weight point costs the provider little. Which points deserve weight therefore depends on what the auditor values and on which changes the provider is inclined to make.

3 Δ-AUDIT: MONITORING A RATIONAL PROVIDER

In the scam-ad example, the two parties judge the same model by different standards. The auditor cares about scams that reach users, weighted by the harm they cause. The provider cares about revenue: blocking a legitimate ad loses a paying customer, and some scam categories are themselves profitable. We capture each party's judgment with a *utility function*. The auditor scores a model h by $U_{\text{aud}}(h, D)$ on the audit set, and inspection requires $U_{\text{aud}}(h_0, D) \geq \tau$. The provider scores it by $U_{\text{prov}}(h, D_{\text{train}})$ on its own training data, and seeks to maximize it.

Example. A simple instance is weighted accuracy, with the auditor weighting points proportional to harm to users and the provider proportional to revenue ($\bar{w}^{\text{aud}}$ and $\bar{w}^{\text{prov}}$ positive and normalized):

$$U_{\text{aud}}(h, D) = \sum_{i=1}^{n}, w_i^{\text{aud}}\, \mathbb{1}[h(x_i) = y_i], \quad U_{\text{prov}}(h, D_{\text{train}}) = \sum_{(x_i, y_i) \in D_{\text{train}}} w_i^{\text{prov}}\, \mathbb{1}[h(x_i) = y_i]. \quad (4)$$

When the two weightings disagree, meeting the requirement costs the provider utility. Some of these updates hurt the auditor and some do not, and a uniform ρ cannot tell them apart.

We can choose the query distribution ρ such that monitoring would block the updates that hurt the auditor and leave room for the rest:

Definition 1 (Δ-Audit)**.** *A Δ-Audit with requirement τ, tolerance γ, query budget b, and level α proceeds in two phases.*

1. Inspection (white-box). *The auditor verifies $U_{aud}(h_0, D) \geq \tau$, records $h_0(D)$, and chooses $\rho \in \Delta^n$ using its access to the provider's model, code, and training data.*
2. Monitoring (black-box). *Using only D, $h_0(D)$, and query access to the served model h_m, the auditor periodically runs the test* (2) *with b queries drawn from ρ. A rejection triggers a fresh inspection.*

3.1 THE PROVIDER'S RESPONSE

After passing inspection, the provider has reason to recover the utility it gave up to meet the requirement. We model it as rational (i.e., utility-maximizing). It knows the audit set D, the inspected model h_0, the monitoring distribution ρ, and the tolerance γ, and it updates its model to maximize its own utility while staying within the monitoring test's null hypothesis:

$$\max_h \; U_{\text{prov}}(h, D_{\text{train}}) \quad \text{s.t.} \quad \delta_\rho(h) \leq \gamma, \quad (5)$$

where the maximum is over the provider's admissible model class. This is an idealization: satisfying the constraint is not the same as passing a particular finite-query run, and for fixed ρ and γ the query budget affects detection but not this response. Because the auditor has access to the provider's training data and procedure during inspection, it can simulate this response for any candidate ρ.

To measure how much the provider gains from updating, we compare its response with the inspected model h_0. Formally, we take h_0 to be the provider's best model that passes inspection,

$$h_0 \in \arg\max_h \; U_{\text{prov}}(h, D_{\text{train}}) \quad \text{s.t.} \quad U_{\text{aud}}(h, D) \geq \tau, \quad (6)$$

so that any further gain comes from the post-inspection update.

Algorithm 1 `Twister`: Query Distribution Design by Bi-Level Optimization. Solving (8) with L_{prov}, C, and F defined in Section 3.3. $\mathrm{sd}(\psi)$ is the standard deviation of the entries of ψ. Core updates; practical variants use warmup on ρ, damping on v, and stochastic derivative estimates.

Require: h_0 with parameters θ_0; auditor data D; provider data D_{train}, penalty λ; step sizes η, β; iterations T

$v_0 \leftarrow 0, \quad \rho_0 \leftarrow \mathbf{1}/n$

for $t = 0, \ldots, T-1$ **do**

▷ (i) Provider step: response to ρ_t

$\theta_{t+1} \leftarrow \theta_t - \eta \left[\nabla_\theta L_{\text{prov}}(\theta_t) + \lambda \nabla_\theta C(\theta_t, \rho_t)\right]$

▷ (ii) Sensitivity of the response to ρ

$v_{t+1} \leftarrow v_t - \eta \left[(\nabla^2_{\theta\theta} L_{\text{prov}}(\theta_t) + \lambda \nabla^2_{\theta\theta} C(\theta_t, \rho_t))\, v_t + \nabla_\theta F(\theta_t, \rho_t)\right]$

▷ (iii) Auditor step: exponentiated gradient on ρ

$\psi_t \leftarrow \nabla_\rho F(\theta_t, \rho_t) + \lambda \nabla^2_{\rho\theta} C(\theta_t, \rho_t)\, v_t$

$\rho_{t+1} \leftarrow \mathrm{softmax}\big(\log \rho_t - \beta\, \psi_t / \mathrm{sd}(\psi_t)\big)$

return the sampling distribution ρ_T

Definition 2 (Manipulation gain)**.** *For a monitoring distribution ρ, let h_m solve* (5)*. The manipulation gain is*

$$\mathrm{MG}(\tau, \gamma; \rho) = U_{prov}(h_m, D) - U_{prov}(h_0, D),$$

and the auditor's cost is $\Lambda(h_0, h_m) = (U_{aud}(h_0, D) - U_{aud}(h_m, D))^+$ with $(a)^+ = \max(a, 0)$.

Since $\delta_\rho(h_0) = 0$, the inspected model is feasible for (5), so the provider's gain is nonnegative under exact optimization; on the audit set D, where we report it, it can differ. We report the two quantities together. Manipulation gain measures how much the provider gains by updating, and the auditor's cost measures how much of that gain comes at the auditor's expense.

3.2 Optimizing The Monitoring Distribution

The auditor commits to a distribution ρ and the provider responds by maximizing its utility under the resulting constraint. This forms a Stackelberg game (Fudenberg & Tirole, 1991), which we express as the following bi-level problem. The auditor moves first by committing to a monitoring strategy ρ. Then the provider plays its best response maximizing utility while passing the audit (Equation (5)). The auditor solves the outer problem to reduce the deterioration $\Lambda(h_0, h_m)$ of its utility compared to the inspected model while being indifferent to any improvement. To amplify the cost of undetected deterioration, we modulate $\Lambda(h_0, h_m)$ by the single-draw agreement probability $1 - \delta_\rho(h^*)$. In the end, problem the auditor has to solve is

$$\begin{aligned} \min_{\rho \in \Delta^n} \quad & (U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h^*, D))^+ \big(1 - \delta_\rho(h^*)\big) \\ \text{s.t.} \quad & h^* \in \arg\max_h \quad U_{\text{prov}}(h, D_{\text{train}}) \\ & \qquad\qquad \text{s.t.} \quad \delta_\rho(h) \le \gamma \end{aligned} \tag{7}$$

By later enforcing the $\delta_\rho(h) \le \gamma$ agreement constraint via the monitoring test, the auditor effectively shapes the incentives of the provider from pure $U_{\text{prov}}(h, D_{\text{train}})$ maximization to having to take U_{aud} into account. An update that improves auditor utility has zero outer cost even if it changes predictions. Thus the objective designs constraints on a rational provider's updates rather than simply maximizing disagreement or test power. An alternative query-subset selection interpretation of the bilevel problem Equation (7) is given in Section C.

3.3 `Twister`: A Bi-Level Solver for Query Distribution

The bi-level optimization problem (7) in its exact form is challenging due to the non-differentiable indicators and its bi-level nature where finding a solution is equivalent to searching for equilibria (i.e. saddle points) of the problem. Moreover, common techniques do not apply to the current form of Problem (7) because the inner problem itself is a constrained optimization problem. To make the problem tractable, we introduce an approximate solver which we coin `Twister`, and present in Algorithm 1.

`Twister` makes two simplifying changes to Problem (7). First, to make the problem differentiable in the query distribution ρ and model parameters θ, we approximate the constrained problem using differentiable surrogates for indicator functions. Each indicator function is replaced by the probability of the same event, where $p_\theta(c \mid x)$ is the model's softmax probability of class c. Second, to use off-the-shelf stochastic bilevel solvers, we approximate the inner constrained optimization problem instead of solving it directly:

$$\begin{aligned} \min_{\rho \in \Delta^n} \quad & F\big(\theta^*(\rho), \rho\big) \\ \text{s.t.} \quad & \theta^*(\rho) \in \arg\min_{\theta} \; L_{\text{prov}}(\theta) + \lambda\, C(\theta, \rho). \end{aligned} \tag{8}$$

θ denotes the provider's model parameters, $\theta^*(\rho)$ its penalized response to ρ, and $\lambda > 0$ the penalty weight. The *surrogate outer objective*

$$F(\theta, \rho) = \big(U_{\text{aud}}(h_0, D) - \widetilde{U}_{\text{aud}}(\theta)\big)^+ \big(1 - C(\theta, \rho)\big)$$

is the outer objective of Equation (7) with $\delta_\rho(h^*)$ and $U_{\text{aud}}(h^*, D)$ replaced by differentiable surrogates. The *soft disagreement* $C(\theta, \rho) = 1 - \sum_{i=1}^n \rho_i p_\theta(h_0(x_i) \mid x_i)$, and the *soft auditor utility* $\widetilde{U}_{\text{aud}}(\theta) = \sum_{i=1}^n, w_i^{\text{aud}}\, \mathbb{1}[h(x_i) = y_i]$. The inner problem uses the *provider loss* $L_{\text{prov}}(\theta) = \sum_{(x_i, y_i) \in D_{\text{train}}} \bar{w}^{\text{prov}}\, \text{CE}\big(f_\theta(x_k), y_k^{\text{prov}}\big)$.

To solve Problem (8), we adapt the single-loop bi-level solver of Dagréou et al. (2022) (Algorithm 1). Starting from the inspected parameters θ_0 and a uniform ρ, each iteration makes three updates:

(i) *Provider step:* a gradient step on the inner objective $L_{\text{prov}}(\theta) + \lambda C(\theta, \rho)$, which tracks the provider's response $\theta^*(\rho)$.

(ii) *Sensitivity step:* This is an intermediate step to update the auxiliary vector v that captures how the provider's response moves with ρ. This is achieved by implicit differentiation of the linearization of the inner problem. Importantly, this step requires only Hessian-vector products $\nabla^2_{\theta\theta} L_{\text{prov}}(\cdot) v_t$ and $\nabla^2_{\theta\theta} C(\cdot)\, v_t$ which is more compute-efficient than calculation of the Hessian, directly.

(iii) *Auditor step:* the resulting hypergradient ψ, the derivative of $F(\theta^*(\rho), \rho)$ with respect to ρ, drives an exponentiated-gradient step $\rho_i \propto \rho_i \exp(-\beta \psi_i)$. This multiplicative update keeps ρ in Δ^n without projection ; we divide ψ by its standard deviation so that the step size β does not depend on the scale of ψ.

Remark. The penalized problem is not an exact reformulation of the hard constraint: λ takes the role of the tolerance γ, and a larger λ yields responses with lower disagreement. Where a hard constraint on γ is required, λ should be tuned to ensure the constraint is satisfied.

With the query distribution ρ optimized, the inspection phase of an Δ-Audit concludes and the monitoring phase begins.

3.4 Monitoring With A Bounded Query Budget

So far, we have used the auditor's access privileges during the inspection phase of the Δ-Audit (Definition 1) to optimize the query distribution. During the monitoring phase, when the auditor has only black-box query access, it wants to detect a violation as fast as possible; or equivalently, with as few queries as possible. In this section, we quantify the probability of correctly rejecting the null, that is, the power of the monitoring hypothesis test of Section 2.

Let h_m be an update of h_0 and b the query budget. The b queries are i.i.d. from ρ, so the number of disagreements is $N_b^{\text{dis}} \sim \text{Bin}\big(b, \delta_\rho(h_m)\big)$. See Equation (3) for more details. The test of $H_0 \colon \delta_\rho(h_m) \le \gamma$ at level α rejects when $N_b^{\text{dis}} \ge k_b$. By solving the tail bound for the binomial, we have the critical value $k_b = \min\Big\{k \in \{0, \dots, b\} : \sum_{j=k}^b \binom{b}{j} \gamma^j (1-\gamma)^{b-j} \le \alpha\Big\}$, and $k_b = b + 1$ (never reject) when that set is empty. The rejection probability is the exact tail

$$\Pr(\text{reject}) = \Pr\big(N_b^{\text{dis}} \ge k_b\big) = \sum_{j=k_b}^{b} \binom{b}{j} \delta_\rho(h_m)^j \big(1 - \delta_\rho(h_m)\big)^{b-j}. \tag{9}$$

The unpaired test of $H_0 \colon U_{\text{aud}}(h_m, D) \ge \tau$, from labeled queries drawn from w^{aud}, is the same computation for the error rate: $\delta_\rho(h_m)$ is replaced by $1 - U_{\text{aud}}(h_m, D)$ and γ by $1 - \tau$.

Table 1: Summary of the scenarios and their auditor and provider utilities.

Scenario	Auditor utility U_{aud}	Provider utility U_{prov}	Application Details
Lending	$w_i^{\text{aud}} \propto 1/n_{g_i,y_i}$, $g = \text{SEX}$	$w_i^{\text{prov}} = c_{\text{fp}}\mathbb{1}\{y_i{=}0\} + c_{\text{fn}}\mathbb{1}\{y_i{=}1\}$, $(c_{\text{fp}}, c_{\text{fn}}) = (4, 1)$	The lender pays attention to the credit default risk. The auditor cares about balanced treatment between genders.
CelebA	$w_i^{\text{aud}} \propto 1/n_{g_i,y_i}$ within each source partition, $g = \texttt{Male}$	$w_i^{\text{prov}} = 1$	Gray-haired women are 0.58% of the training rows, so a model fitted for average accuracy is free to read "female" as evidence of "not gray".

4 Empirical Evaluation

A monitoring alarm triggers a new full white-box inspection by the auditor, rendering false alarms expensive for both the provider and the auditor. Thus, we first look at the detection guarantees provided by `Twister` asking **(RQ1)** Can `Twister` detect harmful model changes in fewer queries than the baseline performance monitoring test? We then study the case of a provider acting as an adaptive adversary, trying to conceal its model change to the auditor: **(RQ2)** Can `Twister` deter an adaptive adversary from cheating?

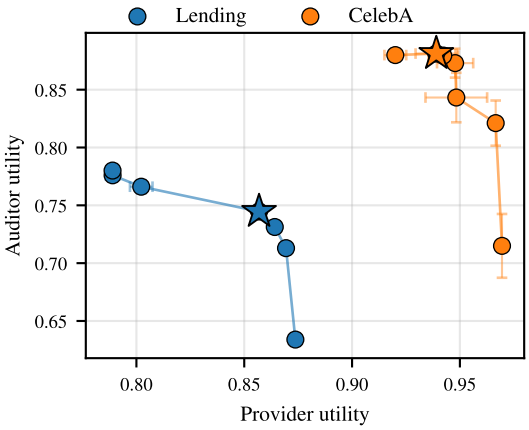


Figure 1: Provider vs. auditor utility of models trained with Equation (6); τ increases from the bottom-right (unconstrained) to the top-left point. $\star$: highest U_{aud} with U_{prov} within 95% of its maximum.

Whether honest or adversarial, we assume the model presented to the auditor is trained to maximize its utility, while respecting the auditor utility constraint (see 6). The constrained optimization problems of Equation (5) (audit manipulation) and Equation (6) (model training) are solved with a stochastic gradient descent-ascent algorithm Cotter et al. (2019) that alternatively optimizes the proxy-Lagrangian and the dual multiplier.

Datasets & models. We evaluate our method on two scenarios, one in the tabular and one in the image domain. We choose datasets commonly used in the fairness literature and designed the scenario to exhibit a realistic trade-off between the provider and auditor utilities. The tabular domain scenario is based on the `ACSIncome` task from the `Folktables` dataset Ding et al. (2021) that aggregates US census data and aims to predict household income from demographic indicators. We train linear models on `Folktables` data, randomly split and preprocessed with an off-the-shelf `Skrub` Varoquaux et al. (2026) table vectorizer, encoding categorical variables and scaling numerical variables. The image scenario is based on CelebA CITE[celeba], a collection of celebrities faces and corresponding attributes. The task is to predict whether a person has gray hair from a face shot. We use the provided train/dev/test split directly as the provider train/provider dev/auditor data and train a simple CNN adapted from the CIFAR10 speedrun project Jordan (2024). The scenarios and corresponding utilities are summarized in Table 1. The parties can differ in their sample weights, target labels, or both, as illustrated by the scenarios in Table 1.

Figure 1 shows the resulting trade-off between the two parties. In both scenarios, the auditor can obtain large gains at a small cost for the provider: compared to the unconstrained model ($\tau = 0$), the starred models improve U_{aud} by about 0.11 (Lending) and 0.17 (CelebA) while lowering U_{prov} by only 0.02 and 0.03. Past this point, the front flattens and each additional point of auditor utility becomes increasingly expensive for the provider. Conversely, once the audit has passed, the provider can recover its lost utility by silently reverting towards the unconstrained model, at a large cost for the auditor: this is the incentive that monitoring must deter.

Hyperparameters choice. For each scenario, we first tune γ by computing the average uniform disagreement rate between models trained with different seeds, then proceed as follows. For each

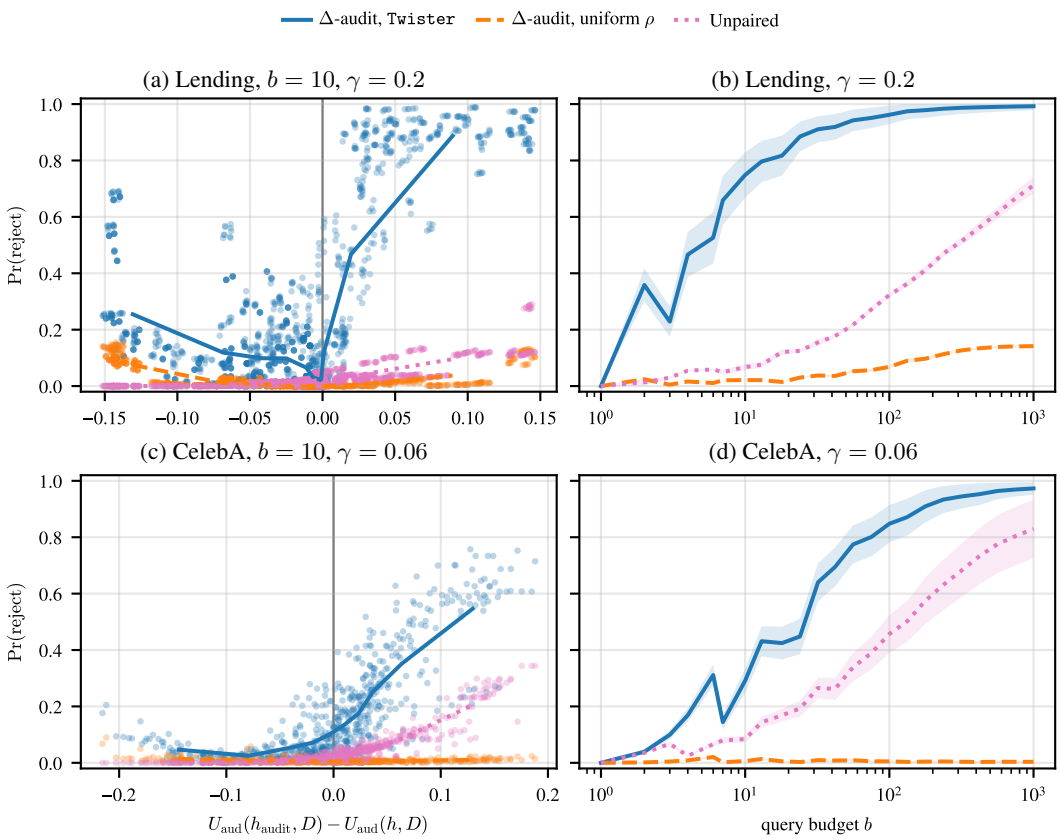


Figure 2: Rejection probability of each monitoring test. Left ($b = 10$): each dot is a model pair (h_0, h_m), placed by the auditor utility change $U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h_m, D)$ it causes; lines average the pairs within bins. Pairs right of the vertical line are harmful and should be flagged, pairs left of it are benign. Right: power (rejection probability on harmful pairs) against the query budget b; shaded areas show the standard deviation. Updates are done either from scratch or from the audited model.

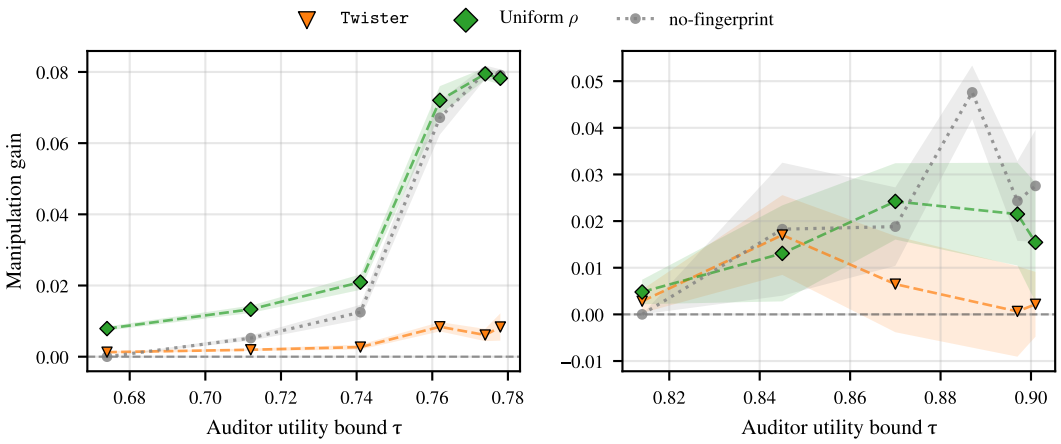
constraint value τ, we tune the learning hyperparameters on the provider dev set and select the feasible iterate that maximizes the provider loss. Then, for each selected model, we tune the fingerprint learning parameters to minimize the manipulation gain on the auditor's data. Finally, for each tuned scenario, τ, model h_0 and fingerprint ρ, we tune the manipulation's learning parameters and select the one that maximizes the platform utility on its dev set while satisfying the auditor constraint on the auditor data D. All the experiments are repeated 5 times varying the batch or split seeds and the results report the average and standard deviation.

Baselines. Along `Twister` with optimized weights ρ, we evaluate the performance of `Twister` with uniform weights ($\rho_i = \frac{1}{n}$ in Equation (1), coined `Uniform-Twister`). Moreover, since Δ-Audit is surrogate test for performance monitoring, we also evaluate an exact one-sided binomial test that distinguishes between $U_{\text{aud}}(h_m, D) \geq \tau$ and $U_{\text{aud}}(h_m, D) < \tau$ from labeled queries, coined the *unpaired* test (see Section 3.4). When evaluating manipulations, we also compare to an unmonitored provider, free to deploy any model (*no-fingerprint*).

Computational cost. Table 2 reports the time spent on each stage of a Δ-Audit. Building the `Twister` fingerprint takes 3 seconds on Lending and under 1.5 minutes on CelebA, i.e. 5 to 11 times a single training run of the provider model. This cost is paid once, at inspection time, and does not grow with the number of monitoring runs.

4.1 Detecting harmful changes in fewer queries

In a Δ-Audit, we test ρ-weighted disagreement, with ρ optimized to put weight on queries whose disagreement affects U_{aud} the most. Δ-Audit tests enjoys exact guarantees (see Section 3.4) but only as a change monitoring test. In this section, we evaluate if the ρ optimization phase `Twister`



(a) Lending, $\gamma = 0.2$

(b) CelebA, $\gamma = 0.06$

Figure 3: Manipulation gain as a function of the audit threshold τ, averaged over data and fingerprint seeds; shaded areas show the standard deviation. Random: `Uniform-Twister`; no-fingerprint: unmonitored provider. On both datasets, `Twister` attains the lowest manipulation gain for most thresholds, including all the strictest ones.

makes it a powerful and efficient monitoring test surrogate. For each scenario, we simulate model changes that affect the auditor utility by either retraining a new model or continuing training from h_0 after the audit, varying the auditor utility tolerance τ (see Equation (6)) and training seeds. We obtain a pool of model pairs (h_0, h_m) that are either *harmful* ($U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h_m, D) > 0$) or benign ($U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h_m, D) \leq 0$). For fixed updated models, we compare the probability of rejection under uniform and optimized query distributions as the budget b varies, using the same tolerance γ and significance level α in Section 2. This evaluates the true query cost of detection separately from the idealized manipulation-gain experiment in Equation (5).

`Twister` flags small harmful changes that the baselines miss. To evaluate the test sensitivity to harmful and benign changes, we bin the model pairs (h_0, h_m) by the auditor utility change $U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h_m, D)$. The left panels of Figure 2 plot the rejection probability of each test at budget $b = 10$ against this change. Each dot represents a model pair and the lines represent the average over all the pairs in one bin. Right of $x = 0$ are the harmful pairs ($\Lambda(h_0, h_m) > 0$) on which a monitoring test should raise an alarm. In the Lending scenario (Figure 2a), the unpaired test and `Uniform-Twister` reject every simulated pair with probability below 0.3, and below 0.15 on average. In contrast, the average power of `Twister` is already close to 0.5 for utility drops of 0.02 and reaches about 0.9 for drops of 0.09. In the CelebA scenario (Figure 2c), all tests are less powerful at $b = 10$, but the gap persists: `Twister` reaches an average power of about 0.55 for utility drops of 0.13, against at most 0.3 for the unpaired test and nearly 0 for `Uniform-Twister`. Left of $x = 0$ are the benign pairs on which a monitoring test should stay silent. The unpaired test directly tests the auditor utility requirement, so by construction it rejects benign pairs with probability below $\alpha = 0.05$. Δ-Audits instead test the disagreement with h_0, which a benign update can also exceed: their false alarm rate is only guaranteed to be below α when $\delta_\rho(h_m) \leq \gamma$. In the CelebA scenario, the average false alarm rate of `Twister` remains close to α, between 0.03 and 0.1. In the Lending scenario, however, it grows with the size of the improvement, from about 0.03 near $x = 0$ to about 0.26 for improvements of 0.13, with some pairs above 0.6.

`Twister` needs orders of magnitude fewer queries. Considering only the harmful pairs, the right panels of Figure 2 plot the test power against the query budget b. As b grows, the power of a Δ-Audit tends to the fraction of harmful pairs with $\delta_\rho(h_m) > \gamma$, and the power of the unpaired test to the fraction with $U_{\text{aud}}(h_m, D) < \tau$. In the Lending scenario (Figure 2b), `Twister` reaches a power of 0.8 in fewer than 20 queries, while after 1000 queries the unpaired test only reaches about 0.7 and `Uniform-Twister` about 0.15. In the CelebA scenario (Figure 2d), `Twister` needs about 80 queries to reach a power of 0.8, against about 1000 for the unpaired test. `Uniform-Twister`

never detects the harmful updates: they change less than a fraction γ of the uniformly weighted predictions and thus remain in its null hypothesis. The non-monotonic power at small budgets (e.g. from $b = 2$ to 3 in Lending and from $b = 6$ to 7 in CelebA) comes from the integer critical value k_b of the exact binomial test, which makes the test more conservative for some values of b. Overall, `Twister` is much more sensitive to small harmful changes and enjoys a better query efficiency than the unpaired test and `Uniform-Twister`, highlighting the benefits of our bi-level ρ optimization design. This comes at the price of more false alarms on large beneficial updates in the Lending scenario. This answers **(RQ1)**.

4.2 Deterring audit manipulations

An adaptive provider will always be able to optimize its model to satisfy the monitoring constraint (see Equation (5)). Thus, the second objective of `Twister` is to choose the monitoring constraint through ρ so that the adaptive provider's best response causes limited deterioration in auditor utility. Manipulation gain measures the provider's additional utility relative to honest compliance; reducing this gain toward zero reduces the incentive to deviate. For each scenario, we sweep the auditor threshold τ, evaluating manipulation gain relative to honest compliance as defined in Definition 2. The provider is assumed to manipulate its model to stay within `Twister`'s null hypothesis as in Equation (5). This is a restrictive assumption as, in theory, knowing the target false alarm rate α and query budget b, the provider could exploit the finite-query uncertainty of the test to "hide in the noise". However, considering the potential high cost of being caught cheating and the fact that this monitoring test is to be run regularly and indefinitely, we believe the provider has an interest in being conservative.

Figure 3 plots the manipulation gain as a function of the enforced auditor utility constraint τ. To evaluate the manipulation deterrence ability of Δ-Audits, we compare `Twister` with a situation where the provider is not monitored and free to update any model that maximizes its utility (no-fingerprint). Its manipulation gain is the provider utility given up to pass the inspection, i.e. the horizontal gap between the unconstrained model and h_0 on the Pareto front of Figure 1. In addition, to evaluate the importance of optimizing the sampling weights ρ_i, we compare `Twister` to `Uniform-Twister`.

In the Lending scenario (Figure 3a), the unmonitored provider recovers up to 0.08 of provider utility for the strictest thresholds. Under `Twister` monitoring, the manipulation gain stays below 0.01 for all values of τ: the provider has almost nothing to gain from deviating. `Uniform-Twister`, on the other hand, does not deter manipulation: its manipulation gain matches or exceeds the unmonitored one. In the CelebA scenario (Figure 3b), the results are more nuanced and vary more across seeds. For the two loosest thresholds ($\tau \leq 0.845$), all methods yield similar manipulation gains, below 0.02. For stricter thresholds ($\tau \geq 0.87$), where the unmonitored provider has the most to gain, `Twister` reduces the manipulation gain below 0.01, while `Uniform-Twister` leaves it between 0.015 and 0.025, close to the unmonitored provider. Overall, optimizing ρ is necessary to deter manipulation: a uniform disagreement test leaves the provider's incentive to deviate essentially intact, while `Twister` removes most of it in the Lending scenario and for the strictest thresholds in the CelebA scenario. This answers **(RQ2)**.

5 Conclusion & discussion

Monitoring ML-enabled system updates is crucial to ensure that ML audits remain a robust ML governance tool. This work introduced the Δ-Audit framework that pairs a model change detection test with the initial audit. Experiments on tabular and image classification tasks proved its efficacy at detecting model changes harmful to the users as measured by the auditor. Finally, even in the case of an adaptive adversarial provider, studying the utility a provider could gain by cheating the test reveals that Δ-Audit reduce the incentives for cheating. All in all, Δ-Audits provide a flexible framework to incorporate as much information (provider utility, model type and training recipe) and data available (training data, evaluation data) to help monitoring model performance and safety over time. Yet it remains an open question whether our framework could be applied to other properties of ML systems such as privacy or robustness. Moreover, the growing limitations of static evaluations due to population shifts, training data contamination or simply benchmark saturation open the question of whether Δ-Audits could help more dynamic notions of model evaluation.

AI USE STATEMENT

In this work, we used generative AI tools for none of the tasks with required disclosure. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, proposing or refining hypotheses, designing research methodology or experiments, implementing methods, cleaning or reformatting datasets, or interpreting results, and formulating or proving mathematical claims, writing proofs, translation, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for editing the paper to improve readability, identifying relevant literature, and creating or editing software code used to run our experiments. We have reviewed all AI-assisted work. All LLM-suggested text edits were reviewed and revised by the authors; every reference surfaced with LLM assistance was checked against the original publication before being cited; and all LLM-assisted experiment code was verified and tested for correctness by the authors before any results were produced. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Ulrich Aivodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. Fairwashing: the risk of rationalization. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 161–170. PMLR, May 2019. URL https://proceedings.mlr.press/v97/aivodji19a.html.

Önder Askin, Tim Kutta, and Holger Dette. Monitoring violations of differential privacy over time, September 2025. URL http://arxiv.org/abs/2509.20283.

Chhandak Bagchi, Filippo Menczer, Jennifer Lundquist, Monideepa Tarafdar, Anthony Paik, and Przemyslaw Grabowicz. Social media algorithms can curb misinformation, but do they? 2024. doi: 10.5281/zenodo.13787981. URL https://zenodo.org/records/13787981.

Kendrea Beers and Helen Toner. Enabling External Scrutiny of AI Systems with Privacy-Enhancing Technologies, February 2025. URL http://arxiv.org/abs/2502.05219.

Jade Garcia Bourrée, Erwan Le Merrer, Gilles Tredan, and Benoît Rottembourg. On the relevance of APIs facing fairwashed audits, May 2023. URL http://arxiv.org/abs/2305.13883.

California State Legislature. Senate Bill 53: Artificial intelligence models: large developers (Transparency in Frontier Artificial Intelligence Act), September 2025. URL https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=202520260SB53. Chapter 138, Statutes of 2025.

Timothée Chauvin, Erwan Le Merrer, François Taïani, and Gilles Trédan. Log probability tracking of LLM APIs, December 2025. URL http://arxiv.org/abs/2512.03816.

Timothée Chauvin, Clément Lalanne, Erwan Le Merrer, Jean-Michel Loubes, François Taïani, and Gilles Trédan. Token-efficient change detection in LLM APIs. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL http://arxiv.org/abs/2602.11083.

Ben Chugg, Santiago Cortes-Gomez, Bryan Wilder, and Aaditya Ramdas. Auditing fairness by betting. In *Thirty-seventh Conference on Neural Information Processing Systems*, November 2023. URL https://openreview.net/forum?id=EEVpt3dJQj.

Andrew Cotter, Heinrich Jiang, Maya Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *Journal of Machine Learning Research*, 20(172):1–59, 2019. ISSN 1533-7928. URL http://jmlr.org/papers/v20/18-616.html.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In *Advances in Neural Information Processing Systems*, volume 35, pp. 26698–26710, December 2022. URL

https://proceedings.neurips.cc/paper_files/paper/2022/hash/aa84ec1ac3f5fdcf77bce2c22705ab77-Abstract-Conference.html.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

European Parliament and Council of the European Union. 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 (Digital Markets Act), September 2022a. URL http://data.europa.eu/eli/reg/2022/1925/oj.

European Parliament and Council of the European Union. Regulation (EU) 2022/2065 of the European Parliament and of the Council of 19 October 2022 on a Single Market For Digital Services (Digital Services Act), October 2022b. URL http://data.europa.eu/eli/reg/2022/2065/oj.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), June 2024. URL http://data.europa.eu/eli/reg/2024/1689/oj.

Jean Feng, Alexej Gossmann, Gene A. Pennello, Nicholas Petrick, Berkman Sahiner, and Romain Pirracchio. Monitoring machine learning-based risk prediction algorithms in the presence of performativity. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, pp. 919–927. PMLR, April 2024. URL https://proceedings.mlr.press/v238/feng24b.html.

Olive Franzese, Ali Shahin Shamsabadi, and Hamed Haddadi. OATH: Efficient and flexible zero-knowledge proofs of end-to-end ML fairness, September 2024. URL http://arxiv.org/abs/2410.02777.

Drew Fudenberg and Jean Tirole. *Game theory*. MIT Press, 1991. ISBN 978-0-262-06141-4.

Kazuto Fukuchi, Satoshi Hara, and Takanori Maehara. Faking Fairness via Stealthily Biased Sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 412–419, April 2020. doi: 10.1609/aaai.v34i01.5377. URL https://ojs.aaai.org/index.php/AAAI/article/view/5377.

Jade Garcia Bourrée, Augustin Godinot, Sayan Biswas, Anne-Marie Kermarrec, Erwan Le Merrer, Gilles Tredan, Martijn De Vos, and Milos Vujasinovic. Robust ML auditing using prior knowledge. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 18794–18810. PMLR, October 2025. URL https://proceedings.mlr.press/v267/garcia-bourree25a.html.

Jade Garcia Bourrée, Erwan Le Merrer, Benoît Rottembourg, and Gilles Tredan. Leveraging imperfect sources to detect fairwashing in black-box auditing. In Michelangelo Ceci, Dragi Kocev, Anna Monreale, Celine Vens, Peter Flach, Giuseppe Manco, Myra Spiliopoulou, and Elio Masciari (eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track*, pp. 409–426. Springer Nature Switzerland, 2027. ISBN 978-3-032-37654-1. doi: 10.1007/978-3-032-37654-1_24.

Jun Geng, Bingwen Zhang, Lauren M. Huie, and Lifeng Lai. Online change detection of linear regression models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4910–4914, March 2016. doi: 10.1109/ICASSP.2016.7472611. URL https://ieeexplore.ieee.org/document/7472611.

Augustin Godinot, Erwan Le Merrer, Gilles Trédan, Camilla Penzo, and Franois Taïani. Under manipulations, are some AI models harder to audit? In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 644–664, April 2024a. doi: 10.1109/SaTML59370.2024.00038. URL https://ieeexplore.ieee.org/document/10516639.

Augustin Godinot, Gilles Tredan, Erwan Merrer, Camilla Penzo, and Francois Taiani. Queries, representation & detection: the next 100 model fingerprinting schemes. In *The 39th Annual AAAI Conference on Artificial Intelligence*, December 2024b. URL https://openreview.net/forum?id=rv0kUJses4.

Andrew M. Guess, Neil Malhotra, Jennifer Pan, Pablo Barberá, Hunt Allcott, Taylor Brown, Adriana Crespo-Tenorio, Drew Dimmery, Deen Freelon, Matthew Gentzkow, Sandra González-Bailón, Edward Kennedy, Young Mie Kim, David Lazer, Devra Moehler, Brendan Nyhan, Carlos Velasco Rivera, Jaime Settle, Daniel Robert Thomas, Emily Thorson, Rebekah Tromble, Arjun Wilkins, Magdalena Wojcieszak, Beixian Xiong, Chad Kiewiet de Jonge, Annie Franco, Winter Mason, Natalie Jomini Stroud, and Joshua A. Tucker. How do social media feed algorithms affect attitudes and behavior in an election campaign? 381(6656):398–404, 2023. doi: 10.1126/science.abp9364. URL https://www.science.org/doi/10.1126/science.abp9364.

Jeff Horwitz. Meta created 'playbook' to fend off pressure to crack down on scammers, documents show. *Reuters*, December 2025. URL https://www.reuters.com/investigations/meta-created-playbook-fend-off-pressure-crack-down-scammers-documents-show-2025-12-31/.

Keller Jordan. 94URL https://arxiv.org/abs/2404.00498.

Erwan Le Merrer and Gilles Trédan. Remote explainability faces the bouncer problem. *Nature Machine Intelligence*, 2(9):529–539, September 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-0216-z. URL https://www.nature.com/articles/s42256-020-0216-z.

Yuanchun Li, Ziqi Zhang, Bingyan Liu, Ziyue Yang, and Yunxin Liu. ModelDiff: testing-based DNN similarity comparison for model reuse detection. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2021, pp. 139–151. Association for Computing Machinery, July 2021. ISBN 978-1-4503-8459-9. doi: 10.1145/3460319.3464816. URL https://dl.acm.org/doi/10.1145/3460319.3464816.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. In *International Conference on Learning Representations*, October 2020. URL https://openreview.net/forum?id=hvdKKV2yt7T.

Xudong Pan, Yifan Yan, Mi Zhang, and Min Yang. MetaV: A Meta-Verifier Approach to Task-Agnostic Model Fingerprinting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 1327–1336. Association for Computing Machinery, August 2022. ISBN 978-1-4503-9385-0. doi: 10.1145/3534678.3539257. URL https://doi.org/10.1145/3534678.3539257.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.225. URL https://aclanthology.org/2022.emnlp-main.225/.

Inioluwa Deborah Raji, Andrew Smart, Rebecca N. White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pp. 33–44. Association for Computing Machinery, 2020. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372873. URL https://doi.org/10.1145/3351095.3372873.

Manoel Horta Ribeiro. Is Facebook "Standard" Algorithm Polarizing?, March 2024. URL https://doomscrollingbabel.manoel.xyz/p/is-facebook-standard-algorithm-polarizing.

Gurvan Richardeau, Gohar Dashyan, Erwan Le Merrer, and Gilles Trédan. FLIPS: Instance-fingerprinting for LLMs via pseudo-random sequences. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL http://arxiv.org/abs/2606.03330.

Leo Richter, Xuanli He, Pasquale Minervini, and Matt Kusner. An auditing test to detect behavioral shift in language models. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=h0jdAboh0o.

Glenn Shafer. Testing by betting: A strategy for statistical and scientific communication. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 184(2):407–431, April 2021. ISSN 0964-1998. doi: 10.1111/rssa.12647. URL https://doi.org/10.1111/rssa.12647.

Ali Shahin Shamsabadi, Sierra Calanda Wyllie, Nicholas Franzese, Natalie Dullerud, Sébastien Gambs, Nicolas Papernot, Xiao Wang, and Adrian Weller. Confidential-PROFITT: Confidential PROof of FaIr Training of Trees. In *The Eleventh International Conference on Learning Representations*, February 2023. URL https://openreview.net/forum?id=iIfDQVyuFD.

Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-Compare: Auditing Black-Box Models Using Transparent Model Distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pp. 303–310. Association for Computing Machinery, December 2018. ISBN 978-1-4503-6012-8. doi: 10.1145/3278721.3278725. URL https://doi.org/10.1145/3278721.3278725.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, February 2020. ISSN 0165-1684. doi: 10.1016/j.sigpro.2019.107299. URL https://www.sciencedirect.com/science/article/pii/S0165168419303494.

Gael Varoquaux, Lilian, Jérôme Dockès, Riccardo Cappuzzo, LeoGrin, Jovan Stojanovic, Vincent M, Patricio Cerda, Pierre Glaser, Guillaume Lemaitre, Thomas Schmitt, skrub bot, Théo Jolivet, Amanda Dsouza, Dimitri Papadopoulos Orfanos, Maxime Cuny, Marie Sacksick, alexis cvetkov, Lisa, Gensollen, Julien Jerphanion, gabrielapgomezji, Jérémie du Boisberranger, Léo DS, emassoulie, Hamed Homaeirad, Saba Siddique, Christine P. Chai, and Sylvain Combettes. skrub-data/skrub: 0.10.1, September 2026. URL https://doi.org/10.5281/zenodo.22231587.

Suppakit Waiwitlikhit, Ion Stoica, Yi Sun, Tatsunori Hashimoto, and Daniel Kang. Trustless audits without revealing data or models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 49808–49821. PMLR, July 2024. URL https://proceedings.mlr.press/v235/waiwitlikhit24a.html.

Chhavi Yadav, Amrita Roy Chowdhury, Dan Boneh, and Kamalika Chaudhuri. FairProof: Confidential and Certifiable Fairness for Neural Networks. In *Forty-first International Conference on Machine Learning*, June 2024. URL https://openreview.net/forum?id=EKye56rLuv.

Tianyu Zhang, Shen Dong, O. Deniz Kose, Yanning Shen, and Yupeng Zhang. A scalable system to prove machine learning fairness in zero-knowledge, May 2025. URL http://arxiv.org/abs/2505.07997.

A Extended Related Work

(Robust) audits. Audits are a critical tool for large online systems governance. By scrutinizing the data, models and objectives behind the systems we use every day, they hold platforms accountable for their systems and are increasingly used as regulation tools European Parliament and Council of the European Union (2024; 2022b;a); California State Legislature (2025). However, for a model provider, it is easy to fake compliance during an audit: they can provide indetectably biased data Fukuchi et al. (2020), manipulate model outputs Garcia Bourrée et al. (2025); Godinot et al. (2024a) and even exploit trustworthiness signals (e.g. explanations) to justify non compliance Aivodji et al. (2019); Le Merrer & Trédan (2020). This fragility has led to questioning the validity of external audits Ribeiro (2024) and has been exploited to avoid regulation Horwitz (2025). Existing attempts at more robust audits either derive audit validity guarantees from comparison with labeled data Garcia Bourrée et al. (2025), existing models Tan et al. (2018) and scraped data Bourrée et al. (2023), or employ cryptographic techniques to prove at inference time that the model used for predictions satisfies some properties Franzese et al. (2024); Shamsabadi et al. (2023); Yadav et al. (2024); Waiwitlikhit et al. (2024); Zhang et al. (2025).

Change detection/anytime hypothesis testing. Close to our goal of detecting harmful model changes post-audits are the fields of change-point detection and anytime hypothesis testing. Given a stream of samples, the goal of change point detection is to uncover changes of the data generating distribution. Common techniques include blah bla blaw. While there are some attempts at detecting parametric model changes Geng et al. (2016), most methods are focused on measuring changes in the data Truong et al. (2020). A more recent take that re-popularized the notion of e-value is anytime valid hypothesis testing Shafer (2021). Since then, this framework has been applied to different trustworthiness metrics such as fairness Chugg et al. (2023) or privacy Askin et al. (2025). However, at each new measurement step, these techniques require fresh samples from the data distribution which we do not need access to in our setup. Moreover, we have access to the full original model a $t = 0$, information that is not taken into account by these existing methods.

Model fingerprinting. Fingerprinting methods exploit various signals such as closeness to decision boundary Li et al. (2021), logits distribution Pan et al. (2022), presence of training data Maini et al. (2020) or even artifacts of the generation process Richardeau et al. (2026); Chauvin et al. (2025) to identify model provenance from black-box queries. Model fingerprints are engineered to be robust to model changes Godinot et al. (2024b), i.e. still identify provenance even if the re-used model was slightly modified. On the other hand, the ideal Δ-Audit would be agnostic to small model changes and only detect changes that affect the audit validity.

B Weighted disagreement and auditor utility

Proposition 1 (Sensitivity of utility to weighted disagreement)**.** *Suppose* $U_{aud}(h, D) = \sum_i w_i^{aud} \mathbb{1}[h(x_i) = y_i]$*, with* $w_i^{aud} \geq 0$ *and* $\sum_i w_i^{aud} = 1$*, and* $U_{aud}(h_0, D) \geq \tau$*. Assume* $\rho_i > 0$ *wherever* $w_i^{aud} > 0$*, and let* $\kappa(\rho) = \max_{i:w_i^{aud}>0} w_i^{aud}/\rho_i$*. For any* h_m *satisfying* $\delta_\rho(h_m) \leq \gamma$*,*

$$U_{aud}(h_m, D) \geq \tau - \kappa(\rho)\delta_\rho(h_m) \geq \tau - \kappa(\rho)\gamma.$$

Proof. Only points on which the predictions differ can contribute to a utility difference. Thus,

$$\begin{aligned} U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h_m, D) &= \sum_i w_i^{\text{aud}} \big(\mathbb{1}[h_0(x_i) = y_i] - \mathbb{1}[h_m(x_i) = y_i]\big) \\ &\leq \sum_{i:h_m(x_i)\neq h_0(x_i)} w_i^{\text{aud}} \\ &\leq \kappa(\rho) \sum_{i:h_m(x_i)\neq h_0(x_i)} \rho_i = \kappa(\rho)\delta_\rho(h_m). \end{aligned}$$

The result follows from the inspection requirement and the disagreement bound. □

This is a deterministic bound on the audit set. Non-rejection by a finite-query test does not establish its assumption $\delta_\rho(h_m) \leq \gamma$. The bound is conservative and does not use the provider's incentives or model class.

C Alternative design by subset selection

The main formulation optimizes a distribution over all audit points. An alternative is to select a subset of size k and use the uniform distribution on that subset; k is distinct from the monitoring query budget b. The following formulations express subset selection, its binary-indicator representation, and a continuous relaxation.

$$\begin{aligned} \min_{(x_1,\ldots,x_k)\subset D} \quad & (U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h^*, D))^+ \tfrac{1}{k} \textstyle\sum_{i=1}^k \mathbb{1}[h^*(x_i) = h_0(x_i)] \\ \text{s.t.} \quad & h^* \in \arg\max_h \quad U_{\text{prov}}(h, D_{\text{train}}) \\ & \qquad \text{s.t.} \quad \tfrac{1}{k} \textstyle\sum_{i=1}^k \mathbb{1}[h(x_i) \neq h_0(x_i)] \leq \gamma \end{aligned} \tag{10}$$

Table 2: Median wall time (seconds) of each stage of a Δ-Audit with `Twister`, over all τ and repetitions: honest constrained fit of h_0, fingerprint optimization, and provider manipulation.

Scenario	Fit	Fingerprint	Manipulation
Lending	0.26	2.9	0.08
CelebA	15	84	5.1

$$\begin{aligned} \min_{(\delta_1,\ldots,\delta_n)\in\{0,1\}^n} \quad & (U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h^*, D))^+ \tfrac{1}{k}\textstyle\sum_{i=1}^n \delta_i \mathbb{1}[h^*(x_i) = h_0(x_i)] \\ \text{s.t.} \quad & h^* \in \arg\max_h \; U_{\text{prov}}(h, D_{\text{train}}) \\ & \qquad \text{s.t.} \;\; \tfrac{1}{k}\textstyle\sum_{i=1}^n \delta_i \mathbb{1}[h(x_i) = h_0(x_i)] \geq 1-\gamma \\ & \textstyle\sum_{i=1}^n \delta_i = k \end{aligned} \tag{11}$$

The indicator formulation is relaxed by letting $\delta_i \in [0,1]$ and setting $\rho_i = \delta_i/k$:

$$\begin{aligned} \min_{\rho\in\Delta^n} \quad & (U_{\text{aud}}(h_0, D) - U_{\text{aud}}(h^*, D))^+ \textstyle\sum_i \rho_i \mathbb{1}[h^*(x_i) = h_0(x_i)] \\ \text{s.t.} \quad & h^* \in \arg\max_h \; U_{\text{prov}}(h, D_{\text{train}}) \\ & \qquad \text{s.t.} \;\; \textstyle\sum_i \rho_i \mathbb{1}[h(x_i) = h_0(x_i)] \geq 1-\gamma \\ & \rho_i \leq 1/k, \;\; \forall i \end{aligned} \tag{12}$$

The cap $\rho_i \leq 1/k$ follows from this subset-selection relaxation. It is not required by the unrestricted distribution design in Equation (7), and a relaxed solution need not be uniform on a subset.

D Computational cost

Table 2 summarizes the wall-clock time spend training the model by the provider, optimizing the fingerprint by the auditor and manipulating the fingerprint by adaptive providers