Robust ML Auditing using Prior Knowledge

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

[1]LAAS, CNRS, Toulouse, France [2]Université de Rennes, Rennes, France [3]Inria, Rennes, France
[4]IRISA/CNRS, Rennes, France [5]PEReN, Paris, France [6]EPFL, Lausanne, Switzerland

What regulators ask for...

Audits require balancing between:

- **Security**: no audit gaming.
- **Data access**: the auditor can have access to training data and filters.
- **Model access**: the auditor can interact with the models via queries or have access to the code/weights.
- **Privacy**: no users' data leak.
- **IP protection**: no industrial secret should be leaked by the audit.

... but auditors are easily identifiable by platforms.

- Access via research APIs — auditor identified
- Anti-scraping IP whitelist — auditor identified
- Audit query patterns — auditor identified

$\Rightarrow$ avenue for audit manipulations!

ML audit you said ?

- **Input space** $\mathcal{X}$. *Example: The space of all possible* 1000×1000 *images.*
- **Model** $h_p \in \arg\min_{h \in \mathcal{H}} L(h, \mathcal{D})$. *Example: a good old GBDT.*
- **Fairness metric** $\mu : \mathcal{H} \to [-1, 1]$. *Example: demographic parity.*
- **Set of fair models** $\mathcal{F} = \{h \in \mathcal{Y}^{\mathcal{X}} : \mu(h) = 0\}$.

Example: make sure that in average, men are not advantaged compared to women by a resume screening algorithm.

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.

Axioms The prior is *reasonable* and the audit is *justified*.

$$h_p \in \mathcal{H}_a \qquad \mathcal{H}_a \cap \mathcal{F} \neq \emptyset$$

Theorem 3.2 (Public priors). If the auditor's prior $\mathcal{H}_a$ is perfectly known by the platform, a manipulative platform always appear fair and honest.

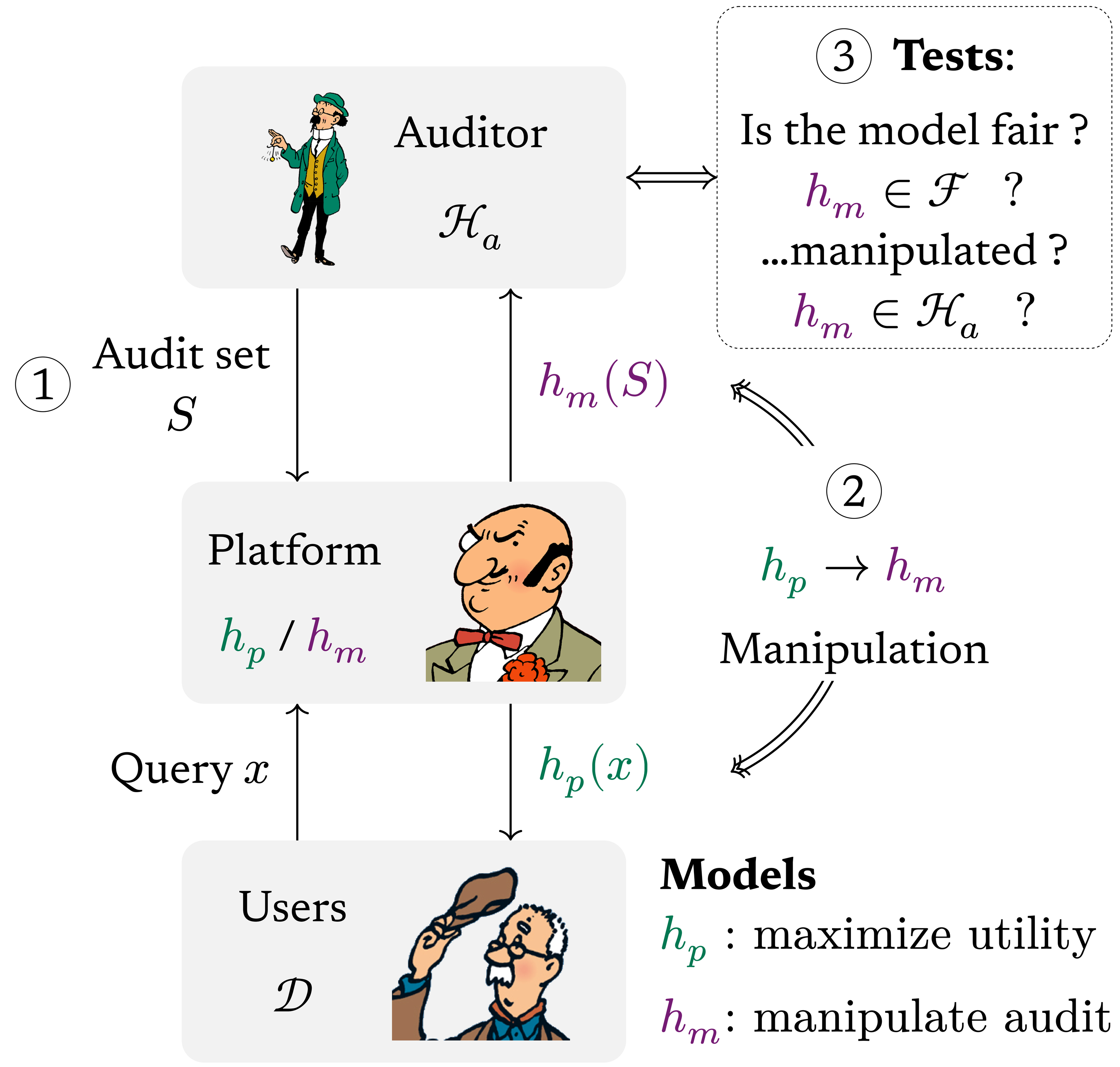
Definition (Optimal manipulation). Let h_p be the model that optimizes the platform's utility. Without knwoledge of the auditor's prior, the optimal platform manipulation is

$$h_m = \mathrm{proj}_{\mathcal{F}}(h_p) = \arg\min_{h \in \mathcal{F}} d(h, h_p)$$

ML audits are easily detectable.
How to prevent fairwashing?

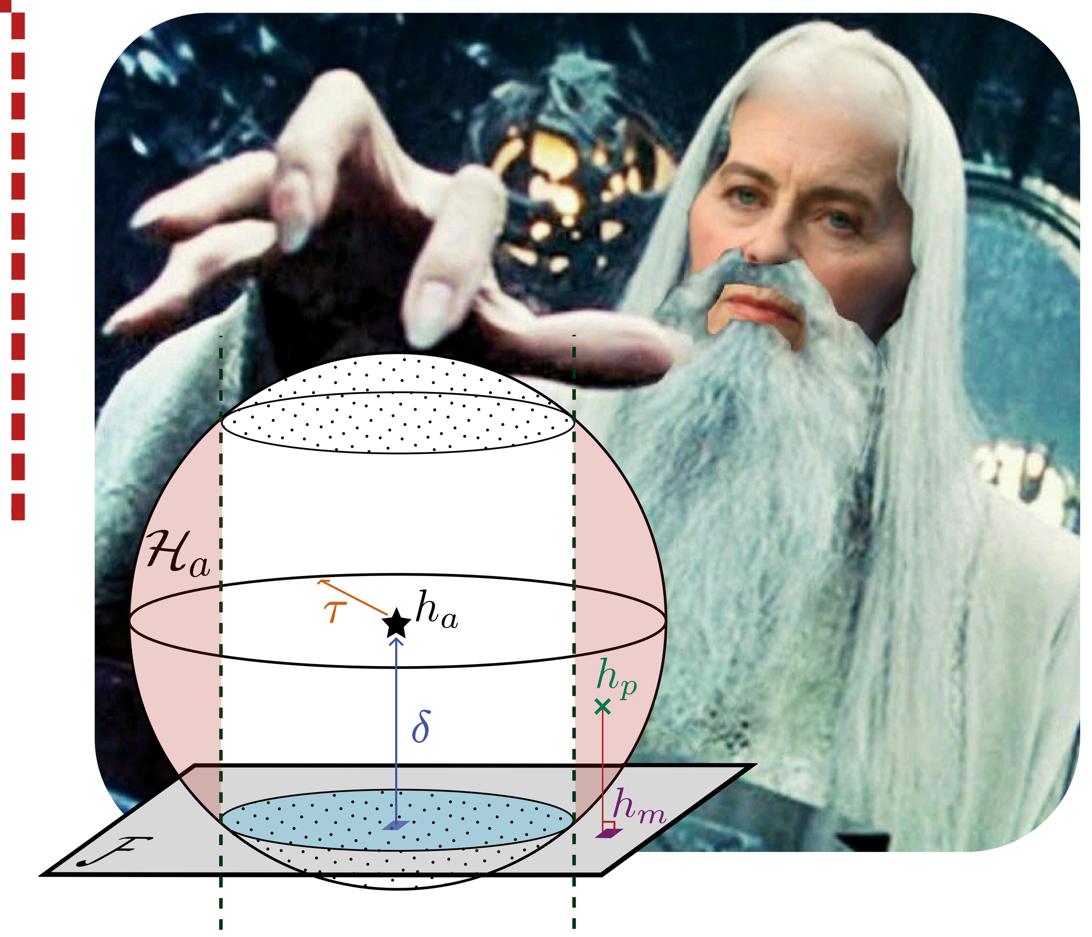
$\Rightarrow$ **This paper**: use prior knowledge (labeled data, pretrained models, public information about the platform) to detect manipulations.

The auditing game



Step ① The auditor builds audit set $S \subset \mathcal{X}$ and queries the platform.
Step ② The platform manipulates the audit: $h_p(S) \to h_m(S)$.
Step ③ The auditor runs the tests on collected data: $h_m \in \mathcal{F} \cap \mathcal{H}_a$?

One instance: the labeled dataset prior



Prior = labeled audit dataset D_a.

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

Base rate $\delta = d(h_a, \mathcal{F})$

Definition 4.2 (Detection rate).
Manipulation detected if $h_m \notin \mathcal{H}_a$.

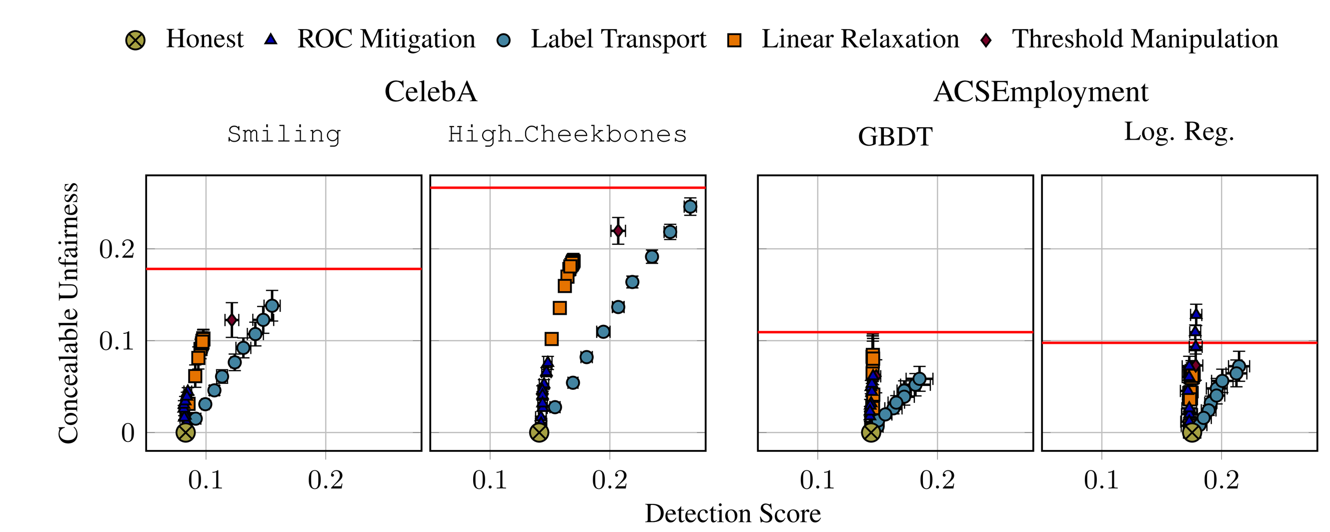
$$P_{\mathrm{uf}} = \mathbb{P}(h_m \notin \mathcal{H}_a \mid h_p \in \mathcal{H}_a)$$

- **Fair world.** If $\delta = 0$, $P_{\mathrm{uf}} = 0$
- **Single solution.** If $\delta = \tau$, $P_{\mathrm{uf}} = 1$

Corollary 4.4 (Choice of δ and τ).

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

Prevent fairwashing on `CelebA` and `folktables`?



Concealable unfairness

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

Detection score

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

Any good fairness repair is a good audit manipulation.

In practice, the auditor uses a statistic $\hat{\mu}(h_m, S)$ to test $h_m \overset{?}{\in} \mathcal{F}$. The platform only needs to manipulate h_p on the audit set S

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

- The platform can easily hide a lot of unfairness (if no verification).
- Lower entropy tasks: manipulations can be prevented.
- But the platform can hide in the noise.