Robust ML Auditing using Prior Knowledge

Jade Garcia Bourrée [1 2 3] **Augustin Godinot** [1 2 3 4] **Martijn De Vos** [5] **Milos Vujasinovic** [5] **Sayan Biswas** [5] **Gilles Tredan** [6] **Erwan Le Merrer** [2] **Anne-Marie Kermarrec** [5]

Abstract

Among all the technical challenges to enforcing AI regulations, one crucial, yet under-explored problem is the risk of audit manipulation. These manipulations occur when a platform deliberately alters its answers to a regulator to pass an audit without modifying its answers to other users. In this paper, we introduce a novel approach to manipulation-proof auditing by taking into account the auditor's prior knowledge of the task solved by the platform. Through both practical and formal analysis of our framework, we argue that 1) Current audits are easily manipulated, 2) Regulators must not rely (only) on public priors (*e.g.*, public datasets), 3) Looking at the accuracy of the platform's answers is a good baseline to detect manipulations.

1. Introduction

Independent *fairness audits* by auditors serve as a critical tool for assessing the fairness of machine learning (ML) models and ensure that model providers remain accountable to the public (Birhane et al., 2024; Raji, 2024; Raji et al., 2022). As models are placed in production, auditors rely on black-box interactions, where queries are sent to the model, and the responses are analyzed to identify potential fairness violations (*e.g.*, see (Kim et al., 2019)). However, this reliance on black-box audits leaves the process vulnerable to *manipulations* by the platform, also known as *fairwashing*. Regulatory practices currently require auditors to notify platforms in advance of an audit. Platforms can thus strategically alter the model or its responses during the audit to create the appearance of fairness, effectively concealing underlying biases and unfair practices from the auditor while maintaining operational efficiency for its users. Consider, for example, a social media platform that employs an ML model to moderate content, automatically removing posts deemed harmful or misleading. During a fairness audit, the platform could deploy a more lenient moderation model that appears unbiased, only to revert to a stricter, potentially biased version once the audit concludes, effectively concealing unfair treatment of certain user groups.

This work presents a novel theoretical framework and a practical implementation for preventing manipulations by the platform. Our analysis starts from a simple observation: auditors can readily collect labeled data, reflecting the platform's service from independent sources – a common practice whose theoretical and empirical implications remain unexplored. For example, in the moderation example discussed earlier, the auditor could have some undeniable evidence at hand, to confront the model under scrutiny, *e.g.*, "A post with this content *must* pass the moderation filter, otherwise there is some bias on a protected feature of the user profile". Thus, by incorporating this dataset, the auditor can independently verify the platform's responses, cross-referencing them against known ground truth labels. By combining black-box interactions with prior knowledge from the labeled dataset, our method enables more reliable detection of fairness violations while reducing the reliance on assumptions about the platform's behavior. Specifically, we aim to answer the following research question: *Can the auditor's prior knowledge of the ground truth prevent fairwashing in fairness audits?*

2. Background: Auditing ML Models

This work studies fairness audits of ML decision-making systems under manipulation by the model-hosting platform. We first formalize the decision-making system and then introduce the dynamics of fairness auditing.

ML decision-making systems. From feature transforms to specific business rules, modern ML decision-making systems can be remarkably complex. We abstract all this complexity by modeling the entire system as a function

[*]Equal contribution [1]Université de Rennes, Rennes, France [2]Inria, Rennes, France [3]IRISA/CNRS, Rennes, France [4]PEReN, Paris, France [5]EPFL, Lausanne, Switzerland [6]LAAS, CNRS, Toulouse, France. Correspondence to: Augustin Godinot <augustin.godinot@inria.fr>, Jade Garcia Bourrée <jade.garcia-bourree@inria.fr>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

$h : \mathcal{X} \to \mathcal{Y}$ (*e.g.*, h can be a ML model). The set of possible queries $\mathcal{X}$ is called the *input space*, and the set of possible answers is called the *output space*. We consider binary classification problems, which is in line with related work in the domain of ML fairness analysis (Yan & Zhang, 2022; Godinot et al., 2024). Each query is associated with a protected attribute $a \in \mathcal{A}$, which the platform is legally required not to discriminate against. Examples of such attributes include gender, age, or race and are typically defined by law. The platform has access to the protected attribute a either as a feature of the input space $\mathcal{X}$ or by a proxy (e.g. looking at the name of the person to determine the gender). We define $\mathcal{D}$ as the data distribution on $\mathcal{X} \times \mathcal{A}$. For any subset $S \subset \mathcal{X} \times \mathcal{A}$ and protected feature value $a \in A$, we will write $S_a = \{x \mid (x, a') \in S, a' = a\}$ and $h(S) = \{h(x) \mid (x, a) \in S\}$. Throughout the paper, when it is clear from the context we will abuse the S notation: S will either be a subset of $\mathcal{X}$, $\mathcal{X} \times \mathcal{A}$ or $\mathcal{X} \times \mathcal{A} \times \mathcal{Y}$.

ML Auditing A ML audit is "any independent *assessment* of an identified *audit target* via an *evaluation* of articulated expectations with the implicit or explicit objective of *accountability*" (Birhane et al., 2024). A ML audit involves three entities. The *platform* is the entity hosting the ML decision-making system. The *users* are those using the service hosted by the platform. The *auditor* is the entity conducting the audit to verify whether the ML model is compliant for *all* users. The auditor could be a state regulator, a consulting firm, or even a group of users.

Fairness metric In this work, we consider ML audit targeting the *fairness* of the studied system. Specifically, the auditor chooses a fairness metric and sends queries to the platform to determine whether the platform abides by their fairness criterion. Among all the (un-)fairness metrics, we study the *Demographic Parity (DP)* (Calders et al., 2009), which is commonly used in the fairness evaluation literature thanks to its simplicity. DP is defined as follows:

$$\mu(h) = \begin{aligned} &\mathbb{P}_{(X,A)\sim\mathcal{D}}\left(h(X) = 1 | A = 1\right) \\ &- \mathbb{P}_{(X,A)\sim\mathcal{D}}\left(h(X) = 1 | A = 0\right) \end{aligned} \tag{1}$$

For a platform, DP is the easiest metric to manipulate (Yan & Zhang, 2022; Ajarra et al., 2024) as it only depends on the *outcome* of the ML model and not on its performance on the different protected groups. Thus, a platform can artificially adjust outputs, *e.g.*, providing more positive outcomes for an underrepresented group. To decide whether a platform passes the audit or not, the auditor builds an *audit set* $S \subset \mathcal{X} \times \mathcal{A}$ and evaluates the plug-in DP estimator: $\hat{\mu}(h, S) = \frac{1}{|S_1|} \sum_{x \in S_1} \mathbb{1}\{h(x) = 1\} - \frac{1}{|S_0|} \sum_{x \in S_0} \mathbb{1}\{h(x) = 1\}$. Based on $\mu(h)$, we also define the set of fair models $\mathcal{F} = \left\{h \in \mathcal{Y}^{\mathcal{X}} \;:\; \mu(h) = 0\right\}$.

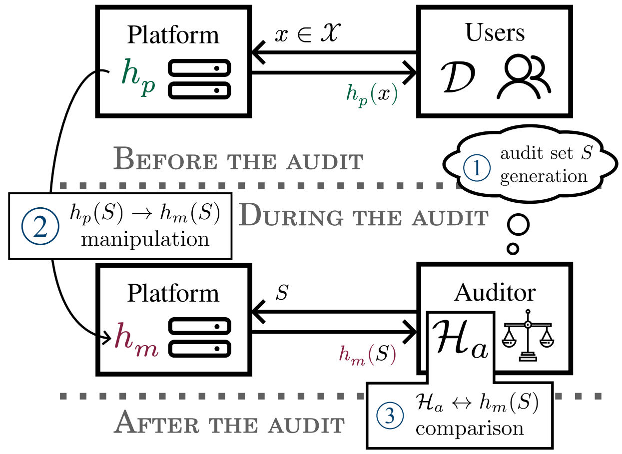

Figure 1. The auditing process as conducted by an auditor, which proceeds in three steps. The platform exposes a model h_p to the users. To appear fair to the auditor while not deteriorating the utility for its users, the platform manipulates its answers on the audit set S.

3. Enhancing Black-box Auditing with a Prior

Since a malicious platform can manipulate the DP metric with relative ease, the auditor has to find ways to prevent these manipulations (*e.g.*, using a different metric) or to detect them. In this section, we explore the latter. To detect manipulations, the auditor must use *prior* knowledge about what constitutes a "likely set of answers" on its audit dataset S. Then, using this prior, they would be able to estimate the likelihood that the set of answers $h_m(S)$ they received has been manipulated.

Definition 3.1 (Auditor prior)**.** The *auditor prior* is a set of models $\mathcal{H}_a \subset \mathcal{Y}^{\mathcal{X}}$ that the auditor can reasonably expect to observe given her knowledge of the decision task by the platform.

Examples of auditor prior For example, in (Tan et al., 2018), the authors study feature importance by training two models — one on a public dataset and another via distillation of the audited ML model — and comparing the resulting models. Building on the active learning literature, Yan & Zhang and Godinot et al. explored the case of an auditor knowing the hypothesis class of the platform, *i.e.*, $\mathcal{H}_a = \mathcal{H}$. Ajarra et al. proposed to use an assumption about the Boolean Fourier coefficients of $\mathcal{H}$ to construct $\mathcal{H}_a$. Finally, Garcia Bourrée et al. and Shamsabadi et al. used side-channel access (*e.g.*, an additional API or explanations) to the ML model to define $\mathcal{H}_a$ and derive guarantees on the measured fairness. In Section 4, we introduce a labeled dataset D_a that the auditor will leverage to define $\mathcal{H}_a$. Definition 3.1 captures all of the situations above and allows to formulate general results about the problem of robust auditing.

The auditing process The auditing process consists of three steps which we visualize in Figure 1. Here, h_p refers to the model that the platform exposes to its users (the top part of Figure 1) and h_m refers to the model exposed to the auditor (bottom part of Figure 1). First, the auditor builds an audit set $S \subset \mathcal{X}$ and sends the queries in S to the platform (step ①). The platform receives S *all at once* and computes the answers using its model h_p. To appear fair if it is not, the platform projects its labels $h_p(S)$ on the set $\mathcal{F}$ of fair models. This defines a manipulated model h_m and the answers $h_m(S)$ the platform will send to the auditor (step ②). The auditor receives $h_m(S)$ and exploits these samples to evaluate whether the platform is fair ($h_m \in \mathcal{F}$) and honest ($h_p = h_m$), (step ③). Since the auditor does not have direct access to h_p, they compare h_m to their prior $\mathcal{H}_a$ to decide whether the platform is honest or malicious. Thus, the auditor tests the two following properties of h_m:

$$\textbf{Is the platform fair}? \quad h_m \overset{?}{\in} \mathcal{F} \tag{2}$$

$$\textbf{Is the platform honest}? \quad h_m \overset{?}{\in} \mathcal{H}_a \tag{3}$$

For dataset priors (*i.e.*, when $\mathcal{H}_a$ is a ball, see Section 4), we draw $\mathcal{F}$ and $\mathcal{H}_a$ in Figure 2. Given a model h_m, the fairness audit is equivalent to checking if h_m belongs to the blue shaded area. In the example of Figure 2, the platform would be flagged as malicious as h_m belongs to $\mathcal{F}$ but not to $\mathcal{H}_a$.

Auditing axioms To avoid trivial audits, we add two modeling assumptions. The first axiom ensures that any honest platform with a model h_p will never appear as lying. The second axiom prevents trivial audits for which the auditor could directly conclude from his prior that the platform is unfair. Those assumptions are expressed as:

$$h_p \in \mathcal{H}_a \qquad \text{and} \qquad \mathcal{H}_a \cap \mathcal{F} \neq \emptyset \tag{4}$$

On public auditor priors A typical auditor proceeds in the following way. Upon examining a platform's model h_m, the auditor must first understand the task addressed by h_m and what constitutes a "good-performing model" on this task. In our moderation example, the auditor might try to look for public moderation datasets to test the performance of h_m with a few examples. It might also look for publicly-available moderation models to compare their resulting input/output pairs with those of h_m. Unfortunately, our first remark is that regardless of the prior the auditor might construct, if these models are public (or at least known by the platform), the platform will always be able to manipulate the audit:

Theorem 3.2. *Assume the platform knows $\mathcal{H}_a$, it can then always pick $h_m \in \{\mathcal{H}_a \cap \mathcal{F}\}$ to appear both fair and honest.*

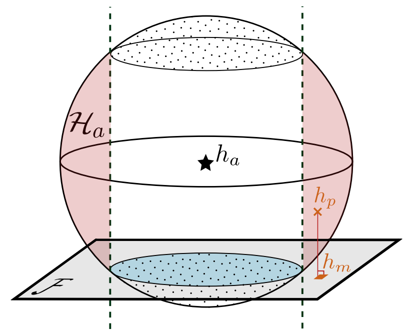

Figure 2. Representation of the auditor prior $\mathcal{H}_a$, the honest platform model h_p and a corresponding malicious model h_m on the fair $\mathcal{F}$ plane. The red area represents the area where platforms optimal manipulations are detected as dishonest: they fall outside of the blue region of $\mathcal{F}$.

4. Using Labeled Datasets for More Robust Audits Against Manipulations

As a simple instanciation of our robust auditing framework, we propose to study the use of a private (because of Theorem 3.2) dataset D_a, collected by the auditor to construct the auditor prior $\mathcal{H}_a$. This idea (coupled with an assumption on the hypothesis class) has been studied experimentally (Tan et al., 2018) but the more recent theoretical works on robust auditing diverged towards studying priors on the model itself rather than on the data (Shamsabadi et al., 2022; Yan & Zhang, 2022; Ajarra et al., 2024).

Definition 4.1 (Dataset prior)**.** Let $D_a = (X_a, A_a, Y_a) \in \mathcal{X}^n \times \mathcal{A}^n \times \mathcal{Y}^n$ be a labeled dataset the auditor has access to. The dataset prior $\mathcal{H}_a$ is defined as the set of models that have a reasonable risk on D_a.

$$\mathcal{H}_a = \left\{ h \in \mathcal{Y}^{\mathcal{X}} : L(h, D_a) < \tau \right\}. \tag{5}$$

Optimal manipulation Given the audit set S and its model h_p, the objective of a manipulative platform is to create a set of answers $h_m(S)$ that appear fair to the auditor but also do not raise suspicions. Ideally, the platform would like to know the auditor prior $\mathcal{H}_a$ (see Theorem 3.2), but in the general case it cannot because it is not public information. As a consequence, the platform cannot directly optimize its answers to be expectable *and* fair. However, it still has cards up its sleeve; it already trained a model h_p on a dataset D that is close to that of the auditor D_a.

Thus, instead of searching h_m in $\mathcal{H}_a \cap \mathcal{F}$, the platform can assume that its true model h_p is expectable – that is, $h_p \in \mathcal{H}_a$ – and try to find a fair model $h_m \in \mathcal{F}$ while flipping as few labels as possible from h_p. Therefore, the optimal manipulation is the projection of h_p on $\mathcal{F}$:

$$h_m^* = \mathrm{proj}_{\mathcal{F}}(h_p) = \underset{h \in \mathcal{F}}{\arg\min}\, d(h, h_p). \tag{6}$$

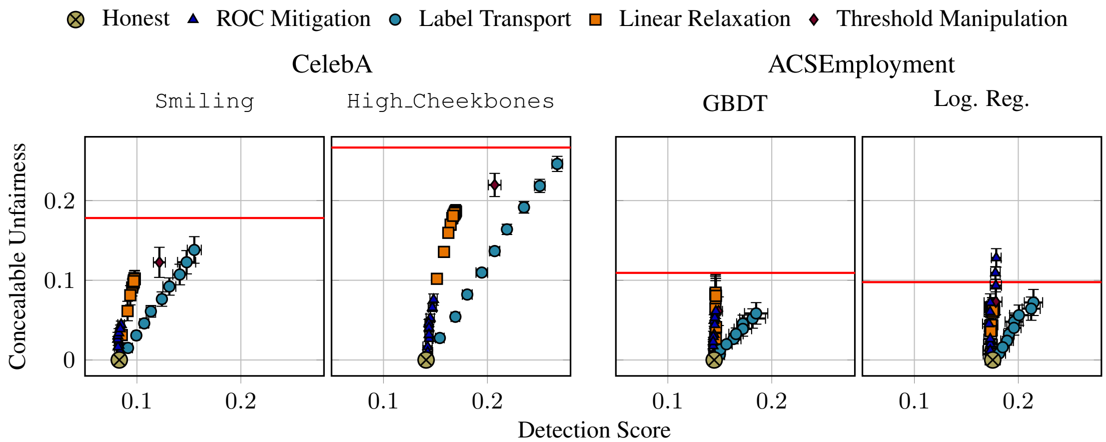

Figure 3. The concealable unfairness by the platform for different detection scores and manipulation strategies. We highlight this for two features of the CelebA dataset (left) and for two different ML models trained on the ACSEmployment dataset (right). The horizontal red line indicates the DP of the most unfair model without manipulation.

The distance d in Equation (6) is the value of risk L of h using the labels of h_m as the ground truth. This scenario captures the fairwashing approach in (Aïvodji et al., 2021) in the context of explanation manipulations. To gain intuition, we represent the audit case for $|S| = 3$ in Figure 2. By definition of the dataset prior, $\mathcal{H}_a$ is a ball of radius τ, centered on Y_a, the labels given in the audit dataset D_a.

5. Evaluating the robustness of audits

While evaluating audits is a broad topic (Costanza-Chock et al., 2022), we focus on their statistical performance. To empirically quantify the extent to which the platform can manipulate the unfairness of its ML model, we introduce the *concealable unfairness*: the maximum level of unfairness a platform can hope to hide before being detected as malicious. Since any practical fairness repair method can be used as a manipulation methods, we ask **(RQ1)** What is the best manipulation strategy implementation? In Appendix E, we also study the dynamics of the concealable unfairness when the audit budget $|S|$ increases.

Concealable unfairness The *concealable unfairness* $\Delta_\mu(h_p, h_m)$ is defined as the Demographic Parity gap between the manipulated and honest models. To decide whether the model observed during the audit is manipulated, the auditor has to decide whether $h_m \in \mathcal{H}_a$ or not. To do so, the auditor estimates $L(h, D_a)$ by computing the *detection score* Detect(h_m, S).

$$\Delta_\mu(h_p, h_m) = |\hat{\mu}(h_m, S) - \hat{\mu}(h_p, S)| \tag{7}$$

$$\text{Detect}(h_m, S) = \sum_{(x,y)\in S} \mathbb{1}\{h_m(x) \neq y\} \tag{8}$$

In Figure 3, we plot the value of the concealable unfairness $\Delta_\mu(h_p, h_m)$ against the detection score Detect(h_m, S) computed by the auditor, for different manipulation methods (ROC Mitigation (ROC), Optimal Label Transport (OT-L), Linear Relaxation (LinR) and Threshold Manipulation (ThreshOpt)). We show the results of LeNet models trained on two CelebA targets (first and second subplots), and Gradient Boosted Decision Tree (GBDT) and Logistic Regression (Log. Reg.) models trained on ACSEmployment (third and fourth subplots). The horizontal red lines indicates the DP of the most unfair model without manipulation.

Results First, we observe that for all the datasets, the platform can conceal significant amounts of unfairness: from 10 to 20 points differences between the two protected groups. Comparing the concealable unfairness values with the DP of the most unfair honest model (red horizontal line), we observe that the manipulation strategies almost all able to totally conceal the original model unfairness. Then, focusing on the x axis, the difference in Detect(h_m, S) between the different honest models highlights the impact the performance of the platform's model should have on the detection threshold τ. In fact, depending on the dataset and on the model, Detect$(honest, S)$ varies from ~ 0.1 to ~ 0.2. In Appendix E, we explore a solution to setup the threshold.

6. Conclusion

We explored both theoretically and experimentally the conditions in which an auditor can be manipulated or not when in the context of auditing with a prior. We provided an empirical method to tune the manipulation detection threshold to maximize the auditor probability to detect malicious platforms. As of futurework, it would be of practical interest to empirically derive means to find suitable radiuses, based on real uses cases an auditor can extract from her activity. In addition, we think that studying the gain for the detection power of the auditor, by combining the various priors proposed in the state of the art with ours would be an interesting outcome.

Impact Statement

This work theoretically and empirically analyzes fairness audits in ML decision-making systems to strategic manipulations by platforms seeking to evade regulatory scrutiny. By demonstrating how prior knowledge by auditors can enhance the robustness of black-box audits, we provide insights into mitigating audit manipulations. Our findings have implications for policymakers, auditors, and ML practitioners, further emphasizing the need for rigorous auditing frameworks that resist adversarial behavior.

The societal impact of this work is twofold. On the positive side, improving the robustness of fairness audits ensures greater accountability for platforms deploying ML models in high-stake domains such as finance or healthcare. By exploring the risk landscape of audit manipulations, our approach contributes to more trustworthy ML systems. However, we also highlight the limitations of current audit practices, suggesting that reliance on public priors can easily be exploited.

References

Aivodji, U., Arai, H., Fortineau, O., Gambs, S., Hara, S., and Tapp, A. Fairwashing: The risk of rationalization. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 161–170. PMLR, 2019. URL https://proceedings.mlr.press/v97/aivodji19a.html.

Aïvodji, U., Arai, H., Gambs, S., and Hara, S. Characterizing the risk of fairwashing. In *Advances in Neural Information Processing Systems*, November 2021.

Ajarra, A., Ghosh, B., and Basu, D. Active Fourier Auditor for Estimating Distributional Properties of ML Models, October 2024.

Albarghouthi, A., D'Antoni, L., Drews, S., and Nori, A. V. FairSquare: Probabilistic verification of program fairness. *Proc. ACM Program. Lang.*, 1(OOPSLA):80:1–80:30, October 2017. doi: 10.1145/3133904.

Anders, C., Pasliev, P., Dombrowski, A.-K., Müller, K.-R., and Kessel, P. Fairwashing explanations with off-manifold detergent. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 314–323. PMLR, 2020. URL https://proceedings.mlr.press/v119/anders20a.html.

Bénesse, C., Gamboa, F., Loubes, J.-M., and Boissin, T. Fairness seen as global sensitivity analysis. *Machine Learning*, 113(5):3205–3232, May 2024. ISSN 1573-0565. doi: 10.1007/s10994-022-06202-y.

Birhane, A., Steed, R., Ojewale, V., Vecchione, B., and Raji, I. D. AI auditing: The Broken Bus on the Road to AI Accountability. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 612–643, April 2024. doi: 10.1109/SaTML59370.2024.00037.

Borca-Tasciuc, G., Guo, X., Bak, S., and Skiena, S. Provable Fairness for Neural Network Models using Formal Verification, December 2022.

Buyl, M. and De Bie, T. Optimal transport of classifiers to fairness. *Advances in Neural Information Processing Systems*, 35:33728–33740, 2022.

Calders, T., Kamiran, F., and Pechenizkiy, M. Building Classifiers with Independency Constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pp. 13–18, December 2009. doi: 10.1109/ICDMW.2009.83.

Caton, S. and Haas, C. Fairness in Machine Learning: A Survey. *ACM Comput. Surv.*, 56(7):166:1–166:38, April 2024. ISSN 0360-0300. doi: 10.1145/3616865.

Cen, S. H. and Alur, R. From Transparency to Accountability and Back: A Discussion of Access and Evidence in AI Auditing, October 2024.

Cherian, J. J. and Candès, E. J. Statistical Inference for Fairness Auditing. *Journal of Machine Learning Research*, 25(149):1–49, 2024. ISSN 1533-7928.

Chugg, B., Cortes-Gomez, S., Wilder, B., and Ramdas, A. Auditing Fairness by Betting. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

Congress, U. Algorithmic accountability act of 2022. https://www.congress.gov/bill/117th-congress/house-bill/6580, 2022.

Costanza-Chock, S., Raji, I. D., and Buolamwini, J. 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 '22, pp. 1571–1583, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/

3531146.3533213. URL https://doi.org/10.1145/3531146.3533213.

Crémer, J., Dinielli, D., Heidhues, P., Kimmelman, G., Monti, G., Podszun, R., Schnitzer, M., Scott Morton, F., and De Streel, A. Enforcing the digital markets act: institutional choices, compliance, and antitrust. *Journal of Antitrust Enforcement*, 11(3):315–349, 2023.

de Vos, M., Dhasade, A., Garcia Bourrée, J., Kermarrec, A.-M., Le Merrer, E., Rottembourg, B., and Tredan, G. Fairness auditing with multi-agent collaboration. In *ECAI 2024*, pp. 1116–1123. IOS Press, 2024.

Deng, W. H., Guo, B., Devrio, A., Shen, H., Eslami, M., and Holstein, K. Understanding Practices, Challenges, and Opportunities for User-Engaged Algorithm Auditing in Industry Practice. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, pp. 1–18, New York, NY, USA, April 2023. Association for Computing Machinery. ISBN 978-1-4503-9421-5. doi: 10.1145/3544548.3581026.

DiCiccio, C., Vasudevan, S., Basu, K., Kenthapadi, K., and Agarwal, D. Evaluating Fairness Using Permutation Tests. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 1467–1477, New York, NY, USA, August 2020. Association for Computing Machinery. ISBN 978-1-4503-7998-4. doi: 10.1145/3394486.3403199.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. Retiring adult: New datasets for fair machine learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 6478–6490. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/32e54441e6382a7fbacbbbaf3c450059-Paper.pdf.

Fokkema, H., de Heide, R., and van Erven, T. Attribution-based Explanations that Provide Recourse Cannot be Robust. *Journal of Machine Learning Research*, 24(360):1–37, 2023. ISSN 1533-7928.

Fukuchi, K., Hara, S., and Maehara, T. Faking Fairness via Stealthily Biased Sampling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):412–419, April 2020. ISSN 2374-3468. doi: 10.1609/aaai.v34i01.5377.

Garcia Bourrée, J., Le Merrer, E., Tredan, G., and Rottembourg, B. On the relevance of APIs facing fairwashed audits, May 2023.

Ghosh, B., Basu, D., and Meel, K. S. Justicia: A Stochastic SAT Approach to Formally Verify Fairness. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):7554–7563, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i9.16925.

Ghosh, B., Basu, D., and Meel, K. S. Algorithmic Fairness Verification with Graphical Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):9539–9548, June 2022. ISSN 2374-3468. doi: 10.1609/aaai.v36i9.21187.

Godinot, A., Le Merrer, E., Trédan, G., Penzo, C., and Taïani, F. Under manipulations, are some AI models harder to audit? In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 644–664, April 2024. doi: 10.1109/SaTML59370.2024.00038.

Hardt, M., Price, E., Price, E., and Srebro, N. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Jiang, R., Pacchiano, A., Stepleton, T., Jiang, H., and Chiappa, S. Wasserstein Fair Classification. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pp. 862–872. PMLR, August 2020.

Kamiran, F., Karim, A., and Zhang, X. Decision theory for discrimination-aware classification. In *2012 IEEE 12th international conference on data mining*, pp. 924–929. IEEE, 2012.

Kim, M. P., Ghorbani, A., and Zou, J. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254, 2019.

Laberge, G., Aïvodji, U., Hara, S., Marchand, M., and Khomh, F. Fooling SHAP with Stealthily Biased Sampling. In *The Eleventh International Conference on Learning Representations*, September 2022.

Lam, M. S., Pandit, A., Kalicki, C. H., Gupta, R., Sahoo, P., and Metaxa, D. Sociotechnical Audits: Broadening the Algorithm Auditing Lens to Investigate Targeted Advertising. *Proc. ACM Hum.-Comput. Interact.*, 7(CSCW2):360:1–360:37, October 2023. doi: 10.1145/3610209.

Le Merrer, E. and Trédan, G. 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.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Li, S. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics & Statistics*, 4(1):66–70, 2010.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015. doi: 10.1109/ICCV.2015.425.

Lohaus, M., Perrot, M., and Von Luxburg, U. Too Relaxed to Be Fair. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6360–6369. PMLR, November 2020.

Maneriker, P., Burley, C., and Parthasarathy, S. Online Fairness Auditing through Iterative Refinement. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1665–1676, Long Beach CA USA, August 2023. ACM. ISBN 979-8-4007-0103-0. doi: 10.1145/3580305.3599454.

Ng, A. Can auditing eliminate bias from algorithms?, Feb 2021. URL https://themarkup.org/the-breakdown/2021/02/23/can-auditing-eliminate-bias-from-algorithms. Accessed: 2025-01-07.

NIST. Nist digital library of mathematical functions, 2013. URL http://dlmf.nist.gov/5.19#E4. Release 1.0.6 of 2013-05-06.

Raji, I. D. The Anatomy of AI Audits: Form, Process, and Consequences. In Bullock, J. B., Chen, Y.-C., Himmelreich, J., Hudson, V. M., Korinek, A., Young, M. M., and Zhang, B. (eds.), *The Oxford Handbook of AI Governance*, pp. 0. Oxford University Press, April 2024. doi: 10.1093/oxfordhb/9780197579329.013.28.

Raji, I. D., Xu, P., Honigsberg, C., and Ho, D. Outsider Oversight: Designing a Third Party Audit Ecosystem for AI Governance. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22, pp. 557–571, New York, NY, USA, July 2022. Association for Computing Machinery. ISBN 978-1-4503-9247-1. doi: 10.1145/3514094.3534181.

Shamsabadi, A. S., Yaghini, M., Dullerud, N., Wyllie, S., Aïvodji, U., Mkean, A. A. A., Gambs, S., and Papernot, N. Washing The Unwashable : On The (Im)possibility of Fairwashing Detection. In *Advances in Neural Information Processing Systems*, October 2022.

Shamsabadi, A. S., Wyllie, S. C., Franzese, N., Dullerud, N., Gambs, S., Papernot, N., Wang, X., and Weller, A. Confidential-PROFITT: Confidential PROof of FaIr Training of Trees. In *The Eleventh International Conference on Learning Representations*, February 2023.

Si, N., Murthy, K., Blanchet, J., and Nguyen, V. A. Testing Group Fairness via Optimal Transport Projections. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 9649–9659. PMLR, July 2021.

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES '20, pp. 180–186, New York, NY, USA, February 2020. Association for Computing Machinery. ISBN 978-1-4503-7110-0. doi: 10.1145/3375627.3375830.

Tan, S., Caruana, R., Hooker, G., and Lou, Y. 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, New York, NY, USA, December 2018. Association for Computing Machinery. ISBN 978-1-4503-6012-8. doi: 10.1145/3278721.3278725.

Taskesen, B., Blanchet, J., Kuhn, D., and Nguyen, V. A. A Statistical Test for Probabilistic Fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 648–665, New York, NY, USA, March 2021. Association for Computing Machinery. ISBN 978-1-4503-8309-7. doi: 10.1145/3442188.3445927.

Union, E. 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). https://eur-lex.europa.eu/eli/reg/2022/1925/oj/eng, 2022.

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

Yadav, C., Moshkovitz, M., and Chaudhuri, K. XAudit : A Theoretical Look at Auditing with Explanations, June 2023.

Yadav, C., Chowdhury, A. R., Boneh, D., and Chaudhuri, K. FairProof : Confidential and Certifiable Fairness for Neural Networks. In *Forty-First International Conference on Machine Learning*, June 2024.

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

Zhao, H. and Gordon, G. J. Inherent Tradeoffs in Learning Fair Representations. *Journal of Machine Learning Research*, 23(57):1–26, 2022. ISSN 1533-7928.

Zietlow, D., Lohaus, M., Balakrishnan, G., Kleindessner, M., Locatello, F., Schölkopf, B., and Russell, C. Leveling Down in Computer Vision: Pareto Inefficiencies in Fair Deep Classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10410–10421, 2022.

A. Characterization of the audit manipulability for dataset priors

Definition A.1. The probability P_{uf} that the auditor accurately detects a manipulative platform with optimal manipulation is $P_{uf} = P(h_m^* \notin \mathcal{H}_a | h_p \in \mathcal{H}_a)$.

In practice, the auditor can estimate P_{uf} by leveraging their access to the dataset D_a. For instance, they can train multiple models on D_a and approximate the set of expectable models $\mathcal{H}_a$ as the set of trained models that satisfy Equation 5. Furthermore, the auditor can estimate P_{uf} as the proportion of these models that remains expectable when projected onto $\mathcal{F}$. This provides the auditor with a convenient method to estimate the probability of detecting malicious platforms. Now that we defined the prior $\mathcal{H}_a$ as the dataset prior (Definition 4.1), it is possible to derive the exact value of P_{uf}. In Theorem A.2, we consider the $l2$-norm, but any metric might be considered. For simplicity, and because the auditor does not have any other additional knowledge of the platform's model, we assume that the distribution of models on $\mathcal{H}_a$ is uniform. Only the final expression of P_{uf} would change without those two assumptions. See Theorem F.3 in Appendix F for the general formula without those assumptions.

Theorem A.2. *If $\mathcal{H}_a$ is a ball centered in the ground-truth h_a with radius τ (i.e. $\mathcal{H}_a = B(h_a, \tau)$) in the space $(\mathcal{Y}^{\mathcal{X}}, \|.\|_2)$ then the probability that the auditor correctly detects a malicious platform trying to be fair is*

$$1 - \frac{1}{W_n}\left(\int_0^{\arccos(\delta/\tau)} \sin^n(\theta)d\theta - \frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2}\right).$$

with $\delta = d(h_a, \mathcal{F})$, the distance of h_a to $\mathcal{F}$ and W_n is the n-term of Wallis' integrals.

The proof of Theorem A.2 is deferred to Appendix F. It is also possible to prove that, if $h_a \in \mathcal{F}$ then $P_{uf} = 0$ (Corollary F.5). In other words, if the ground truth is fair and the prior $\mathcal{H}_a$ is a ball centered in the ground truth, then the auditor has zero chance to detect a manipulated model as non-expectable.

More generally, it is possible to choose τ (the threshold to characterize the expectable set) depending on δ such that P_{uf} is high. For instance, if $\tau = \delta$ then $P_{uf} = 1$ (Corollary F.4) so it is always possible to achieve audit with accuracy P_{uf} by correctly setting τ.

We now proceed to study the expression of P_{uf} in Theorem A.2. In particular, we derive a bound on P_{uf} to investigate the case when there is a strictly positive likelihood of the auditor successfully detecting a malicious platform.

Corollary A.3. *If n even, P_{uf} lies in between $\frac{1}{W_n}\frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2}$ and 1,* i.e., $\frac{1}{W_n}\frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2} \leq P_{uf} \leq 1$.

The proof has been postponed to Appendix F. Moreover, in order to develop an insight into the behavior of the lower bound of P_{uf}, further analysis of the expression derived in Corollary A.3 and of the case where P_{uf} has been carried out in Appendix F. In particular, Corollary A.3 combined with Corollary F.4 and Corollary F.5 help the auditor choose τ to maximize the probability of detecting malicious platforms.

In practice, τ is determined by the prior knowledge of the auditor. In the *PAC-learning* theory, the sample complexity is the number of examples that are required to guarantee a Probably Approximately Correct model. That is to say, it is possible to link the sample complexity with accuracy, a confidence parameter and some properties of the hypothesis class. For PAC-learnable hypothesis class, the more examples, the better the accuracy. If the auditor wants a high P_{uf}, they must have $\frac{\delta}{\tau} \to 1$ and so they must have the largest possible prior dataset.

Takeaway. The auditor can always calculate *a priori* the probability to correctly detect a malicious platform trying to be fair. This probability depends on multiple factors, among them the sample complexity: the more examples the auditor has in the prior dataset, the best the detection.

B. Online v.s. batch auditing

Note that we assume that the platform receives all audit queries at once and that it is possible to detect all the audit queries. In practice, the queries are usually issued online (that is, one-by-one) by the auditor, through web-scraping or through an API. Compared to online auditing, it is easier for the platform to manipulate an audit if it knows all the audit queries before having to answer. On the other hand, because the auditor has to send all their queries at once, they cannot use the answers of the platform to actively guide the generation of the audit questions (e.g. as in (Yan & Zhang, 2022; Godinot et al., 2024)). Ultimately, our setting is built as a worst-case analysis of the auditing game.

C. Experimental setup

C.1. Datasets and models

We conduct our experiments on tabular and vision modalities. The tabular dataset comes from the ACSEmployment task for the state of Minnesota in 2018, which is derived from US Census data and provided in folktables (Ding et al., 2021). The objective of this task is to predict whether an individual between the age of 16 and 90 is employed or not. As input features of the model h_p, we consider several attributes of the individual, including gender, race, and age. The fairness of the models is evaluated along the race attribute given in the dataset: one group consists of individuals identified as "white alone", while the other includes all remaining individuals.

For the vision modality, we study CelebA (Liu et al., 2015), which consists of images of celebrities along with several binary attributes associated with each image, such as whether the person in the photo is blond, smiling, or if the photo is blurry. As input to a vision model, we use the image to predict one of the associated attributes. The target attribute varies across experiments and will be specified accordingly. The Demographic Parity is evaluated along the gender attribute given in the dataset. For the ACSEmployment dataset, we train GBDT and Log. Reg. models, while for CelebA, we train a LeNet convolutional neural network (Lecun et al., 1998). GBDT and Log. Reg. are trained using the default parameters of their respective implementations in SCIKIT-LEARN. Meanwhile, LeNet is trained irrespective of the target attribute using the Adam optimizer with a learning rate of $\gamma = 0.001$, a batch size of 32, and for two epochs, which is sufficient for the model to converge on all features.

C.2. Implementing optimal audit manipulations

In practice, computing the optimal manipulation $h_m = \text{proj}_{\mathcal{F}}(h_p)$ amounts to solving:

$$h_m(S) \in \arg\min \quad L(h, \{(x, h_p(x)) : x \in S\}) \quad \text{s.t.} \quad \hat{\mu}(h, S) < \tau \tag{9}$$

We note that this problem is the same problem solved by in-processing and post-processing fairness repair methods (Caton & Haas, 2024). Thus, ironically, computing the optimal manipulation is equivalent to choosing the optimal fairness repair method. The only difference being on which set the fairness constraints and accuracy objectives are defined: the audit set S instead of the training dataset. Thus, since any practical fairness repair method can be repurposed for manipulation, we adapted four classical fairness repair methods: ROC (Kamiran et al., 2012), OT-L (Jiang et al., 2020), LinR (Lohaus et al., 2020) and ThreshOpt (Hardt et al., 2016).

D. Related Work

Companies are often motivated to bypass fairness audits to hide the unfairness of their ML models (Aivodji et al., 2019). Addressing fairness issues often requires compromising model performance for advantaged groups which can discourage companies from embracing fair training practices (Zietlow et al., 2022; Zhao & Gordon, 2022). At the same time, there is an increase in regulatory efforts to combat such manipulations and enforce fairness (Crémer et al., 2023). Frameworks such as the Algorithmic Accountability Act (AAA) (Congress, 2022) (US) and the Digital Markets Act (DMA) (Union, 2022) (EU) impose penalties on platforms failing to meet fairness standards.

Fairness auditing evaluates ML models to ensure fairness and accountability, often without access to proprietary model internals (Ng, 2021). This black-box auditing approach relies on querying the model and analyzing its outputs against predefined fairness metrics (Birhane et al., 2024; de Vos et al., 2024). Current attempts to enhance fairness audits with tangible guarantees draw inspiration from hypothesis testing (Si et al., 2021; Taskesen et al., 2021; DiCiccio et al., 2020; Cen & Alur, 2024; Cherian & Candès, 2024; Bénesse et al., 2024), online fairness auditing (Chugg et al., 2023; Maneriker et al., 2023), and formal methods for fairness certification (Albarghouthi et al., 2017; Ghosh et al., 2021; 2022; Borca-Tasciuc et al., 2022). Beyond statistical methods, the work of Yadav et al. explore the role of explanations in the auditing process (Yadav et al., 2023). Recent works also stress the importance of broadening the lens of algorithm auditing by incorporating user perspectives and sociotechnical factors (Lam et al., 2023; Deng et al., 2023). On another line of research, Confidential-

PROFITT and FairProof propose to integrate cryptographic techniques in cooperation with the platforms, to ensure the faithfulness of platform responses during audits (Yadav et al., 2024; Shamsabadi et al., 2023; Waiwitlikhit et al., 2024); this is, however, more intrusive and technically restrictive, and thus awaits for adoption.

Manipulating fairness audits is an active area of research. It has been shown that fairness can be faked through biased sampling when the decision maker is allowed to publish a labeled dataset as proof of model fairness (Fukuchi et al., 2020). Adversarial attacks on explanation methods, such as LIME and SHAP, can be employed to produce misleading interpretations of model behavior (Fokkema et al., 2023; Shamsabadi et al., 2022; Laberge et al., 2022; Aïvodji et al., 2021; Slack et al., 2020; Anders et al., 2020; Aivodji et al., 2019; Le Merrer & Trédan, 2020). Platforms can also modify the output of their models to create the appearance of fairness without addressing underlying biases (Yan & Zhang, 2022; Garcia Bourrée et al., 2023; Godinot et al., 2024). However, the challenge of designing audits that are robust to advanced manipulation strategies remains open.

This work studies leveraging prior knowledge, such as labeled datasets owned by auditors, to enhance the robustness of fairness audits. Distill-and-Compare is a distillation approach that assumes that the auditor has a prior about the ground truth and hypothesis class (Tan et al., 2018). Other audit approaches also assume prior knowledge of the hypothesis class (Yan & Zhang, 2022; Godinot et al., 2024). This work goes beyond existing approaches by deriving theoretical guarantees and bounds.

E. Dynamics of the concealable unfairness as the audit budget increases

In this section, we study the dynamics of the concealable unfairness when the audit budget $|S|$ increases.: **(RQ2)** Can the auditor always find an audit budget that prevents the platform from hiding any unfairness, *i.e.*, that always allows to flag the platform if malicious (Appendix E)?

The probability of detecting manipulations (via the the detection score) should intuitively increase as the auditor gains access to a larger number of data samples (*i.e.*, has a higher audit budget) since this allows for a more accurate comparison of h_p with the data prior $\mathcal{H}_a$. In this experiment, we explore how well this intuition holds in practice. For this purpose, we fix the hyperparameters for each manipulation method by selecting those that result in the highest concealable unfairness for a given base model. Then, for each base model–target attribute pair, we determine the maximum concealable unfairness that a platform can achieve while ensuring that its detection score (see eq. 8) remains below the detection threshold. As proposed in Section 4 the threshold for each model is set to $1 - x$, where x represents the maximum accuracy achieved when training a set of models on the corresponding target. This process is repeated for audit budgets ranging from 100 to $5,000$.

The results of this experiment are shown in Figure 4. The two plots on the left display the results for CelebA using the same base model but different target attributes, while the two plots on the right show results for ACSEmployment using the same target attribute but different base models. These results reveal two distinct cases. In the first case (CelebA `Smiling` in Figure 4), the concealable unfairness converges to zero as the audit budget increases. This is due to the low aleatoric uncertainty associated to the `Smiling` target. Since the task is easier, the accuracy range of models trained on `Smiling` is narrower, leading to a tighter detection threshold τ. In the second case (all the other facets of Figure 4), the concealable unfairness remains nonzero despite an increasing budget.

Furthermore, in many cases, even with a high audit budget, some increase of unfairness remains undetectable by the auditor. Consequently, the platform retains some capacity to conceal unfairness even at high audit budgets. This stresses the hardness of the auditor's task in some configurations, and lead to a negative answer to **(RQ2)**. In that light, we also observe that –in response to **(RQ1)**–, the Linear Relaxation and ROC Mitigation manipulation strategies are the most effective for a manipulative platform.

F. Proofs and additional theoretical results

As in (Buyl & De Bie, 2022), let $\mathcal{Z} \triangleq \mathcal{X} \times \mathcal{A} \times \{0, 1\}$ denote the sample space, from which the auditor draws samples $Z \triangleq (X, A, Y)$. The auditor sample the binary predictions $\hat{Y} \in \{0, 1\}$ from a probabilistic classifier $h : \mathcal{X} \to [0, 1]$ that assigns a score $h(X)$ to the belief that a sample with features X belongs to the positive class. It is assumed that $\mathcal{X} \subset \mathbb{R}^{d_\mathcal{X}}$ and $\mathcal{A} = \{[A_0, A_1]/A_0, A_1 \in \{0, 1\}\} = \{[1, 0], [0, 1]\}$(the one-hot encoding of the protected feature with two groups). We also assume that $\mathcal{H}_a$ is an open set of $\mathcal{Z}$.

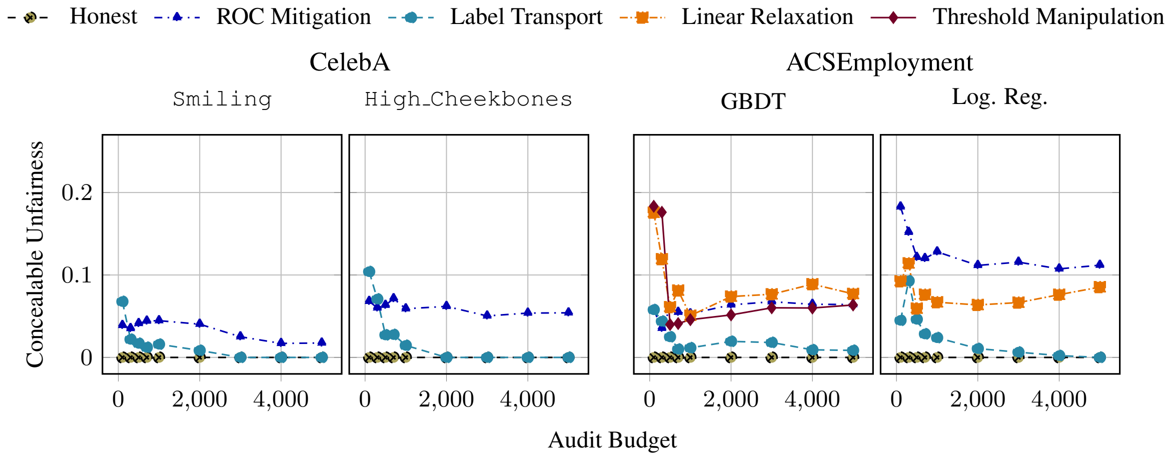

Figure 4. The concealable unfairness for different audit budgets (*i.e.*, data samples from the labeled dataset). We highlight this for two features of the CelebA dataset (left) and for two different ML models trained on the ACSEmployment dataset (right).

Table 1. Notations

$\mathcal{H}$	HYPOTHESIS CLASS
$\mathcal{F}$	SET OF FAIR MODELS
$\mathcal{H}_a$	SET OF EXPECTABLE MODELS
h_a	GROUND TRUTH
δ	DISTANCE BETWEEN THE GROUNDTRUTH AND THE SET OF EXPECTABLE MODEL
h_p	ORIGINAL MODEL OF THE PLATFORM
h_m	MANIPULATED MODEL OF THE PLATFORM
$\mathcal{X}$	INPUT SPACE
$\mathcal{D}$	DATA DISTRIBUTION
X	SAMPLE FROM INPUT SPACE
$\mathcal{Y}$	OUTPUT SPACE
Y	SAMPLE FROM OUTPUT SPACE
$\mathcal{A}$	PROTECTED FEATURE
$\mathcal{Z}$	SAMPLE SPACE
Z	SAMPLE
n	DIMENSION OF $\mathcal{Z}$

We denote $\mathcal{F}$ the set of all score functions $f : \mathcal{X} \to \{0, 1\}$ that satisfy (PDP):

$$\mathcal{F} \triangleq \{f : \mathcal{X} \to \{0,1\} : \mathbb{E}_Z\left[g(Z)f(X)\right] = 0_n\}$$

with $\forall k \in [2], g_k = \frac{A_k}{\mathbb{E}_Z[A_k]} - 1$, 0_n a vector of $n = d_{\mathcal{F}}$ zeros.

Assuming that the predictions $\hat{Y}|X$ are randomly sampled from a probabilistic classifier $h(X)$, then the traditional fairness notion of demographic parity (DP) is equivalent to PDP. But if $\hat{Y}$ is not sampled from $h(X)$ but instead decided by a threshold, DPD is a relaxation of the actual DP notion. That is to say, $\mathcal{F}$ is the set of all score functions that are fair regarding the demographic parity on $\mathcal{A}$.

As $\mathcal{F}$ is the kernel of the linear transformation $f : \mathbb{E}_Z\left[g(Z)f(X)\right]$, $\mathcal{F}$ is a hyperplane of $\mathcal{Z}$.

As $\mathcal{F}$ is a hyperplane of $\mathcal{Z}$, it is dense or closed in $\mathcal{Z}$.

F.1. Cases where $\mathcal{F}$ is dense in $\mathcal{Z}$.

Lemma F.1. *If $\mathcal{F}$ is dense in $\mathcal{Z}$, the auditor has a probability to detects it as manipulated equals to zero.*

Proof. If $\mathcal{F}$ is dense in $\mathcal{Z}$ then for every function $f \in \mathcal{Z}$, every open neighborhood of f intersects $\mathcal{F}$. In particular, it always exists a model $h_m \in \mathcal{F}$ that is in a neighborhood of h_p and in $\mathcal{H}_a$. In that case, h_m is fair and expectable, so the auditor has a probability to detects it as manipulated equals to zero. $\square$

This case is a pathological case where the platform can still appear fair and honest. For the next theoretical results, we are interested in the case where $\mathcal{F}$ is closed in $\mathcal{Z}$.

F.2. Cases where $\mathcal{F}$ is closed in $\mathcal{Z}$.

If $\mathcal{F}$ is a hyperplan closed in $\mathcal{Z}$, it has an empty interior (*i.e.* $\partial\mathcal{F} = \emptyset$) as its codimension is 1. In the following, we can thus use $\mathcal{F}$ instead of $\partial\mathcal{F}$, as both are equals.

Similarly, we can define the normal vector to $\mathcal{F}$ which is actually the vector that is used for all the projections we use in this paper. In Equation (6), we defined $h_m^* = \text{proj}_{\mathcal{F}}(h_p)$ (i.e. h_m^* is the orthographic projection of the expectable model h_p in the set of fair models $\mathcal{F}$).

Having an hyperplan lead to the natural definition of (hyper)cylinder, that we use in the following theorem.

Definition F.2. A right cylinder $C(H, B)$ is the set of all points whose orthographic projection on a hyperplane H lies in a set B with B a subset of the boundary of H. B is called the base of the cylinder.

Theorem F.3. *The probability P_{uf} that the auditor correctly detects a malicious platform trying to be fair is $P(\mathcal{H}_a \backslash C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) | \mathcal{H}_a)$.*

Proof. The auditor correctly detects a malicious platform trying to be fair if and only if the manipulated model is fair but not expectable. The manipulated model is fair but not expectable if and only if the orthographic projection h_m^* of h_p in $\mathcal{F}$ is not in $\mathcal{H}_a \cap \partial\mathcal{F}$. Thus, the manipulated model is fair but not expectable if and only if $h_p \notin C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F})$ (following Definition F.2). As by assumption $h_p \in \mathcal{H}_a$ (Equation (4)), it means that $h_p \in \mathcal{H}_a \backslash C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F})$. The auditor correctly detects a malicious platform trying to be fair with probability $P(\mathcal{H}_a \backslash C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) | \mathcal{H}_a)$. $\square$

Theorem A.2 is a special case of Theorem F.3 with additional assumption. We now prove the main Theorem Theorem A.2.

Theorem A.2. *If $\mathcal{H}_a$ is a ball centered in the ground-truth h_a with radius τ (i.e. $\mathcal{H}_a = B(h_a, \tau)$) in the space $(\mathcal{Y}^{\mathcal{X}}, \|.\|_2)$ then the probability that the auditor correctly detects a malicious platform trying to be fair is*

$$1 - \frac{1}{W_n}\left(\int_0^{\arccos(\delta/\tau)} \sin^n(\theta) d\theta - \frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2}\right).$$

with $\delta = d(h_a, \mathcal{F})$, the distance of h_a to $\mathcal{F}$ and W_n is the n-term of Wallis' integrals.

Proof. As established in Theorem F.3, $P_{uf} = P(\mathcal{H}_a \backslash C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) | \mathcal{H}_a)$.

The probability $P(\mathcal{H}_a \backslash C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F}) | \mathcal{H}_a)$ is the probability to be in the ball $\mathcal{H}_a$ without the probability to be in the intersection between the ball $\mathcal{H}_a$ and the cylinder $C(\mathcal{F}, \mathcal{H}_a \cap \partial\mathcal{F})$. In the following, we denote $V_n^{\#}(\tau, \delta)$ this quantity.

As $\mathcal{H}_a$ is an ball, its volume is:

$$V_n^{ball}(\tau) = \frac{\pi^{n/2}\tau^n}{\Gamma(\frac{n+2}{2})}$$

with $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$ (NIST, 2013).

The volume of the intersection between the cylinder and the ball is the sum of the three following volumes:

- the solid cylinder with height between $-\delta$ and δ
- the spherical cap of $\mathcal{H}_a$ that is *above* the previous cylinder (i.e. the part of $\mathcal{H}_a$ with height between δ and τ)
- the spherical cap of $\mathcal{H}_a$ that is *bellow* the previous cylinder (i.e. the part of $\mathcal{H}_a$ with height between $-\delta$ and $-\tau$)

According to (Li, 2010), the volume of each spherical cap is

$$V_n^{cap}(\tau,\delta) = \frac{\pi^{(n-1)/2}\tau^n}{\Gamma(\frac{n+1}{2})}\int_0^{\arccos(\delta/\tau)} \sin^n(\theta)d\theta$$

And the volume of the cylinder of height 2δ is

$$V_n^{cylinder}(\tau,\delta) = 2\delta V_{n-1}^{ball}(\sqrt{\tau^2-\delta^2})$$

Thus,

$$\begin{aligned} V_n^{\#}(\tau,\delta) &= V_n^{ball}(\tau) - 2V_n^{cap}(\tau,\delta) - V_n^{cylinder}(\tau,\delta) \\ &= \frac{\pi^{n/2}\tau^n}{\Gamma(\frac{n+2}{2})} - 2\frac{\pi^{(n-1)/2}\tau^n}{\Gamma(\frac{n+1}{2})}\int_0^{\arccos(\delta/\tau)} \sin^n(\theta)d\theta - 2\delta\frac{\pi^{(n-1)/2}(\sqrt{\tau^2-\delta^2})^{n-1}}{\Gamma(\frac{n+1}{2})} \end{aligned}$$

According to Theorem F.3, the probability that the auditor correctly detects a malicious platform trying to be fair is $P(\mathcal{H}_a \backslash C(\mathcal{F},\mathcal{H}_a\cap\partial\mathcal{F})|\mathcal{H}_a)$. That is to say, it is the ratio of $V_n^{\#}(\tau,\delta)$ over $V_n^{ball}(\tau)$:

$$\begin{aligned} P_{uf} &= P(\mathcal{H}_a \backslash C(\mathcal{F},\mathcal{H}_a\cap\partial\mathcal{F})|\mathcal{H}_a) \\ &= \frac{V_n^{\#}(\tau,\delta)}{V_n^{ball}(\tau)} \\ &= 1 - 2\frac{\Gamma(\frac{n+2}{2})}{\Gamma(\frac{n+1}{2})}\frac{\pi^{(n-1)/2}}{\pi^{n/2}}\int_0^{\arccos(\delta/\tau)} \sin^n(\theta)d\theta - 2\delta\frac{(\tau^2-\delta^2)^{(n-1)/2}}{\tau^n}\frac{\Gamma(\frac{n+2}{2})}{\Gamma(\frac{n+1}{2})}\frac{\pi^{(n-1)/2}}{\pi^{n/2}} \\ &= 1 - \frac{2}{\sqrt{\pi}}\frac{\Gamma(\frac{n+2}{2})}{\Gamma(\frac{n+1}{2})}\int_0^{\arccos(\delta/\tau)} \sin^n(\theta)d\theta - \frac{2\delta}{\sqrt{\pi}}\frac{(\tau^2-\delta^2)^{(n-1)/2}}{\tau^n}\frac{\Gamma(\frac{n+2}{2})}{\Gamma(\frac{n+1}{2})} \\ &= 1 - \frac{2}{\sqrt{\pi}}\frac{\Gamma(\frac{n+2}{2})}{\Gamma(\frac{n+1}{2})}\left(\int_0^{\arccos(\delta/\tau)} \sin^n(\theta)d\theta - \delta\frac{(\tau^2-\delta^2)^{(n-1)/2}}{\tau^n}\right) \end{aligned}$$

The function Γ can be written with Wallis' integrals as: $W_n = \frac{\sqrt{\pi}}{2}\frac{\Gamma(\frac{n+1}{2})}{\Gamma(\frac{n+2}{2})}$ with $\forall n, W_n = \int_0^{\pi/2}\sin^n(\theta)d\theta$.

In the other hand,

$$\begin{aligned} \delta\frac{(\tau^2-\delta^2)^{(n-1)/2}}{\tau^n} &= \frac{\delta}{\tau}\frac{(\tau^2-\delta^2)^{(n-1)/2}}{\tau^{n-1}} \\ &= \frac{\delta}{\tau}\left(\frac{\tau^2-\delta^2}{\tau^2}\right)^{(n-1)/2} \\ &= \frac{\delta}{\tau}\left(1-\frac{\delta^2}{\tau^2}\right)^{(n-1)/2} \end{aligned}$$

Thus, $P_{uf} = 1 - \frac{1}{W_n}\left(\int_0^{\arccos(\delta/\tau)}\sin^n(\theta)d\theta - \frac{\delta}{\tau}\left(1-\frac{\delta^2}{\tau^2}\right)^{(n-1)/2}\right)$. □

Before dealing with this complete expression, we propose some particular cases that are easily interpretable.

Corollary F.4. *If $\mathcal{H}_a$ is a ball centered in the ground-truth h_a that is tangent to $\mathcal{F}$, then the auditor has a probability one to correctly detect a malicious platform trying to be fair.*

$$\mathcal{H}_a = B(h_a, \tau) \wedge \tau = \delta \implies P_{uf} = 1.$$

with $\delta = d(h_a, \mathcal{F})$*, the distance of* h_a *to* $\mathcal{F}$*.*

Proof. If $\mathcal{H}_a$ is tangent to $\mathcal{F}$ then $\delta = \tau$. Thus, $\arccos(\delta/\tau) = \arccos(1) = 0$ and $\int_0^{\arccos(\delta/\tau)} \sin^n(\theta) d\theta = 0$.

Thanks to the formula of Theorem A.2 with $\delta/\tau = 1$, $P_{uf} = 1 - \frac{1}{W_n}(0 - 0) = 1$.

If $\mathcal{H}_a$ is tangent to $\mathcal{F}$, $P_{uf} = 1$. □

This corollary means that by reducing the threshold τ to the minimal value (δ), the auditor is sure to detect any manipulation of the platform.

Corollary F.5. *If* $\mathcal{H}_a$ *is a ball centered in the ground-truth* h_a *that is fair, then the auditor has a probability zero to correctly detect a malicious platform trying to be fair.*

$$\mathcal{H}_a = B(h_a, \tau) \wedge h_a \in \partial\mathcal{F} \implies P_{uf} = 0.$$

Proof. If $h_a \in \partial\mathcal{F}$ then $\delta = 0$ and $\arccos(\delta/\tau) = \arccos(0) = \pi/2$ in the formula of Theorem A.2. Thus, $P_{uf} = 1 - \frac{1}{W_n}(W_n - 0) = 0$.

□

This last case is the case where the h_a of the auditor is fair. Intuitively, if h_a is fair, half of the model that the platform can construct are naturally fair and the other half are naturally unfair. Thus, it is very easy to change from an unfair model to a fair model without changing too much the honest model. Thus, detecting such manipulation is very hard for the auditor.

Now, we study the general expression of P_{uf} in Theorem A.2. In particular, we study a lower bound of P_{uf} to study when the probability is strictly positive.

Corollary A.3. *If* n *even,* P_{uf} *lies in between* $\frac{1}{W_n}\frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2}$ *and* 1, i.e., $\frac{1}{W_n}\frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2} \leq P_{uf} \leq 1$.

Proof. $P_{uf} = 1 - \frac{1}{W_n}\left(\int_0^{\arccos(\delta/\tau)} \sin^n(\theta) d\theta - \frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2}\right)$

$W_n = \int_0^{\arccos(\delta/\tau)} \sin^n(\theta) d\theta + \int_{\arccos(\delta/\tau)}^{\pi/2} \sin^n(\theta) d\theta$

So, $\frac{1}{W_n}\int_0^{\arccos(\delta/\tau)} \sin^n(\theta) d\theta \leq 1$ (with n even).

And $P_{uf} \geq \frac{1}{W_n}\frac{\delta}{\tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2}$ □

Lemma F.6. *The lower bound according to* δ/τ *has two extremums that are for* $\delta/\tau = 1$ *or* $\delta/\tau = \gamma$ *with* $\gamma = \frac{\sqrt{n+3} - \sqrt{n-1}}{2}$.

Remark. Note that γ only depend on the dimension n and leads to 0 when n leads to infinity.

Proof. We define f_n (the lower bound) s.t.

$$f_n(\delta, \tau) = \frac{\delta}{W_n \tau}\left(1 - \frac{\delta^2}{\tau^2}\right)^{(n-1)/2}$$

Change of variable $x = \frac{\delta}{\tau}$, $f(x) = \frac{x}{Wn}(1 - x^2)^{(n-1)/2}$.

We are interested in cases where $\tau > \delta$, *i.e.* $0 < x < 1$.

Moreover, f has an extremum iff $f' = 0$ somewhere in $[0, 1]$.

$$\begin{aligned}\forall x \in [0,1], W_n f'(x) &= (1-x^2)^{(n-1)/2} - (n-1)x^2(1-x^2)^{(n-3)/2} \\ &= (1-x^2)^{(n-3)/2}(x^2 + \sqrt{n-1}x - 1)(x^2 - \sqrt{n-1}x - 1)\end{aligned}$$

i.e. $f'(x) = 0$ for the following elements:

- $x = -1 < 0$
- $x = 1$
- $\frac{-\sqrt{n-1}-\sqrt{n+3}}{2} < 0$
- $\frac{-\sqrt{n-1}+\sqrt{n+3}}{2} \in [0, 1]$
- $\frac{\sqrt{n-1}-\sqrt{n+3}}{2} < 0$
- $\frac{\sqrt{n-1}+\sqrt{n+3}}{2} > 1$ (if $n \geq 2$)

So f has two local extremums in $[0, 1]$, one for 1 and one for $\gamma = \frac{\sqrt{n+3}-\sqrt{n-1}}{2}$. □