Auditing Pay-Per-Token in Large Language Models

Ander Artola Velasco
Max Planck Institute
for Software Systems

Stratis Tsirtsis
Hasso Plattner Institute

Manuel Gomez Rodriguez
Max Planck Institute
for Software Systems

Abstract

Millions of users rely on a market of cloud-based services to obtain access to state-of-the-art large language models. However, it has been very recently shown that the de facto *pay-per-token* pricing mechanism used by providers creates a financial incentive for them to strategize and misreport the (number of) tokens a model used to generate an output. In this paper, we develop an auditing framework based on martingale theory that enables a trusted third-party auditor who sequentially queries a provider to detect token misreporting. Crucially, we show that our framework is guaranteed to *always* detect token misreporting, regardless of the provider's (mis-)reporting policy, and not falsely flag a faithful provider as unfaithful with high probability. To validate our auditing framework, we conduct experiments across a wide range of (mis-)reporting policies using several large language models from the `Llama`, `Gemma` and `Ministral` families, and input prompts from a popular crowdsourced benchmarking platform. The results show that our framework detects an unfaithful provider after observing fewer than $\sim$70 reported outputs, while maintaining the probability of falsely flagging a faithful provider below $\alpha = 0.05$.

1 INTRODUCTION

State-of-the-art large language models (LLMs) require a vast amount of resources, often involving hundreds or thousands of GPUs or TPUs, along with specialized infrastructure to handle massive parallel computations (Narayanan et al., 2021; Samsi et al., 2023; Jiang et al., 2024). As a consequence, most (enterprise) users cannot operate them locally, and instead rely on a rapidly growing market of cloud-based providers that offer LLMs-as-a-service (Chen et al., 2023; Snell et al., 2024; Pais et al., 2022; Patel et al., 2024).

In a typical LLM-as-a-service, a user submits a prompt to the provider via an application programming interface (API). Then, the provider feeds the prompt into an LLM running on their own hardware, which generates a sequence of tokens[1] as an output using a (non-deterministic) generative process. Finally, the provider shares the output with the user and charges them based on a simple pricing mechanism—a fixed price per token.[2]

Very recently, Velasco et al. (2025) have argued, both theoretically and empirically, that the above pricing mechanism creates a financial incentive for providers to strategize. Their key observation is that, in the interaction between a user and a provider, there is an asymmetry of information (Milgrom and Roberts, 1987; Rasmusen, 1989; Mishra et al., 1998), which enables a situation known in economics as moral hazard (Holmström, 1979). In particular, the provider observes the entire generative process used by the LLM to generate an output, whereas the user only observes and pays for the output shared with them by the provider. As a consequence, the provider has the opportunity to misreport the number of tokens in an output to increase their profit, at the expense of the user, and the user cannot know whether a provider is overcharging them.

The core of the problem lies in the fact that the tokenization of a string is not unique, and an LLM can in principle generate different tokenizations of the same string (Geh et al., 2024; Cao and Rimell, 2021; Chirkova et al., 2023). For example, consider that the user submits the prompt "`Where does the next AISTATS take place?`" to the provider, the provider feeds it

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

[1] Tokens are units that make up sentences and paragraphs, such as (sub-)words, symbols and numbers.

[2] `https://ai.google.dev/gemini-api/docs/pricing`, `https://openai.com/api/pricing/`.

into an LLM, and the model generates the output "|Tang|ier|,| Morocco|" consisting of four tokens. Exploiting the asymmetry of information, a self-serving provider could simply claim that the LLM generated the tokenization "|Tang|ier|,| |Mor|oc|co|" and overcharge the user for seven tokens instead of four!

In this paper, we consider a forward-looking yet realistic[3] scenario in which, to eliminate the incentive for providers to engage in misreporting, they are required to share information about the generative process with a third-party trusted auditor who can verify they are faithful.

Our contributions. We start by formalizing the problem of auditing for token misreporting by a provider as a sequential hypothesis test. In doing so, we consider that the third-party trusted auditor has access to the next-token probability distribution of the model served by the provider. Then, building upon this formalization, we introduce an auditing framework that uses a sequential statistical test based on the theory of martingales (Ramdas et al., 2023) to detect token misreporting. Along the way, we develop a novel, unbiased, and efficient estimator of the average length of the token sequences used by a model to encode any given output string, which our statistical test uses and may be of independent interest. Further, we provide sufficient conditions under which our framework is guaranteed to eventually detect an unfaithful provider, regardless of their (mis-)reporting policy, and not falsely flag faithful providers as unfaithful with high probability.

To validate our auditing framework, we conduct experiments using several large language models from the `Llama`, `Gemma` and `Ministral` families, and input prompts from a popular crowdsourced benchmarking platform. The results show that, for several (mis-)reporting policies introduced in prior work (Velasco et al., 2025), our auditing framework detects an unfaithful provider after observing just up to $\sim$70 reported outputs, while maintaining the probability of falsely flagging a faithful provider below a prespecified threshold $\alpha = 0.05$.[4]

Further related work. Our work builds upon further related work on the economics of LLMs-as-a-service, tokenization, and sequential statistical tests.

[3]According to Article 74(13) of the EU AI Act, "market surveillance authorities shall be granted access to the source code of the high-risk AI system [...] when testing or auditing procedures and verifications based on the data and documentation provided by the provider have been exhausted or proved insufficient."

[4]The code for our experiments is publicly available at `https://github.com/Human-Centric-Machine-Learning/token-audit`.

Within the rapidly growing literature on the economics of LLMs-as-a-service (La Malfa et al., 2024; Mahmood, 2024; Laufer et al., 2024; Cai et al., 2025; Saig et al., 2024; Bergemann et al., 2025; Velasco et al., 2025), there has been increasing interest in the ways in which providers may strategically act at the expense of users. Within this literature, our work is most closely related to a line of work on algorithmic auditing (Bourrée et al., 2025; Sun et al., 2025; Wang et al., 2025), which has focused on detecting whether an LLM provider is unfaithful about the model they serve or the token counts during hidden reasoning steps. Yet, this line of work has largely overlooked the possibility that an LLM provider may be unfaithful about the tokenization of the outputs—a threat that has not been studied until very recently (Velasco et al., 2025).

Multiple lines of empirical evidence have shown that tokenization plays a central role in developing and analyzing LLMs (Geh et al., 2024; Giulianelli et al., 2024; Geh et al., 2025; Petrov et al., 2023; Ovalle et al., 2024; Chatzi et al., 2025). Consequently, there have been numerous efforts to better understand and improve byte-pair encoding (BPE), the tokenization algorithm most commonly used in LLMs (Bostrom and Durrett, 2020; Zouhar et al., 2023; Lian et al., 2024b; Sennrich et al., 2016; Lian et al., 2024a). However, this line of work has not studied the economic implications of tokenization (in the context of LLMs-as-a-service), which is the main focus of our work.

Our work also builds upon the active and expanding body of research on sequential statistical testing with martingales and e-values (Ramdas et al., 2023; Ramdas and Wang, 2025; Waudby-Smith et al., 2025), which has derived stronger guarantees than classical testing approaches and has been successfully applied to a wide range of statistical problems (Shin et al., 2024; Shekhar and Ramdas, 2024; Xu and Ramdas, 2024). In a concurrent work, Gauthier et al. (2026) obtain results similar to ours at a technical level. However, they focus on detecting deviations from Nash equilibria in multi-agent games. To the best of our knowledge, we are the first to use techniques based on e-values in the context of auditing LLMs-as-a-service.

2 AUDITING FOR TOKEN MISREPORTING AS A SEQUENTIAL HYPOTHESIS TEST

We model the process of auditing a provider for token misreporting as a sequential interaction between the provider serving an LLM $\mathcal{M}$ and an auditor. At each time step $i = 1, 2, \ldots$, the auditor selects a prompt $Q \sim P^Q$ from a fixed prompt distribution and queries

the provider for a response to Q.[5] The provider then generates a sequence of tokens $\mathbf{T} \in \mathcal{V}^*$ by autoregressively sampling one token at a time, where $\mathcal{V}^*$ is the set of finite sequences of tokens in the vocabulary of tokens $\mathcal{V}$ used by the LLM $\mathcal{M}$.

Importantly, since only the provider observes the sequence $\mathbf{T}$, they have the capacity to report a different sequence of tokens $\widetilde{\mathbf{T}} \sim \pi(Q, \mathbf{T})$ using a (non-deterministic) reporting policy π, which is unknown to the auditor.[6] While, in principle, a provider can choose any reporting policy they wish, we narrow our focus to reporting policies that misreport the tokens in $\mathbf{T}$ *while preserving its string-level representation*, similarly as in Velasco et al. (2025); that is, $\texttt{str}(\widetilde{\mathbf{T}}) = \texttt{str}(\mathbf{T})$ for any $\widetilde{\mathbf{T}} \sim \pi(Q, \mathbf{T})$, where $\texttt{str}\colon \mathcal{V}^* \to \mathcal{S}$ maps a sequence of tokens to the respective string, and $\mathcal{S}$ denotes the set of all possible strings. Here, it is also important to note that, for a given prompt q and sequence of tokens $\mathbf{t}$, the provider can only obtain a financial benefit if

$$\mathbb{E}_{\widetilde{\mathbf{T}} \sim \pi(q, \mathbf{t})} \Big[\texttt{len}(\widetilde{\mathbf{T}}) \Big] \geq \texttt{len}(\mathbf{t}),$$

where $\texttt{len}\colon \mathcal{V}^* \to \mathbb{N}$ denotes the number of tokens in a given sequence. This is because, under the de facto standard pay-per-token pricing, the price charged to users for a reported output sequence $\widetilde{\mathbf{T}}$ increases linearly with its length. Therefore, we can naturally characterize the financial benefit the provider obtains from misreporting by measuring the average number of additional tokens in $\widetilde{\mathbf{T}}$ compared to $\mathbf{T}$ across all prompts q sampled from P^Q, which we refer to as the *misreporting intensity* $\mathcal{I}(\pi)$, *i.e.*,

$$\mathcal{I}(\pi) = \mathbb{E}_{Q \sim P^Q} \Big[\mathbb{E}_{\mathbf{T} \sim P^{\mathcal{M}}(\cdot \,|\, q)} \Big[\Delta_\pi(\mathbf{t}) \,|\, \mathbf{T} = \mathbf{t} \Big] \,|\, Q = q \Big], \quad (1)$$

where

$$\Delta_\pi(\mathbf{t}) = \mathbb{E}_{\widetilde{\mathbf{T}} \,\sim\, \pi(q, \mathbf{t})} \Big[\texttt{len}(\widetilde{\mathbf{T}}) \Big] - \texttt{len}(\mathbf{t}), \quad (2)$$

and $P^{\mathcal{M}}(\cdot \,|\, q)$ denotes the probability distribution of the sequences $\mathbf{T}$ generated by the LLM $\mathcal{M}$ in response to the prompt q. Intuitively, the misreporting intensity measures the severity of the misreporting and hence, for an auditor, it is more critical to detect misreporting policies with high intensity—a policy satisfying $\mathcal{I}(\pi) \approx 0$ would minimally harm the user. Moreover, as we will demonstrate later, both theoretically and empirically, the difficulty of detecting if a provider using an unknown policy π is engaging in misreporting is fundamentally determined by $\mathcal{I}(\pi)$.

[5] We denote random variables with capital letters (X) and their realizations with lower case letters (x).

[6] In principle, a provider could use, at each time step i, a different reporting policy. We discuss this possibility in Section 5.

Under the above characterization, auditing for token misreporting can be framed as a (sequential) hypothesis test on the misreporting intensity, where an auditor (sequentially) gathers sufficient statistical evidence to conclude that the lengths of the reported sequences $\widetilde{\mathbf{T}}$ do not *match*, in expectation, the lengths of the sequences $\mathbf{T}$ generated by the model. More concretely, we can define the following null and alternative hypotheses:

$$\begin{cases} H_0 = \{\pi_0\} & \text{(null)} \\ H_1 = \{\pi : \mathcal{I}(\pi) > 0\} & \text{(alternative)} \end{cases}, \quad (3)$$

where π_0 is the policy that faithfully reports tokens in $\mathbf{T}$, *i.e.*, $\pi_0(Q, \mathbf{T}) \coloneqq \mathbf{T}$, with $\mathcal{I}(\pi_0) = 0$, and note that the hypothesis H_1 includes any family of (non-deterministic) reporting policies, potentially of arbitrary sophistication.

In the next section, we will develop a framework to test the above hypothesis in a setting in which the auditor has access to the next-token probabilities of the model.[7] Such a setting fits a variety of real-world scenarios, for example, a scenario in which the provider serves an open-weight model, or a scenario in which the provider is required, by regulation, to grant trusted auditors access to a proprietary model (refer to Section 5 for further discussion on this assumption).

3 A SEQUENTIAL HYPOTHESIS TEST AUDITING FRAMEWORK

Our starting point is the key observation that, as long as the reporting policy π preserves the string-level representation $\boldsymbol{S}$ of the generated sequences $\mathbf{T}$, the inner expectation in the misreporting intensity defined by Eq. 1 can be expressed as follows:

$$\mathbb{E}_{\boldsymbol{S} \sim P^{\mathcal{M}}(\cdot \,|\, q)} \Big[\mathbb{E}_{\mathbf{T} \sim P^{\mathcal{M}}_{\mathbf{s}}(\cdot \,|\, q)} \Big[\Delta_\pi(\mathbf{t}) \,|\, \mathbf{T} = \mathbf{t} \Big] \,|\, \boldsymbol{S} = \mathbf{s} \Big],$$

where $\Delta_\pi(\mathbf{t})$ is as in Eq. 2 and $P^{\mathcal{M}}_{\mathbf{s}}(\cdot \,|\, q)$ denotes the conditional distribution of the sequences $\mathbf{T}$ generated by the LLM $\mathcal{M}$ in response to the prompt q whose string-level representation matches $\mathbf{s}$. As a consequence, auditing for token misreporting reduces to gathering sufficient (statistical) evidence to conclude that the lengths of the reported sequences $\widetilde{\mathbf{T}}$ do not *match*, in expectation, the average length of the sequences $\mathbf{T}$ used by the LLM $\mathcal{M}$ to encode the respective string $\mathbf{s} = \texttt{str}(\widetilde{\mathbf{T}})$, *i.e.*, they do not match $\mathbb{E}_{\mathbf{T} \sim P^{\mathcal{M}}_{\mathbf{s}}(\cdot \,|\, q)} [\texttt{len}(\mathbf{T})]$.

[7] Although this is necessary for our theoretical analysis, in practice, our auditing framework performs similarly in settings where the auditor has access to approximate values of next-token probabilities (see Appendix D.4).

In what follows, we will first introduce an efficient and unbiased estimator of the conditional average mentioned above for any given string $\mathbf{s}$, and then leverage this estimator to design a statistical (sequential) test to determine whether a provider is engaging in token misreporting, that is, to test H_0 against H_1.

3.1 Estimating the Average Length of Token Sequences Encoding a Given String

To estimate the average length $\mathbb{E}_{\mathbf{T}\sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)}[\texttt{len}(\mathbf{T})]$, we first modify the autoregressive generation process used by the LLM $\mathcal{M}$ so that it always generates sequences of tokens $\mathbf{T}$ whose string-level representation satisfies $\texttt{str}(\mathbf{T}) = \mathbf{s}$, and then use this modified process to compute an unbiased Monte Carlo estimate of the average token sequence length.

Let $P^{\mathcal{M}}(t\,|\,q, \mathbf{t})$ denote the next-token probability that the LLM $\mathcal{M}$ assigns to a token t given a prompt q and a partial output token sequence $\mathbf{t}$. The modified generation process instead utilizes the following *masked* next-token probability:

$$\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(t\,|\,q, \mathbf{t}) := \begin{cases} P^{\mathcal{M}}(t\,|\,q, \mathbf{t})/Z & \text{if } \mathbf{t} \shortmid t \models \mathbf{s} \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\mathbf{t} \shortmid t \models \mathbf{s}$ indicates that the string-level representation of the concatenated sequence $\mathbf{t} \shortmid t$ is a prefix of $\mathbf{s}$, and $Z = \sum_{t\in\mathcal{V}:\ \mathbf{t}\shortmid t\models\mathbf{s}} P^{\mathcal{M}}(t\,|\,q, \mathbf{t})$ is a normalization constant that ensures that the values $\widehat{P}_{s}^{\mathcal{M}}(t\,|\,q, \mathbf{t})$ lead to a valid probability distribution over the vocabulary $\mathcal{V}$. In words, the modified generation process simply masks any token that would lead to a string different than $\mathbf{s}$. Even though one could think otherwise, the distribution of token sequences $\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)$ generated using the above modified process does not in general match the conditional distribution $P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)$, as recently noted by Lipkin et al. (2025). However, perhaps surprisingly, it can be used to efficiently construct an unbiased estimate of the average length $\mathbb{E}_{\mathbf{T}\sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)}[\texttt{len}(\mathbf{T})]$, as we show next.

Let $K \sim P^K$, where P^K is a distribution with support over $\mathbb{N}$, and $\widehat{\mathbf{T}}_1, \ldots, \widehat{\mathbf{T}}_K$ be K independent samples from $\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)$, generated using the modified generation process mentioned above. Moreover, define $R_0 = 0$ and, for $k \in \{1, \ldots, K\}$, let R_k be the weighted average

$$R_k = \frac{\sum_{j=1}^{k} w_j \cdot \texttt{len}(\widehat{\mathbf{T}}_j)}{\sum_{j=1}^{k} w_j}, \tag{5}$$

where $w_j = P^{\mathcal{M}}(\widehat{\mathbf{T}}_j\,|\,q)/\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}}_j\,|\,q)$ is the relative likelihood of the j-th sample between the original and the modified generation process of the model $\mathcal{M}$. Then, as formalized by Algorithm 1 and Proposition 1, we can efficiently construct an unbiased estimate of the average length $\mathbb{E}_{\mathbf{T}\sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)}[\texttt{len}(\mathbf{T})]$ using a weighted sum of the increments $R_k - R_{k-1}$.[8]

Algorithm 1 It returns an unbiased estimate of the average length $\mathbb{E}_{\mathbf{T}\sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)}[\texttt{len}(\mathbf{T})]$.

function EstimateLength(Prompt q, output string $\mathbf{s}$, next-token probability $P^{\mathcal{M}}$, distribution P^K)
  Sample $K \sim P^K$
  for $k = 1, \ldots, K$ **do**
    Sample $\widehat{\mathbf{T}}_k \sim \widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)$
    $w_k \leftarrow \frac{P^{\mathcal{M}}(\widehat{\mathbf{T}}_k|q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}}_k|q)}$
    $Z_k \leftarrow \sum_{j=1}^{k} w_j$
  end for
  $R_0 \leftarrow 0$
  for $k = 1, \ldots, K$ **do**
    $w'_k \leftarrow \frac{w_k}{Z_k}$
    $R_k \leftarrow \sum_{j=1}^{k} w'_j \cdot \texttt{len}(\widehat{\mathbf{T}}_j)$
  end for
  Return $\sum_{k=1}^{K} \frac{R_k - R_{k-1}}{P^K(K \geq k)}$
end function

Proposition 1. *Let $K \sim P^K$ and R_k be defined by Eq. 5 for each $k \in \{0, \ldots, K\}$. Then, it holds that:*

$$\mathbb{E}_{K\sim P^K,\, \widehat{\mathbf{T}}_k \sim \widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)} \left[\sum_{k=1}^{K} \frac{R_k - R_{k-1}}{P^K(K \geq k)} \right] = \mathbb{E}_{\mathbf{T}\sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)}[\texttt{len}(\mathbf{T})]\,.$$

In the next section, we will show how an auditor who sequentially queries the provider can leverage the above estimator to conclude that the provider is (not) engaging in misreporting.

3.2 Constructing a Sequential Statistical Test via Martingales

To determine if a provider is misreporting, our framework compares the length of the reported sequence of tokens $\widetilde{\mathbf{T}}$ as a response to a prompt Q with the estimator of the average length of the sequences used by the LLM to encode the string $\texttt{str}(\widetilde{\mathbf{T}})$ using Algorithm 1. More concretely, our framework computes the quantity

$$E = \texttt{len}(\widetilde{\mathbf{T}}) - \texttt{EstimateLength}(Q, \texttt{str}(\widetilde{\mathbf{T}}), P^{\mathcal{M}}, P^k), \tag{6}$$

which represents the (statistical) evidence in favor of rejecting the hypothesis H_0, *i.e.*, flagging the provider as unfaithful. As the sequential interaction between the auditor and the provider unfolds, our framework aggregates the observed evidence E against H_0 at each

[8] All proofs can be found in Appendix A.

time step using a (stochastic) process M defined as follows:

$$M_i = \begin{cases} 1 & i = 0 \\ M_{i-1} \cdot (1 + \lambda_i \cdot E_i) & i \geq 1, \end{cases} \tag{7}$$

where each E_i is defined as in Eq. 6 for the respective prompt Q_i and reported sequence $\widetilde{\mathbf{T}}_i$ at time i, and $\lambda_i \geq 0$ is a given (potentially time-varying) parameter that weighs the observed evidence. Here, if one chooses $\lambda_i \approx 0$, the process M stays constant, ignoring any evidence, and, if one chooses a high value of λ_i, the process M is very sensitive to any evidence. Later, we will show that, while in principle high values of λ_i are desirable to quickly detect misreporting, the auditor may also risk incorrectly flagging a truthful provider as suspicious, and we will discuss a prescription to address this trade-off.

Leveraging the process M, our framework flags the provider as unfaithful as soon as the aggregated evidence exceeds a predefined threshold. More formally, it rejects H_0 using the following (sequential) statistical test:

$$\phi_\alpha = \mathbb{1}\left\{\exists i \in \mathbb{N} \colon M_i > \frac{1}{\alpha}\right\}, \tag{8}$$

where $\alpha \in (0, 1)$ and the threshold $1/\alpha$ controls how much evidence needs to be gathered by the framework to conclude that the provider is (mis-)reporting tokenizations. Algorithm 2 summarizes the overall procedure followed by our framework.

Importantly, since the function `EstimateLength` returns unbiased estimates of the average length of the token sequences used by the LLM $\mathcal{M}$ to encode a given string $\mathbf{s}$, we can first show that, if the provider is faithful, the process M is a martingale, as formalized by the following proposition:

Proposition 2. *Assume the provider is faithful and implements the reporting policy π_0. Then, for any sequence λ_i, the process M defined by Eq. 7 is a martingale. That is, for each time step i, it holds that:*

$$\mathbb{E}_{H_0}\left[M_i \,|\, M_{i-1}\right] = M_{i-1}$$

Put differently, the above result reveals that, on average, if the provider is faithful, the evidence for misreporting given by the process M does not increase over time. Consequently, one may expect the test ϕ_α defined by Eq. 8 not to flag a faithful provider as unfaithful. The next theorem formalizes this expectation by giving sufficient conditions under which the test ϕ_α provably controls the false positive rate $P_{H_0}(\phi_\alpha = 1)$, that is, the probability of falsely flagging a faithful provider.

Theorem 3. *If $1 + \lambda_i \cdot E_i \geq 0$ for all $i \geq 1$ under H_0, then, it holds that $P_{H_0}(\phi_\alpha = 1) \leq \alpha$.*

Algorithm 2 It audits a provider for token misreporting.

Input Distribution of prompts P^Q, next-token probabilities $P^{\mathcal{M}}$, sequence $\lambda_i \in \mathbb{R}_+$, bound on the false positive rate $\alpha \in (0, 1)$, distribution P^K.
Initialize $M_0 \leftarrow 1$, `misreport` $\leftarrow$ False
for $i = 1, 2, \ldots$ **do**
  Sample $q_i \sim P^Q$
  Query the provider with q_i to obtain $\tilde{\mathbf{t}}_i$
  $s_i \leftarrow \texttt{str}(\tilde{\mathbf{t}}_i)$
  ▷ Estimate the avg. length of such tokenizations
  $l_i \leftarrow \texttt{EstimateLength}(q_i, s_i, P^{\mathcal{M}}, P^K)$
  ▷ Compute the evidence for misreporting
  $E_i \leftarrow \texttt{len}(\tilde{\mathbf{t}}_i) - l_i$
  ▷ Aggregate all evidence so far
  $M_i \leftarrow M_{i-1} \cdot (1 + \lambda_i \cdot E_i)$
  ▷ If the aggregated evidence is sufficient, flag the provider as misreporting
  if $M_i > 1/\alpha$ **then**
    `misreport` $\leftarrow$ True
    return `misreport`
  end if
end for

Intuitively, the above theorem tells us that, as long as the weights λ_i are sufficiently small relative to the negative values of E_i originating from sampled tokenizations $\mathbf{t}$ with below-average length, then the likelihood that the process M exceeds the threshold $1/\alpha$ remains below α. However, it does not rule out the possibility that an unfaithful provider goes undetected. In what follows, we provide sufficient conditions under which the number of time steps $\tau = \inf\{i \colon M_i > 1/\alpha\}$ needed for our test ϕ_α to flag an unfaithful provider is finite:

Theorem 4. *The (sequential) statistical test ϕ_α satisfies the following:*

i) ***Decreasing*** *λ_i: Let $\lambda_i = \lambda_0 / i > 0$ for all $i \geq 1$. If $1 + \lambda_0 \cdot E > 0$ under H_0, then, it holds that*

$$P_{H_1}(\tau < \infty) = 1 \;\; \text{and} \;\; \mathbb{E}_{H_1}[\tau] < \infty.$$

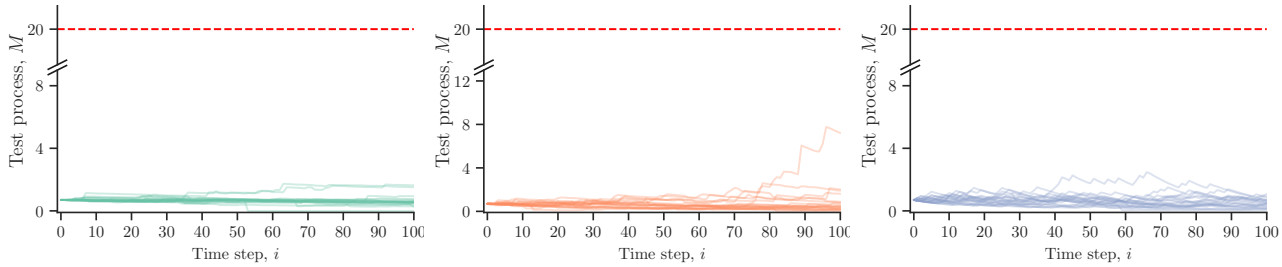
ii) ***Constant*** *λ_i: Let $\lambda_i = \lambda_0 > 0$ for all $i \geq 1$. If $1 + \lambda_0 \cdot E \in (B_-, B_+)$ with $B_- > 0$ under H_0 and*

$$\log(1 + \lambda_0 \cdot \mathcal{I}(\pi)) > \mathrm{Var}(E) \cdot \frac{\lambda_0^2}{2(\mathrm{B}_-)^2} \;\; \text{under } H_1,$$

then, it holds that $P_{H_1}(\tau < \infty) = 1$ and

$$\mathbb{E}_{H_1}[\tau] \leq \frac{\log 1/\alpha + \log \mathrm{B}_+}{\log(1 + \lambda_0 \cdot \mathcal{I}(\pi)) - \mathrm{Var}(E) \cdot \frac{\lambda_0^2}{2(\mathrm{B}_-)^2}}.$$

The above theorem gives sufficient conditions for the execution of Algorithm 2 to terminate. Loosely speaking, at each time step i, the average evidence that the



(a) `Llama-3.2-1B-Instruct` (b) `Ministral-8B-Instruct-2410` (c) `Gemma-3-1B-It`

Figure 1: **Auditing faithful providers.** The panels show realizations of the test process M for three (simulated) faithful providers, each serving a different large language model. In each realization, we sequentially query the provider using prompts picked uniformly at random from the LMSYS Chatbot Arena dataset, and compute M_i using Eq. 7 with $\lambda = 0.07, 0.13$ and 0.19, respectively. In all panels, the dashed line illustrates the threshold $1/\alpha$ needed to flag a provider and, for clarity, we display 30 realizations randomly sampled from a total of 150. Moreover, we set the false positive rate bound to $\alpha = 0.05$ and the temperature of the models to 1. Refer to Appendix D for qualitatively similar results using other temperature values.

framework observes is $\mathbb{E}_{H_1}[1+\lambda_i \cdot E_i] = 1+\lambda_i \cdot \mathcal{I}(\pi)$ and, if the provider indeed misreports and λ_i is sufficiently small, the process M_i grows exponentially fast and eventually surpasses the threshold $1/\alpha$. Importantly, Theorem 4 makes no assumption on the specific form of the provider's reporting policy π, and it ensures our auditing framework enjoys worst-case guarantees based on the magnitude $\mathcal{I}(\pi)$ of misreporting.

In the next section, we validate our auditing framework by auditing both providers that faithfully report tokenizations and providers that misreport with positive intensity.

Remark. We have considered two simple choices for the sequence λ_i, namely constant and proportional to $1/i$, however, one could, in principle, construct a more sophisticated weighting sequence to lower the number of steps $\mathbb{E}_{H_1}[\tau]$ needed to detect an unfaithful provider. In particular, to detect an unfaithful provider in the lowest number of steps, the auditor could consider the weight λ_i, which, in hindsight, would have led to the higher (logarithmic) growth for the process $\mathcal{M}$, *i.e.*,

$$\lambda_i = \min\left(\operatorname*{argmax}_{\lambda \geq 0} \frac{1}{i-1} \sum_{s=1}^{i-1} \log\left(1 + \lambda \cdot E_s\right), \lambda_-\right),$$

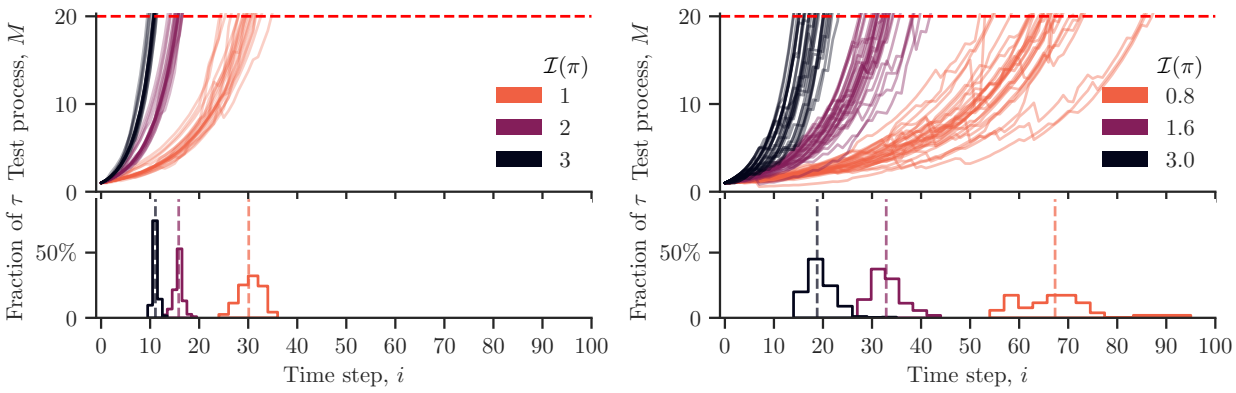
where λ_- is the largest weight such that $1+\lambda_- \cdot E \geq 0$, which is required for Theorem 3 to hold. The above construction can be shown to be asymptotically optimal in terms of detection time (Waudby-Smith et al., 2025). Unfortunately, obtaining meaningful bounds on $\mathbb{E}_{H_1}[\tau]$ that generalize Theorem 4 becomes challenging.

4 EXPERIMENTS

In this section, we use our framework to audit several (simulated) providers who serve large language models from the `Llama`, `Gemma` and `Ministral` families. Here, we experiment with both providers who are faithful and use the faithful reporting policy π_0, and providers who are unfaithful and use the (mis-)reporting policies introduced by Velasco et al. (2025).

Experimental setup. To instantiate our auditing framework, we sequentially query the (simulated) providers, who serve one model from the above-mentioned families, using prompts picked uniformly at random from a set of 4,000 prompts from the LMSYS Chatbot Arena platform dataset (Zheng et al., 2024). Refer to Appendix B for further details regarding the dataset and models used by the (simulated) providers. Within our framework, we use a Poisson distribution P^K with parameter 7 to control the number of samples used by our estimator in Algorithm 1,[9] and set $\lambda_i = \lambda$ to a model specific constant value for all $i \geq 1$, which we determine as follows. First, we generate 400 realizations of the random variable E defined in Eq. 6 using the faithful reporting policy π_0 on prompts picked uniformly at random from a held-out set of 400 prompts from the LMSYS Chatbot Arena platform dataset. Then, we compute the largest value of λ (namely, λ_-) that ensures that $1 + \lambda \cdot E$ is positive for every realization of E, and set $\lambda = 0.9\lambda_-$. This results in $\lambda = 0.07, 0.13$ and 0.19 for `Llama-3.2-1B-Instruct`,

[9]We have experimented with Poisson and geometric distributions and have found that $K \sim \text{Poisson}(7)$ performs the best in achieving low variance in the estimator with a small number of samples. However, note that Proposition 1 guarantees that our estimator is unbiased for any distribution P^K with support over $\mathbb{N}$.



(a) Random policies

(b) Heuristic policies

Figure 2: **Auditing an unfaithful provider who serves `Llama-3.2-1B-Instruct`.** The two panels show realizations of the test process M (top) and the distribution of detection times $\tau = \inf\{i : M_i > 1/\alpha\}$ (bottom) when the provider uses, respectively, random and heuristic policies π of varying intensity $\mathcal{I}(\pi)$. For the heuristic policy, we report an estimate of the intensity by averaging the number of additional tokens that it reports. In each realization, we sequentially query the provider using prompts picked uniformly at random from the LMSYS Chatbot Arena dataset, and compute M_i using Eq. 7 with $\lambda = 0.07$. In each panel, the three different intensity values correspond to policies π parameterized by $m = 1, 2, 3$, with higher values of m leading to higher (darker) intensities and, for each m, we show 30 realizations. In all panels, we set the false positive rate bound to $\alpha = 0.05$ and the temperature of the models to 1. Refer to Appendix D for qualitatively similar results using other models and temperature values.

`Ministral-8B-Instruct-2410` and `Gemma-3-1B-It`, respectively. We empirically verify that this choice of λ always leads to positive realizations of the process M, as required by Theorem 3. Lastly, we set the false positive rate bound α to 0.05, and terminate each audit once the testing process M_i exceeds the threshold $1/\alpha$ or the audit performs a maximum of 100 iterations.

Can our audit falsely flag a faithful provider as unfaithful? To answer this question, we audit several (simulated) providers who serve each a different model and use the faithful reporting policy π_0. We repeat each audit 150 times and, each time, we measure how the test process M evolves over time. Figure 1 summarizes the results, which show that, as expected from the martingale property of M shown in Proposition 2, the vast majority of realizations of the test process M remain very close to their initial value $M_0 = 1$, with no apparent drift towards larger positive values. Moreover, the results also show that, across all (simulated) providers, none of the realizations of the test process M exceed the threshold of $1/\alpha$. This suggests that, although our audit is designed to have a false positive rate at most $\alpha = 0.05$ based on Theorem 3, the probability of falsely flagging a faithful provider as unfaithful is negligible in practice.[10]

[10]This may be due to the fact that Theorem 3 relies on Ville's inequality (Ville, 1939), which leads to conservative upper bounds.

Can our audit successfully detect an unfaithful provider? To answer this question, we audit several (simulated) providers who serve each a different model and use two different types of (mis-)reporting policies (Velasco et al., 2025). These policies construct a tokenization $\tilde{\mathbf{t}}$ of the string $\texttt{str}(\mathbf{t})$ by iteratively selecting m tokens and splitting them into two separate tokens. We briefly describe the two types of misreporting policies below and provide their full description in Appendix C:

- **Random:** For each generated output, it iteratively selects a random token to modify and splits it into a random pair of tokens (if any). It terminates once it has performed m splits and reports the modified sequence $\tilde{\mathbf{t}}$. As a consequence, the misreporting intensity $\mathcal{I}(\pi)$ for a policy π of this type is (approximately) m.

- **Heuristic:** For each generated output, it prioritizes reported tokenizations that are not very unlikely to be generated. To this end, in each iteration, it selects the token with the highest index in the vocabulary and splits it into a pair of tokens with the highest minimum index. Once it has performed m successful splits resulting in an (intermediate) modified token sequence $\mathbf{t}'$, the policy computes the next-token probabilities given by the model $\mathcal{M}$ at each point in $\mathbf{t}'$ and verifies if

each token is in its respective top-p set (Holtzman et al., 2020). If this condition is satisfied, the policy reports $\tilde{\mathbf{t}} = \mathbf{t}'$ to the user; otherwise, it falls back to $\tilde{\mathbf{t}} = \mathbf{t}$. Consequently, the misreporting intensity $\mathcal{I}(\pi)$ for a policy π of this type is at most m, depending on the fraction of modified tokenizations $\mathbf{t}'$ that pass the verification step, and it can be estimated by computing the average number of additional tokens reported.

Figure 2 summarizes the results for an unfaithful provider who serves a model from the `Llama` family. Refer to Appendix D for qualitatively similar results for unfaithful providers who serve other models. We find that, in all cases, our auditing framework succeeds in detecting token misreporting (*i.e.*, the test process M crosses the threshold $1/\alpha$). Importantly, the results also show that our framework detects token misreporting after observing fewer than $\sim$70 reported outputs. Moreover, consistent with Theorem 4, we observe that the average number of necessary reported outputs to detect token misreporting decreases rapidly as the misreporting intensity $\mathcal{I}(\pi)$ increases. However, we also find that auditing an unfaithful provider who uses the heuristic policy rather than the random policy generally leads to a higher variance in the number of queries needed to flag them, which, in some cases, can increase the time and resources needed to detect misreporting.

5 DISCUSSION AND LIMITATIONS

In this section, we highlight several limitations of our work and discuss avenues for future research as well as its broader impact.

Model access. Our auditing framework's strong theoretical guarantees (*i.e.*, Theorems 3 and 4) require sandboxed or regulatory access to the language model served by the provider. We believe that some form of model access is necessary for any effective auditing method since detecting manipulations in reported token sequences fundamentally requires a baseline reflecting the behavior of the unmanipulated model; in our framework, the `EstimateLength` estimator provides precisely such a baseline. Nevertheless, in future work, it would be very interesting to develop auditing techniques relying on weaker forms of model access. For example, one promising direction is to investigate whether access limited to the model's tokenizer could still enable meaningful audits. In this context, it is also important to note that, in high-stakes settings, regulatory regimes such as the EU AI Act, Article 74(13) already foresee that trusted third-party auditors, whether public or private, may be granted privileged access to proprietary models.

Reporting policies. In our theoretical analysis, we have assumed that the reporting policy is static because this allows us to obtain closed-form bounds for the number of steps $\mathbb{E}_{H_1}[\tau]$ required to detect misreporting in Theorem 4. However, our auditing framework can, in principle, be instantiated even if the provider varies their reporting policy over time. This is because the validity of Theorem 4 (*i.e.*, the condition $1+\lambda_i \cdot E_i \geq 0$ under H_0) does not require any assumption on the provider's reporting policy, and because the estimator in Algorithm 1 remains unbiased at each time step i, regardless of how the policy may change at the next step $i+1$. That said, a non-static policy would require a more refined theoretical analysis to obtain guarantees on $\mathbb{E}_{H_1}[\tau]$. Intuitively, if the provider uses a time-varying misreporting policy π_i at each time step i, then, on expectation, the test process M is multiplied at each step by the quantity $1 + \lambda_i \cdot \mathcal{I}(\pi_i)$, where the misreporting intensity $\mathcal{I}(\pi_i)$ is no longer constant. Thus, if the intensities decay over time (*e.g.*, as the provider becomes more cautious), detection is expected to become increasingly difficult—though misreporting also becomes correspondingly less harmful to the user.

Further, we would like to emphasize that our auditing framework can use historical billing data collected offline, *i.e.*, lines 4 and 5 in Algorithm 2 can take place before the provider learns they will be audited. For example, the reported tokenizations may come from users who suspect misreporting and save the tokenizations reported to them. As a consequence, even if the provider has stopped misreporting at the time the audit is taking place, they may be unable to evade detection if they have misreported in the past.

Evaluation. We have conducted experiments with state-of-the-art open-weight LLMs from the `Llama`, `Gemma` and `Ministral` families. However, it would be interesting to evaluate of our auditing framework using proprietary LLMs, as well as conducting evaluations with real providers. Such evaluations are out of scope of the current work since they would come with additional technical and regulatory challenges; however, we believe they may prove useful at building trust between users and providers in LLM-as-a-service. Moreover, we have validated our framework against the set of (mis-)reporting policies recently introduced by Velasco et al. (2025). In practice, however, providers may employ other policies. Documenting and systematizing such policies—similarly to how the jailbreaking literature collects adversarial prompts (Rao et al., 2024; Shen et al., 2024)—would provide a valuable benchmark for future auditing frameworks.

Broader impact. As the multi-billion-dollar market of LLM-as-a-service keeps growing, it is increasingly

critical to develop statistical techniques for algorithmic auditing, especially since the provider's incentives under the current pay-per-token pricing model are misaligned with those of end-users (Velasco et al., 2025). Our work enables a trusted third-party auditor with access to the LLMs served by providers to detect token misreporting regardless of the (mis)-reporting policy used by unfaithful providers, however, the potential impact of our framework will (partially) depend on the timely development of regulatory frameworks for LLMs and, more broadly, generative AI.

6 CONCLUSIONS

In this work, we have introduced a first-of-a-kind auditing framework that enables a third-party trusted auditor with access to the LLM served by a provider to detect token misreporting. Along the way, we developed an unbiased estimator of the average length of token sequences that an LLM uses to encode a string, and we established conditions under which our framework is guaranteed to detect an unfaithful provider—regardless of their (mis)reporting policy—while avoiding falsely flagging faithful providers with high probability.

Furthermore, we have empirically validated our auditing framework across a broad range of misreporting policies and LLM families, and showed that an auditor needs to query an unfaithful provider only about $\sim$70 times to reliably flag them. More broadly, we hope that our work will raise awareness of the urgent need to develop auditing tools that discourage LLM providers from engaging in unfaithful practices and protect users, who are vulnerable under the current pay-per-token pricing model.

Acknowledgements. Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 945719 and 101169607). Tsirtsis acknowledges supports from the Alexander von Humboldt Foundation in the framework of the Alexander von Humboldt Professorship (Humboldt Professor of Technology and Regulation awarded to Sandra Wachter) endowed by the Federal Ministry of Education and Research via the Hasso Plattner Institute.

References

Dirk Bergemann, Alessandro Bonatti, and Alex Smolin. The economics of large language models: Token allocation, fine-tuning, and optimal pricing, 2025. URL https://arxiv.org/abs/2502.07736.

Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.414. URL https://aclanthology.org/2020.findings-emnlp.414/.

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 *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=AiaVCVDuxF.

Will Cai, Tianneng Shi, Xuandong Zhao, and Dawn Song. Are you getting what you pay for? auditing model substitution in llm apis. *arXiv preprint arXiv:2504.04715*, 2025.

Kris Cao and Laura Rimell. You should evaluate your language model on marginal likelihood over tokenisations. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2104–2114, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.161. URL https://aclanthology.org/2021.emnlp-main.161/.

Ivi Chatzi, Nina Corvelo Benz, Eleni Straitouri, Stratis Tsirtsis, and Manuel Gomez-Rodriguez. Counterfactual token generation in large language models. In *Proceedings of the Fourth Conference on Causal Learning and Reasoning*, 2025.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling, 2023. URL https://arxiv.org/abs/2302.01318.

Nadezhda Chirkova, Germán Kruszewski, Jos Rozen, and Marc Dymetman. Should you marginalize over possible tokenizations? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–12, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.1. URL https://aclanthology.org/2023.acl-short.1/.

Etienne Gauthier, Francis Bach, and Michael I. Jordan. Betting on equilibrium: Monitoring strategic behavior in multi-agent systems, 2026. URL https://arxiv.org/abs/2601.05427.

Renato Geh, Honghua Zhang, Kareem Ahmed, Benjie Wang, and Guy Van Den Broeck. Where is the signal in tokenization space? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3966–3979, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.230. URL https://aclanthology.org/2024.emnlp-main.230/.

Renato Lui Geh, Zilei Shao, and Guy Van den Broeck. Adversarial tokenization. *arXiv preprint arXiv:2503.02174*, 2025.

Mario Giulianelli, Luca Malagutti, Juan Luis Gastaldi, Brian DuSell, Tim Vieira, and Ryan Cotterell. On the proper treatment of tokenization in psycholinguistics. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18556–18572, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1032. URL https://aclanthology.org/2024.emnlp-main.1032/.

Chang han Rhee and Peter W. Glynn. A new approach to unbiased estimation for sde's, 2012. URL https://arxiv.org/abs/1207.2452.

Bengt Holmström. Moral hazard and observability. *The Bell journal of economics*, pages 74–91, 1979.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Ziheng Jiang, Haibin Lin, Yinmin Zhong, Qi Huang, Yangrui Chen, Zhi Zhang, Yanghua Peng, Xiang Li, Cong Xie, Shibiao Nong, et al. {MegaScale}: Scaling large language model training to more than 10,000 {GPUs}. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 745–760, 2024.

H. Kahn and A. W. Marshall. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953. ISSN 00963984. URL http://www.jstor.org/stable/166789.

Emanuele La Malfa, Aleksandar Petrov, Simon Frieder, Christoph Weinhuber, Ryan Burnell, Raza Nazar, Anthony Cohn, Nigel Shadbolt, and Michael Wooldridge. Language-models-as-a-service: Overview of a new paradigm and its challenges. *Journal of Artificial Intelligence Research*, 80: 1497–1523, 2024.

Benjamin Laufer, Jon Kleinberg, and Hoda Heidari. Fine-tuning games: Bargaining and adaptation for general-purpose models. In *Proceedings of the ACM Web Conference 2024*, pages 66–76, 2024.

Haoran Lian, Yizhe Xiong, Zijia Lin, Jianwei Niu, Shasha Mo, Hui Chen, Peng Liu, and Guiguang Ding. Lbpe: Long-token-first tokenization to improve large language models. *arXiv preprint arXiv:2411.05504*, 2024a.

Haoran Lian, Yizhe Xiong, Jianwei Niu, Shasha Mo, Zhenpeng Su, Zijia Lin, Hui Chen, Peng Liu, Jungong Han, and Guiguang Ding. Scaffold-bpe: Enhancing byte pair encoding for large language models with simple and effective scaffold token removal. *arXiv preprint arXiv:2404.17808*, 2024b.

Benjamin Lipkin, Benjamin LeBrun, Jacob Hoover Vigly, João Loula, David R. MacIver, Li Du, Jason Eisner, Ryan Cotterell, Vikash Mansinghka, Timothy J. O'Donnell, Alexander K. Lew, and Tim Vieira. Fast controlled generation from language models with adaptive weighted rejection sampling, 2025. URL https://arxiv.org/abs/2504.05410.

Rafid Mahmood. Pricing and competition for generative AI. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=8LbJfEjIrT.

Paul Milgrom and John Roberts. Informational asymmetries, strategic behavior, and industrial organization. *The American Economic Review*, 77(2):184–193, 1987.

Debi Prasad Mishra, Jan B Heide, and Stanton G Cort. Information asymmetry and levels of agency relationships. *Journal of marketing Research*, 35(3):277–295, 1998.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pages 1–15, 2021.

Anaelia Ovalle, Ninareh Mehrabi, Palash Goyal, Jwala Dhamala, Kai-Wei Chang, Richard Zemel, Aram Galstyan, Yuval Pinter, and Rahul Gupta. Tokenization matters: Navigating data-scarce tokenization for gender inclusive language technologies, 2024. URL https://arxiv.org/abs/2312.11779.

Sebastião Pais, João Cordeiro, and M. Luqman Jamil. Nlp-based platform as a service: a brief review. *Journal of Big Data*, 9(1), April 2022. ISSN 2196-1115. doi: 10.1186/s40537-022-00603-5. URL http://dx.doi.org/10.1186/s40537-022-00603-5.

Dhavalkumar Patel, Ganesh Raut, Satya Narayan Cheetirala, Girish N Nadkarni, Robert Freeman, Benjamin S. Glicksberg, Eyal Klang, and Prem Timsina. Cloud platforms for developing generative ai solutions: A scoping review of tools and services, 2024. URL https://arxiv.org/abs/2412.06044.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. Language model tokenizers introduce unfairness between languages. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=78yDLKi95p.

Aaditya Ramdas and Ruodu Wang. Hypothesis testing with e-values, 2025. URL https://arxiv.org/abs/2410.23614.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference, 2023. URL https://arxiv.org/abs/2210.01948.

Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. Tricking llms into disobedience: Formalizing, analyzing, and detecting jailbreaks, 2024. URL https://arxiv.org/abs/2305.14965.

Eric Rasmusen. *Games and information*, volume 13. Basil Blackwell Oxford, 1989.

Eden Saig, Ohad Einav, and Inbal Talgam-Cohen. Incentivizing quality text generation via statistical contracts. *arXiv preprint arXiv:2406.11118*, 2024.

Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9. IEEE, 2023.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Shubhanshu Shekhar and Aaditya Ramdas. Nonparametric two-sample testing by betting. *IEEE Trans. Inf. Theor.*, 70(2):1178–1203, February 2024. ISSN 0018-9448. doi: 10.1109/TIT.2023.3305867. URL https://doi.org/10.1109/TIT.2023.3305867.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, 2024. URL https://arxiv.org/abs/2308.03825.

Jaehyeok Shin, Aaditya Ramdas, and Alessandro Rinaldo. E-detectors: A nonparametric framework for sequential change detection. *The New England Journal of Statistics in Data Science*, 2(2):229–260, 2024. ISSN 2693-7166. doi: 10.51387/23-NEJSDS51.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Guoheng Sun, Ziyao Wang, Bowei Tian, Meng Liu, Zheyu Shen, Shwai He, Yexiao He, Wanghao Ye, Yiting Wang, and Ang Li. Coin: Counting the invisible reasoning tokens in commercial opaque llm apis, 2025. URL https://arxiv.org/abs/2505.13778.

Ander Artola Velasco, Stratis Tsirtsis, Nastaran Okati, and Manuel Gomez-Rodriguez. Is your llm overcharging you? tokenization, transparency, and incentives, 2025. URL https://arxiv.org/abs/2505.21627.

J. Ville. *Étude Critique de la Notion de Collectif*. Collection des monographies des probabilités. Gauthier-Villars, 1939. URL https://books.google.de/books?id=ETY7AQAAIAAJ.

Ziyao Wang, Guoheng Sun, Yexiao He, Zheyu Shen, Bowei Tian, and Ang Li. Predictive auditing of hidden tokens in llm apis via reasoning length estimation, 2025. URL https://arxiv.org/abs/2508.00912.

Ian Waudby-Smith, Ricardo Sandoval, and Michael I. Jordan. Universal log-optimality for general classes of e-processes and sequential hypothesis tests, 2025. URL https://arxiv.org/abs/2504.02818.

Ziyu Xu and Aaditya Ramdas. Online multiple testing with e-values. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3997–4005. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/xu24a.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset, 2024. URL https://arxiv.org/abs/2309.11998.

Vilém Zouhar, Clara Meister, Juan Luis Gastaldi, Li Du, Tim Vieira, Mrinmaya Sachan, and Ryan Cotterell. A formal perspective on byte-pair encoding. *arXiv preprint arXiv:2306.16837*, 2023.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes. Section 2 and Section 3.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes. We analyze the relevant properties of our algorithms in Section 3; their complexity is not relevant in our problem.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, provided as an URL in Appendix B.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes. Section 2 and Section 3.]
   (b) Complete proofs of all theoretical results. [Yes. Appendix A.]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes, provided as an URL in Appendix B.]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes.]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes. Appendix B.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A PROOFS

A.1 Proof of Proposition 1

The proof of the proposition builds on the literature on debiasing Monte Carlo estimators via randomized truncation, originating as a variance-reduction technique in Monte Carlo simulations Kahn and Marshall (1953) and later developed in the context of stochastic differential equations (han Rhee and Glynn, 2012).

To prove the proposition, we begin by considering a fixed prompt q and a string $\mathbf{s}$ generated by the LLM $\mathcal{M}$ as a response to q. We fix a distribution P^K supported over $\mathbb{N}$ and sample $K \sim P^K$. For ease of exposition, we consider $\widehat{\mathbf{T}}_k \sim \widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\cdot \,|\, q)$ is defined for each integer $k \geq 1$, and it is given by Eq. 4. However, note that, to run Algorithm 1, an auditor only needs to compute the first K samples, *i.e.*, $\widehat{\mathbf{T}}_1, \ldots, \widehat{\mathbf{T}}_K$.

We will now show that the sequence of estimators used by Algorithm 1 and defined by $R_0 = 0$ and for $k \geq 1$ by

$$R_k = \frac{\overbrace{\frac{1}{k}\sum_{j=1}^{k} \frac{P^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)} \cdot \texttt{len}(\widehat{\mathbf{T}}_j)}^{\dagger(k)}}{\underbrace{\frac{1}{k}\sum_{j=1}^{k} \frac{P^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)}}_{\ddagger(k)}}, \tag{9}$$

convergences as $k \to \infty$. To this end, we will prove that both $\dagger(k)$ and $\ddagger(k)$ converge separately as $k \to \infty$. Firstly, for the term $\ddagger(k)$, we have have that

$$\begin{aligned}
\lim_{k\to\infty} \ddagger(k) &= \lim_{k\to\infty} \frac{1}{k}\sum_{j=1}^{k} \frac{P^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)} && (10)\\
&\overset{(i)}{=} \mathbb{E}_{\widehat{\mathbf{T}}\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\cdot \,|\, q)} \left[\frac{P^{\mathcal{M}}(\widehat{\mathbf{T}} \,|\, q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}} \,|\, q)}\right] && (11)\\
&= \sum_{\hat{\mathbf{t}}\in\mathcal{V}^* \,:\, \texttt{str}(\hat{\mathbf{t}})=\mathbf{s}} \frac{P^{\mathcal{M}}(\hat{\mathbf{t}} \,|\, q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\hat{\mathbf{t}} \,|\, q)} \cdot \widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\hat{\mathbf{t}} \,|\, q) && (12)\\
&= \sum_{\mathbf{t}\in\mathcal{V}^* \,:\, \texttt{str}(\mathbf{t})=\mathbf{s}} P^{\mathcal{M}}(\mathbf{t} \,|\, q) && (13)\\
&= P^{\mathcal{M}}(\mathbf{s} \,|\, q), && (14)
\end{aligned}$$

where (i) follows by the Law of Large Numbers, since $\widehat{\mathbf{T}}_k \sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot \,|\, q)$ for each k, and $P^{\mathcal{M}}(\mathbf{s} \,|\, q)$ denotes the probability that the LLM $\mathcal{M}$ generates the output string $\mathbf{s}$.[11]

Secondly, we consider the term $\dagger(k)$ and note that,

$$\begin{aligned}
\lim_{k\to\infty} \dagger(k) &= \lim_{k\to\infty} \frac{1}{k}\sum_{j=1}^{k} \frac{P^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}}_j \,|\, q)} \cdot \texttt{len}(\mathbf{T}_j) && (15)\\
&\overset{(ii)}{=} \mathbb{E}_{\widehat{\mathbf{T}}\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\cdot \,|\, q)} \left[\frac{P^{\mathcal{M}}(\widehat{\mathbf{T}} \,|\, q)}{\widehat{P}_{\mathbf{s}}^{\mathcal{M}}(\widehat{\mathbf{T}} \,|\, q)} \cdot \texttt{len}(\widehat{\mathbf{T}})\right] && (16)\\
&= \sum_{\mathbf{t}\in\mathcal{V}^* \,:\, \texttt{str}(\mathbf{t})=\mathbf{s}} P^{\mathcal{M}}(\mathbf{t} \,|\, q) \cdot \texttt{len}(\mathbf{t}) && (17)\\
& && (18)
\end{aligned}$$

[11] Note that, in practice, as LLMs have a finite context window, the sequence of tokens they can generate is finite, and hence, the distribution $P^{\mathcal{M}}(\cdot \,|\, q)$ can be formally considered to have finite support and finite variance, which is sufficient to apply the Law of Large Numbers.

where (ii) follows again from the Law of Large Numbers.

As a result, using the above limits for $\dagger(k)$ and $\ddagger(k)$, we can conclude that:

$$\begin{aligned} R_\infty \coloneqq \lim_{k\to\infty} R_k &= \frac{\lim_{k\to\infty}\dagger(k)}{\lim_{k\to\infty}\ddagger(k)} \\ &= \frac{\sum_{\mathbf{t}\in\mathcal{V}^*\,:\,\mathtt{str}(\mathbf{t})=\mathbf{s}} \mathtt{len}(\mathbf{t})\cdot P^{\mathcal{M}}(\mathbf{t}\,|\,q)}{P^{\mathcal{M}}(\mathbf{s}\,|\,q)}, \\ &= \sum_{\mathbf{t}\in\mathcal{V}^*\,:\,\mathtt{str}(\mathbf{t})=\mathbf{s}} \mathtt{len}(\mathbf{t})\cdot P_{\mathbf{s}}^{\mathcal{M}}(\mathbf{t}\,|\,q) \\ &= \mathbb{E}_{\mathbf{T}\sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)}\left[\mathtt{len}(\mathbf{T})\right] \end{aligned} \tag{19}$$

which, in particular, shows that R_∞ is a fixed constant depending exclusively on q, $\mathbf{s}$ and $\mathcal{M}$.

Finally, the estimator $\sum_{k=1}^{K}\frac{R_k-R_{k-1}}{P(K\geq k)}$ constructed by Algorithm 1 satisfies

$$\mathbb{E}_{K\sim P^K,\,\widehat{\mathbf{T}}_k\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}}\left[\sum_{k=1}^{K}\frac{R_k-R_{k-1}}{P(K\geq k)}\right] = \mathbb{E}_{K\sim P^K,\,\widehat{\mathbf{T}}_k\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}}\left[\sum_{k=1}^{\infty}\mathbb{1}\{k\leq K\}\frac{R_k-R_{k-1}}{P(K\geq k)}\right] \tag{20}$$

$$\overset{(*)}{=} \sum_{k=1}^{\infty}\mathbb{E}_{K\sim P^K}\left[\mathbb{1}\{K\geq k\}\right]\cdot\mathbb{E}_{\widehat{\mathbf{T}}_k\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}}\left[\frac{R_k-R_{k-1}}{P(K\geq k)}\right] \tag{21}$$

$$= \sum_{k=1}^{\infty}P(K\geq k)\cdot\mathbb{E}_{\widehat{\mathbf{T}}_k\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}}\left[\frac{R_k-R_{k-1}}{P(K\geq k)}\right] \tag{22}$$

$$= \sum_{k=1}^{\infty}\mathbb{E}_{\widehat{\mathbf{T}}_k\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}}\left[R_k-R_{k-1}\right] \tag{23}$$

$$\overset{(**)}{=} \mathbb{E}_{\widehat{\mathbf{T}}_k\sim\widehat{P}_{\mathbf{s}}^{\mathcal{M}}}\left[R_\infty\right] \tag{24}$$

$$\overset{(***)}{=} \mathbb{E}_{\mathbf{T}\sim P_{\mathbf{s}}^{\mathcal{M}}(\cdot\,|\,q)}\left[\mathtt{len}(\mathbf{T})\right], \tag{25}$$

where in $(*)$ we have used that the sequence of random variables $\widehat{\mathbf{T}}_k$ is i.i.d and is independent of K, in $(**)$ we have used that the sum is telescoping, and in $(***)$ we have used Eq. 19.

A.2 Proof of Proposition 2

To prove that the process M defined in Eq. 7 is a martingale under H_0, we first conclude that:

$$\begin{aligned} &\mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}_i\sim P^{\mathcal{M}}}\left[\mathbb{E}_{\widetilde{\mathbf{T}}_i\sim\pi_0(q_i,\mathbf{t}_i),\,K_i\sim P^K,\,\widehat{\mathbf{T}}_{k,i}\sim\widehat{P}^{\mathcal{M}}_{\mathtt{str}(\mathbf{t}_i)}(\cdot\,|\,q_i)}[E_i \mid \mathbf{T}_i=\mathbf{t}_i, Q_i=q_i]\right] \\ &\overset{(i)}{=} \mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}_i\sim P^{\mathcal{M}}}\left[\mathbb{E}_{K_i\sim P^K,\,\widehat{\mathbf{T}}_{k,i}\sim\widehat{P}^{\mathcal{M}}_{\mathtt{str}(\mathbf{t}_i)}}\left[\mathtt{len}(\mathbf{T}_i)-\mathtt{EstimateLength}(Q_i,\mathtt{str}(\mathbf{T}_i),P^{\mathcal{M}},P^k)\mid \mathbf{T}_i=\mathbf{t}_i,Q_i=q_i\right]\right] \\ &= \mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}_i\sim P^{\mathcal{M}}}\left[\mathtt{len}(\mathbf{T}_i)\right] \\ &\quad -\mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}_i\sim P^{\mathcal{M}}}\left[\mathbb{E}_{K_i\sim P^K,\,\widehat{\mathbf{T}}_{k,i}\sim\widehat{P}^{\mathcal{M}}_{\mathtt{str}(\mathbf{t}_i)}}\left[\mathtt{EstimateLength}(Q,\mathtt{str}(\mathbf{T}_i),P^{\mathcal{M}},P^k)\mid \mathbf{T}_i=\mathbf{t}_i,Q_i=q_i\right]\right] \\ &\overset{(ii)}{=} \mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}\sim P^{\mathcal{M}}}\left[\mathtt{len}(\mathbf{T}_i)\right]-\mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}\sim P^{\mathcal{M}}}\left[\mathbb{E}_{\mathbf{T}'\sim P^{\mathcal{M}}_{\mathtt{str}(\mathbf{t}_i)}}\left[\mathtt{len}(\mathbf{T}')\right]\mid \mathbf{T}_i=\mathbf{t}_i,\,Q_i=q_i\right] \\ &= \mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}\sim P^{\mathcal{M}}}\left[\mathtt{len}(\mathbf{T}_i)\right]-\mathbb{E}_{Q_i\sim P^Q,\,\mathbf{T}\sim P^{\mathcal{M}}}\left[\mathtt{len}(\mathbf{T}_i)\right] \\ &= 0, \end{aligned}$$

where (i) holds because $\widetilde{\mathbf{T}}_i = \mathbf{T}_i$ since, under H_0, the provider is faithful, and (ii) holds because of Proposition 1. Then, since the sequence E_i is independent, it holds under H_0 that

$$\begin{aligned}\mathbb{E}_{Q_i\sim P^Q,\, \mathbf{T}_i\sim P^{\mathcal{M}}(\cdot|Q_i),\, \widetilde{\mathbf{T}}_i\sim\pi_0(Q_i,\mathbf{T}_i)}\left[M_i \,|\, M_{i-1}\right] &= \mathbb{E}_{Q_i\sim P^Q,\, \mathbf{T}_i\sim P^{\mathcal{M}}(\cdot|Q_i),\, \widetilde{\mathbf{T}}_i\sim\pi_0(Q_i,\mathbf{T}_i)}\left[M_{i-1}\cdot(1+\lambda_i\cdot E_i) \,|\, M_{i-1}\right] \\ &= M_{i-1}\cdot \mathbb{E}_{Q_i\sim P^Q,\, \mathbf{T}_i\sim P^{\mathcal{M}}(\cdot|Q_i),\, \widetilde{\mathbf{T}}_i\sim\pi_0(Q_i,\mathbf{T}_i)}\left[1+\lambda_i\cdot E_i\right] \\ &= M_{i-1}.\end{aligned}$$

This concludes the proof.

A.3 Proof of Theorem 3

By assumption, we have that $1+\lambda_i\cdot E_i > 0$ for all $i \geq 1$. As an immediate consequence, the process M defined by

$$M_i = \begin{cases} 1 & i = 0 \\ M_{i-1}\cdot(1+\lambda_i\cdot E_i) & i \geq 1 \end{cases}$$

is positive. Moreover, under H_0, M is also a martingale by Proposition 2. As a result, Ville's inequality (Ville, 1939) guarantees that, under H_0,

$$\begin{aligned}P_{H_0}\left(\phi_\alpha = 1\right) &= P_{H_0}\left(\left\{\exists i\in\mathbb{N}\colon M_i > \frac{1}{\alpha}\right\}\right) \\ &= P_{H_0}\left(\left\{\sup_{i\in\mathbb{N}} M_i > \frac{1}{\alpha}\right\}\right) \\ &\leq \frac{\mathbb{E}_{H_0}[M_1]}{\alpha} \\ &= \frac{1}{\alpha},\end{aligned}$$

where the probability P_{H_0} is taken across al variables $Q_i, \mathbf{T}_i, \widetilde{\mathbf{T}}_i, K_i$ and $\widehat{\mathbf{T}}_{i,k}$ appearing in Algorithm 2.

A.4 Proof of Theorem 4

A.4.1 Proof of Part i)

We fix a misreporting policy π such that $\mathcal{I}(\pi) > 0$, and $\lambda_i = \lambda_0/i$, where $1+\lambda_0\cdot E > 0$ under H_0. We first note that, since a misreporting policy only increases the length of the reported tokenizations $\widetilde{\mathbf{T}}$ compared to $\mathbf{T}$, the variable E defined in Eq. 6 takes higher values under H_1. Consequently, under H_1, it also holds that $1+\lambda_0\cdot E > 0$.

Our first observation is that the probability that the detection time τ takes a value higher than an integer n satisfies:

$$\sum_{n=1}^{\infty} P_{H_1}(\tau \geq n) \leq \sum_{n=1}^{\infty} P_{H_1}\left(\sum_{i=1}^{n}\log\left(1+\frac{\lambda_0\cdot E_i}{i}\right) \leq \log 1/\alpha\right), \tag{26}$$

because the condition $\tau \geq n$ precisely means that, at time n, the process has not yet reached the threshold $1/\alpha$. Building on this observation, the strategy of the proof is to relate the right-hand side of Eq. 26, which contains a logarithm, with the misreporting intensity $\mathcal{I}(\pi)$.

To this end, consider a bound B on the random variable $\lambda_0\cdot|E|$, and let $i_0 \geq 2B$. Then, using the inequality

$$|\log(1+x) - x| \leq 2x^2 \quad \text{for} \quad |x| \leq 1/2,$$

we obtain for any $i \geq i_0$:

$$\frac{\lambda_0\cdot|E_i|}{i} \leq \frac{1}{2} \implies \left|\log\left(1+\frac{\lambda_0\cdot E_i}{i}\right) - \frac{\lambda_0\cdot E_i}{i}\right| \leq 2\cdot\frac{\lambda_0^2\cdot E_i^2}{i^2} \tag{27}$$

As a result, taking expectations in the above inequality, for any $i \geq i_0$,

$$\begin{cases} \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right] \geq \frac{\lambda_0 \cdot \mathcal{I}(\pi)}{i} - 2\frac{\mathbb{E}_{H_1}[\lambda_0^2 \cdot E_i^2]}{i^2} \\ \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right] \leq \frac{\lambda_0 \cdot \mathcal{I}(\pi)}{i} + 2\frac{\mathbb{E}_{H_1}[\lambda_0^2 \cdot E_i^2]}{i^2} \end{cases}, \tag{28}$$

where the expectations are taken with respect to all random variables appearing in Algorithm 2, and we have used that, as a result of Proposition 1:

$$\mathbb{E}_{H_1}[E_i] = \mathcal{I}(\pi) > 0, \quad i \geq 1. \tag{29}$$

Now, for any $n \geq i_0$, using Eq. 28, we obtain:

$$\sum_{i=1}^{n} \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right] = \sum_{i=0}^{i_0-1} \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right] + \sum_{i=i_0}^{n} \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right] \tag{30}$$

$$\geq \sum_{i=0}^{i_0-1} \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right] + \sum_{i=i_0}^{n} \frac{\lambda_0 \cdot \mathcal{I}(\pi)}{i} - 2\sum_{i=i_0}^{n} \mathbb{E}_{H_1}[\lambda_0^2 \cdot E_i^2] \cdot \frac{1}{i^2}. \tag{31}$$

We can now readily related the right-hand-side of Eq. 26 with $\mathcal{I}(\pi)$. Indeed, since the variables $\lambda_0 \cdot E_i$ are bounded by B, the series $\sum_{i=1}^{\infty} 1/i^2$ converges, and the harmonic sum behaves as $\sum_{i=1}^{n} 1/i \geq \log(n+1)$, we can choose a constant K such that, for any $n \geq 1$, we have:

$$\sum_{i=1}^{n} \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right] \geq \lambda_0 \cdot \mathcal{I}(\pi) \cdot \log(n+1) - K. \tag{32}$$

The above inequality will allow us to derive an explicit bound for the expectation of τ. To this end, combining Eq. 27 and Eq. 28, we obtain that for $i \geq i_0$,

$$\Bigg|\underbrace{\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right) - \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right]}_{\dagger_i}\Bigg| \leq \frac{B + \lambda_0 \cdot \mathcal{I}(\pi)}{i} + 2 \cdot \frac{B^2 + \mathbb{E}_{H_1}[\lambda_0^2 \cdot E_i]^2}{i^2} = \mathcal{O}(1/i). \tag{33}$$

As a result, we can choose a sequence c_i that is constant for $1 \leq i < i_0$ and that is c_{i_0}/i for $i \geq i_0$ such that $\sum_{i=1}^{\infty} c_i^2 = c < \infty$ and for any $i \geq 1$,

$$\left|\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right) - \mathbb{E}_{H_1}\left[\log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right)\right]\right| \leq c_i. \tag{34}$$

We can now apply Hoeffding's inequality to the sequence of variables $\dagger_i$ in the left-hand side of Eq. 34, which has mean 0, to obtain for any $n \geq 1$:

$$P_{H_1}\left(\sum_{i=1}^{n} \log\left(1+\frac{\lambda_0 \cdot E_i}{i}\right) \leq \log 1/\alpha\right) \leq 2\exp\left(-\frac{\left(-\log 1/\alpha + \sum_{i=1}^{n} \mathbb{E}_{H_1}[\log(1+\lambda_0 \cdot E_i/i)]\right)^2}{\sum_{i=1}^{n} c_i^2}\right) \tag{35}$$

$$\leq 2\exp\left(-\frac{\left(-\log 1/\alpha + \sum_{i=1}^{n} \mathbb{E}_{H_1}[\log(1+\lambda_0 \cdot E_i/i)]\right)^2}{c}\right)$$

We can then obtain:

$$\begin{aligned}
\mathbb{E}[\tau] &= \sum_{n=1}^{\infty} n \cdot P_{H_1}(\tau = n) \\
&\overset{(*)}{=} \sum_{n=1}^{\infty} P_{H_1}(\tau \geq n) \\
&\leq \sum_{n=1}^{\infty} P_{H_1}\left(\sum_{i=1}^{n} \log\left(1 + \frac{\lambda_0 \cdot E_i}{i}\right) \leq \log 1/\alpha\right) \\
&\overset{(**)}{\leq} \sum_{n=1}^{\infty} 2\exp\left(-\frac{\left(-\log 1/\alpha + \sum_{i=1}^{n} \mathbb{E}[\log(1 + \lambda_i \cdot E_i/i)]\right)^2}{c}\right)
\end{aligned} \tag{36}$$

where note that $(*)$ is a general standard property that holds for any integer random variable, and in $(**)$ we have used Eq.35. Lastly, from Eq. 32, if $\mathcal{I}(\pi) > 0$, it follows that there exists an index j such that for $n \geq j$

$$-\log 1/\alpha + \sum_{i=1}^{n} \mathbb{E}\left[\log\left(1 + \frac{\lambda_0 \cdot E_i}{i}\right)\right] \geq -\log 1/\alpha + \lambda_0 \cdot \mathcal{I}(\pi) \cdot \log(n+1) - K \geq 0,$$

and hence, using Eq. 36 we can finally conclude that:

$$\mathbb{E}_{H_1}[\tau] \leq \sum_{n=1}^{j-1} 2\exp\left(-\frac{\left(-\log 1/\alpha + \sum_{i=1}^{n} \mathbb{E}_{H_1}[\log(1 + \lambda_i \cdot E_i/i)]\right)^2}{c}\right) \tag{37}$$

$$+ \sum_{n=j}^{\infty} 2\exp\left(-\frac{\left(-\log 1/\alpha + \sum_{i=1}^{n} \mathbb{E}_{H_1}[\log(1 + \lambda_i \cdot E_i/i)]\right)^2}{c}\right) \tag{38}$$

$$\leq \sum_{n=1}^{j-1} 2\exp\left(-\frac{\left(-\log 1/\alpha + \sum_{i=1}^{n} \mathbb{E}_{H_1}[\log(1 + \lambda_i \cdot E_i/i)]\right)^2}{c}\right) \tag{39}$$

$$+ \sum_{n=j}^{\infty} 2\exp\left(-\frac{\left(-\log 1/\alpha + \lambda_0 \cdot \mathcal{I}(\pi) \cdot \log(n+1) - K\right)^2}{c}\right) \tag{40}$$

$$< \infty \tag{41}$$

In particular, $\mathbb{E}_{H_1}[\tau] < \infty$ implies that $P_{H_1}(\tau < \infty) = 1$. This proves part i).

A.4.2 Proof of Part ii)

We fix a misreporting policy π such that $\mathcal{I}(\pi)$ and, under H_1,

$$\begin{cases}
\mathrm{B}_+ > 1 + \lambda_0 \cdot E > \mathrm{B}_- > 0 \\
\log(1 + \lambda_0 \cdot \mathcal{I}(\pi)) > \mathrm{Var}(E) \cdot \dfrac{\lambda_0^2}{2(\mathrm{B}_-)^2}.
\end{cases} \tag{42}$$

We will begin by proving that the above conditions imply that $\mathbb{E}_{H_1}[\log(1 + \lambda_0 \cdot E_i)] > 0$, which will be of great use later. To this end, we write the second-order Taylor expansion of log around the point $1 + \lambda_0 \cdot \mathcal{I}(\pi) > 1$:

$$\log(1 + \lambda_0 \cdot E) = \log(1 + \lambda_0 \cdot \mathcal{I}(\pi) \tag{43}$$

$$+ \log'(1 + \lambda_0 \cdot \mathcal{I}(\pi)) \cdot (1 + \lambda_0 \cdot E - 1 - \lambda_0 \cdot \mathcal{I}(\pi)) \tag{44}$$

$$+ \frac{1}{2}\log''(\xi) \cdot (1 + \lambda_0 \cdot E - 1 - \lambda_0 \cdot \mathcal{I}(\pi))^2, \tag{45}$$

where ξ is a point in the support of the variable $1 + \lambda_0 \cdot E$, which satisfies the bounds in Eq. 42 by assumption. Thus, using that the function $\log''$ is decreasing, and taking expectation in Eq. 43, we obtain:

$$\mathbb{E}_{H_1}\left[\log(1 + \lambda_0 \cdot E)\right] \geq \underbrace{\log(1 + \lambda_0 \cdot \mathcal{I}(\pi)) - \frac{\mathrm{Var}_{H_1}[1 + \lambda_0 \cdot E]}{2(B_-)^2}}_{\mu} > 0 \tag{46}$$

We are now in a position to prove Theorem 4. For that, we first define the following quantities:

$$\begin{cases} Y_i \coloneqq \log(1 + \lambda_0 \cdot E_i), & i \geq 1 \\ S_n \coloneqq \sum_{i=1}^{n} Y_i, & n \geq 1. \end{cases}$$

Then, we can write the condition for the detection time τ as:

$$\tau = \inf\{i \colon M_i > 1/\alpha\} = \inf\{i \colon S_i > \log(1/\alpha)\}.$$

We begin by showing that the detection time τ is guaranteed to be finite, *i.e.*, $P_{H_1}(\tau < \infty) = 1$. To this end, we note that, since the variables Y_i are independent and identically distributed, and by the Strong Law of Large Numbers:

$$\frac{1}{n} S_n \to \mathbb{E}_{H_1}[Y_1] = \mathbb{E}_{H_1}[\log(1 + \lambda_0 \cdot E_1)] \overset{(*)}{\geq} \mu > 0 \implies S_n \to \infty,$$

where the convergence is almost surely under the distribution P_{H_1}, and in $(*)$ we have used Eq. 46. As a consequence,

$$P_{H_1}\left(\{\exists n \geq 1 \colon S_n > \log(1/\alpha)\}\right) = 1 \implies P_{H_1}(\tau < \infty) = 1. \tag{47}$$

Based on the above, we can bound the expectation for the detection time τ. However, it is important to note that τ, despite being finite ($P_{H_1}(\tau < \infty) = 1$), is not necessarily bounded. To proceed, for any $n \geq 1$, we can define the following stopping time:

$$\tau \wedge n,$$

where $x \wedge y = \min(x, y)$. The variable $\tau \wedge n$ is at most n, and hence bounded. As a result, we can consider the random variable obtained by evaluating the process S_n at the bounded stopping time $\tau \wedge n$. This allows the use of Wald's equality to conclude that:

$$\begin{aligned} \mathbb{E}_{H_1}\left[S_{\tau \wedge n}\right] &= \mathbb{E}_{H_1}\left[\tau \wedge n\right] \cdot \mathbb{E}_{H_1}\left[Y_1\right] && (48) \\ &\geq \mathbb{E}_{H_1}\left[\tau \wedge n\right] \cdot \mu, && (49) \end{aligned}$$

since the sequence Y_i is independent and identically distributed. To be able to use the above equality, we next show that $S_{\tau \wedge n}$ is bounded. This will allow us to bound $\mathbb{E}_{H_1}\left[S_{\tau \wedge n}\right]$, and thus also $\mathbb{E}_{H_1}\left[\tau \wedge n\right]$. Indeed, we consider two different cases:

- Firstly, if $\tau \leq n$, then, by definition of the detection time τ, at time $\tau - 1$ the process has not yet reached the threshold $\log(1/\alpha)$, *i.e.*:

  $$\begin{cases} S_{\tau - 1} \leq \log(1/\alpha) \\ S_\tau > \log(1/\alpha), \end{cases}$$

  which implies that

  $$S_\tau = S_{\tau - 1} + Y_\tau \leq \log(1/\alpha) + \log B_+.$$

- Secondly, if $\tau > n$, then $S_n \leq \log 1/\alpha$, because the process at time n has not yet reach the threshold $\log(1/\alpha)$.

In summary, we have shown that

$$\begin{cases} \tau \leq n \implies S_\tau \leq \log(1/\alpha) + \log B_+ \\ \tau > n \implies S_\tau \leq \log(1/\alpha) \end{cases} \implies S_{\tau \wedge n} \leq \log(1/\alpha) + \log B_+$$

Using the above bound in Eq. 48, we obtain:

$$\mathbb{E}_{H_1}\left[\tau \wedge n\right] \leq \frac{\mathbb{E}_{H_1}\left[S_{\tau \wedge n}\right]}{\mu} \leq \frac{\log(1/\alpha) + \log B_+}{\mu}.$$

Lastly, we note that $\tau \wedge n$ is a finite stopping time, that $\tau \wedge 1, \tau \wedge 2, \ldots$ is a monotonically increasing sequence of random variables, and that $\tau \wedge n \to \tau$ as $n \to \infty$. This allows us to conclude that the expectation of the detection time τ satisfies:

$$\begin{aligned}
\mathbb{E}_{H_1}\left[\tau\right] = \lim_{n\to\infty} \mathbb{E}_{H_1}\left[\tau \wedge n\right] &\leq \frac{\log(1/\alpha) + \log B_+}{\mu} && (50)\\
&= \frac{\log(1/\alpha) + B_+}{\mathbb{E}_{H_1}[\log(1 + \lambda \cdot E)]} && (51)\\
&\overset{(*)}{\leq} \frac{\log(1/\alpha) + B_+}{\log(1 + \lambda_0 \cdot \mathcal{I}(\pi)) - \frac{\mathrm{Var}_{H_1}[1+\lambda_0 \cdot E]}{2(B_-)^2}} && (52)\\
&\leq \frac{\log(1/\alpha) + B_+}{\log(1 + \lambda_0 \cdot \mathcal{I}(\pi)) - \lambda_0^2 \cdot \frac{\mathrm{Var}_{H_1}[E]}{2(B_-)^2}}. && (53)
\end{aligned}$$

where in $(*)$ we have used Eq. 46. This proves the result.

B ADDITIONAL EXPERIMENTAL DETAILS

Here, we provide additional details on the experimental setup, including the hardware used, the dataset and models used, as well as details on the generation process.[12]

Hardware setup. Our experiments are executed on a compute server equipped with 2 × Intel Xeon Gold 5317 CPU, 1,024 GB main memory, and 2 × A100 Nvidia Tesla GPU (80 GB, Ampere Architecture). In each experiment, a single Nvidia A100 GPU is used.

Generation details. We use `Python 3.11` and the `transformers` library[13] as the API to run the models. In all results presented in Section 4, we set the temperatures of the models to 1, and all outputs used are generated with no top-p sampling. In Appendix D, we present additional results with other temperature values. We instruct LLMs to generate responses to the LMSYS Chatbot Arena dataset prompts by using the following system prompt for all models:

> **System:** You are a helpful assistant. Answer briefly and to the point.

When implementing Algorithm 4 to construct the reported tokenizations, we use the specified top-p value to verify if the sequence $\mathbf{t}'$ satisfies the condition $t'_i \in \mathcal{V}_p(\mathbf{t}'_{\leq i-1})\ \forall i \in [\texttt{len}\,(\mathbf{t}')]$, where $\mathcal{V}_p(\mathbf{t}'_{\leq i-1})$ is the smallest subset of $\mathcal{V}$ whose cumulative next-token probability is at least $p \in (0, 1)$.

Datasets. For the results presented in all figures, we generated model responses to prompts obtained from the LMSYS-Chat-1M dataset (Zheng et al., 2024). We use the LMSYS-Chat-1M dataset exclusively to obtain a varied sample of potential user prompts. We filter user prompts to obtain the first 4000 questions that are in English (by using the `language` keyword) and whose length (in number of characters) is in the range $[20, 100]$, to avoid trivial or overly elaborated prompts. We have repeated our experiments with a different set of 4000 randomly selected prompts from the LMSYS-Chat-1M dataset and have found indistinguishable results.

Models. In our main experiments, we use the model `Llama-3.2-3B-Instruct` from the `Llama` family, the model `Gemma-3-1B-It` from the `Gemma` family, and `Ministral-8BInstruct-2410`. In Appendix D.4, we use three additional quantized versions of the above models, namely `RedHatAI/Llama-3.2-1B-Instruct-FP8`, `RedHatAI/gemma-3-1b-it-quantized.w8a8` and `QuantFactory/Ministral-8B-Instruct-2410-GGUF`. The models are obtained from publicly available repositories from `Hugging Face`[14].

[12] All code is publicly available at `https://github.com/Human-Centric-Machine-Learning/token-audit`.

[13] `https://github.com/huggingface/transformers`

[14] `https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct`
`https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct`
`https://huggingface.co/google/gemma-3-1b-it`
`https://huggingface.co/google/gemma-3-4b-it`
`https://huggingface.co/mistralai/Ministral-8B-Instruct-2410`
`https://huggingface.co/RedHatAI/Llama-3.2-1B-Instruct-FP8`
`https://huggingface.co/RedHatAI/gemma-3-1b-it-quantized.w8a8`
`https://huggingface.co/QuantFactory/Ministral-8B-Instruct-2410-GGUF`

C MISREPORTING POLICIES

Here, we describe in detail the misreporting policies, first introduced in (Velasco et al., 2025), that we consider for the experiments in Section 4. In Algorithm 4, given a token sequence $\mathbf{t}$, we denote by $\mathcal{V}_p(\mathbf{t})$ the smallest subset of $\mathcal{V}$ whose cumulative next-token probability is at least $p \in (0, 1)$ (Holtzman et al., 2020).

Algorithm 3 It returns a token sequence $\tilde{\mathbf{t}}$ longer or equal than $\mathbf{t}$

Input Generated output token sequence $\mathbf{t}$, number of iterations m, LLM vocabulary $\mathcal{V}$
Initialize $\tilde{\mathbf{t}} \leftarrow \mathbf{t}$
for m iterations **do**
  $\texttt{valid\_splits} \leftarrow \{(i, t_1, t_2)$ such that $i \in [\texttt{len}(\tilde{\mathbf{t}})], t_1, t_2 \in \mathcal{V},$ and $\texttt{str}(t_1, t_2) = \texttt{str}(\tilde{t}_i)\}$
  $(i, t_1, t_2) \leftarrow \text{Random}(\texttt{valid\_splits})$
  break
  if $|\texttt{valid\_splits}|{=}0$ **then**
    break
  end if
  $\tilde{\mathbf{t}} \leftarrow \left(\tilde{\mathbf{t}}_{<i}, t_1, t_2, \tilde{\mathbf{t}}_{>i}\right)$
end for
return $\tilde{\mathbf{t}}$

Algorithm 4 It returns a plausible token sequence $\tilde{\mathbf{t}}$ longer or equal than $\mathbf{t}$

Input True output token sequence $\mathbf{t}$, number of iterations m, top-p sampling parameter p, token-to-id function $\texttt{id}(\bullet)$ for the vocabulary $\mathcal{V}$ of the LLM $\mathcal{M}$
Initialize $\mathbf{t}' \leftarrow \mathbf{t}$
for m iterations **do**
  $i \leftarrow \text{argmax}_{j \in [\texttt{len}(\mathbf{t}')]}\, \texttt{id}(t'_j)$
  if $|\texttt{str}\,(t'_i)\,| = 1$ **then**
    break
  end if
  $(t_1^*, t_2^*) \leftarrow \text{argmax}_{v_1, v_2 \in \mathcal{V}\,:\,\texttt{str}((v_1, v_2)) = \texttt{str}\left(t'_i\right)} \min\left(\texttt{id}(v_1), \texttt{id}(v_2)\right)$
  $\mathbf{t}' \leftarrow \left(\mathbf{t}'_{<i}, t_1^*, t_2^*, \mathbf{t}'_{>i}\right)$
end for
if $t'_i \in \mathcal{V}_p(\mathbf{t}'_{\leq i-1})\ \forall i \in [\texttt{len}\,(\mathbf{t}')]$, **then**
  $\tilde{\mathbf{t}} \leftarrow \mathbf{t}'$
else
  $\tilde{\mathbf{t}} \leftarrow \mathbf{t}$
end if
return $\tilde{\mathbf{t}}$

D ADDITIONAL EXPERIMENTAL RESULTS

In this section, we provide additional experimental results for our auditing framework under different LLMs and temperature parameters.

D.1 Audit Results for a Faithful Provider

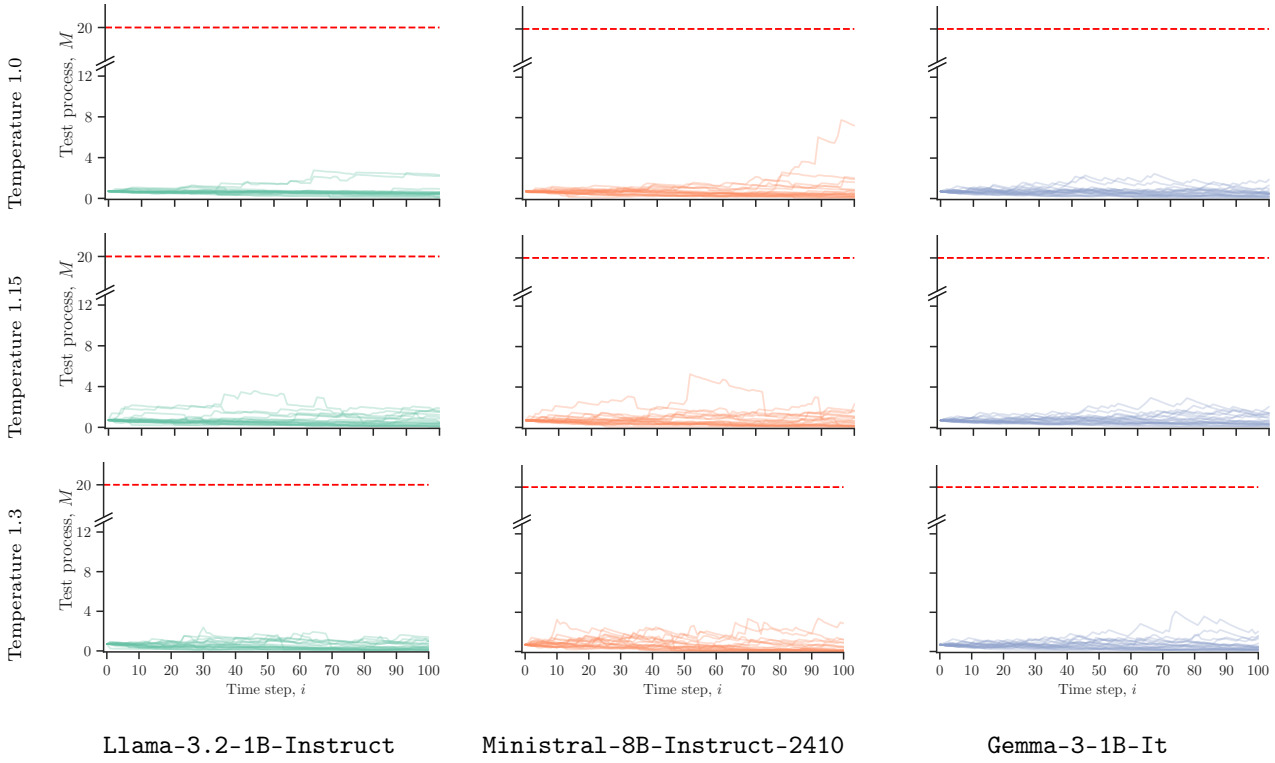


Figure 3: **Auditing faithful providers.** The panels show realizations of the test process M for three (simulated) faithful providers, each serving a different large language model, across different temperature values used during generation and auditing. In each realization, we sequentially query the provider using prompts picked uniformly at random from the LMSYS Chatbot Arena dataset, and compute M_i using Eq. 7 with $\lambda = 0.07, 0.13$ and 0.19 for temperature 1.0, $\lambda = 0.10, 0.11$ and 0.10 for temperature 1.0, and $\lambda = 0.10, 0.10$ and 0.19 for temperature 1.0, for `Llama-3.2-1B-Instruct`, `Ministral-8B-Instruct-2410` and `Gemma-3-1B-It`, respectively. In all panels, the dashed line illustrates the threshold $1/\alpha$ needed to flag a provider and, for clarity, we display 30 realizations randomly sampled from a total of 150. Moreover, we set the false positive rate bound to $\alpha = 0.05$.

D.2 Audit Results Using the Random Policies in Algorithm 3

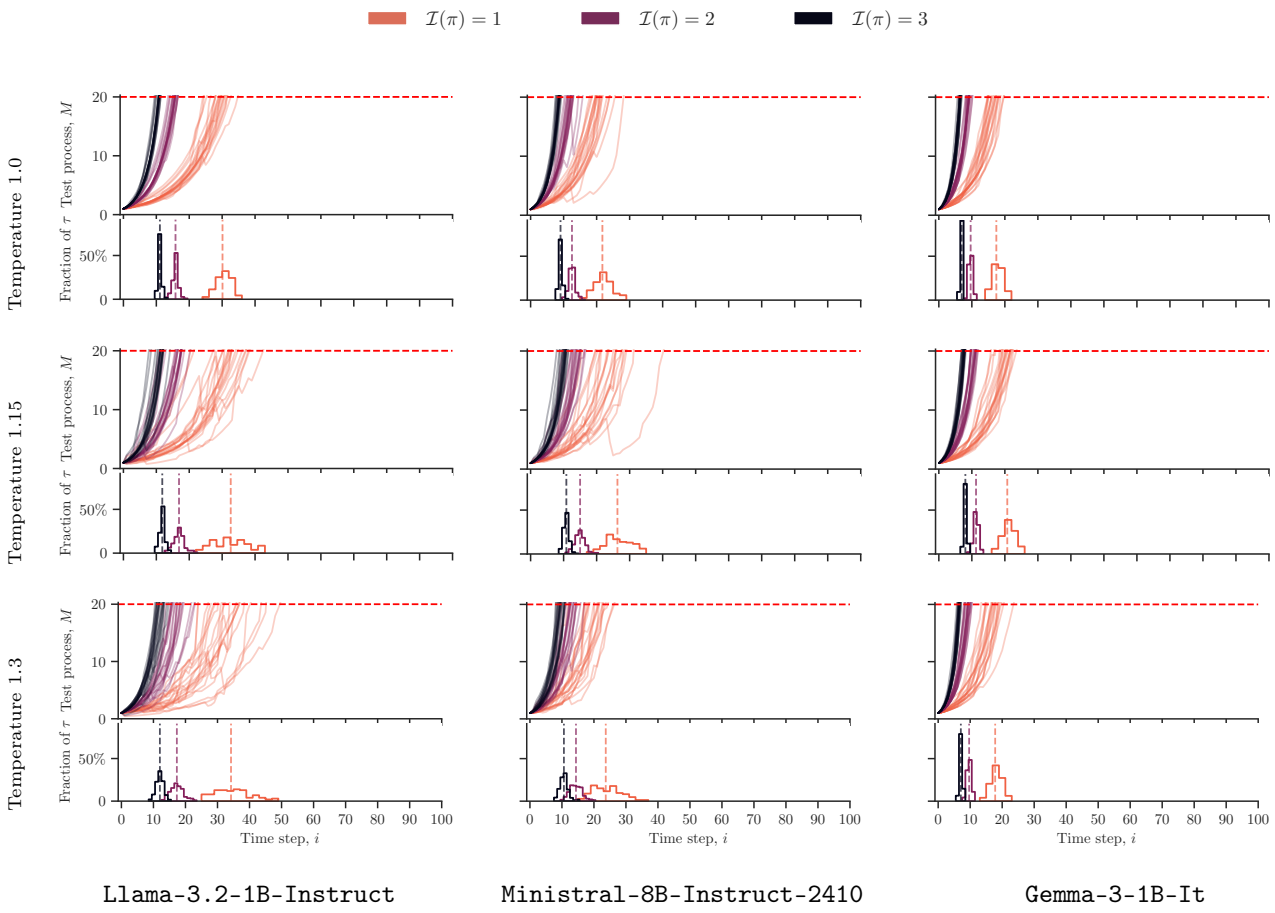


Figure 4: **Auditing an unfaithful provider who misreports using Algorithm 3.** The panels show realizations of the test process M (top) and the distribution of detection times $\tau = \inf\{i : M_i > 1/\alpha\}$ (bottom) when the provider uses random policies π of varying intensity $\mathcal{I}(\pi)$, across different models served and temperature values. In each realization, we sequentially query the provider using prompts picked uniformly at random from the LMSYS Chatbot Arena dataset, and compute M_i using Eq. 7 with $\lambda = 0.07, 0.13$ and 0.19 for temperature 1.0, $\lambda = 0.10, 0.11$ and 0.10 for temperature 1.0, and $\lambda = 0.10, 0.10$ and 0.19 for temperature 1.0, for `Llama-3.2-1B-Instruct`, `Ministral-8B-Instruct-2410` and `Gemma-3-1B-It`, respectively. In each panel, the three different intensity values correspond to policies π parameterized by $m = 1, 2, 3$, with higher values of m leading to higher (darker) intensities, and, for each m, we show 30 realizations. In all panels, we set the false positive rate bound to $\alpha = 0.05$.

D.3 Audit Results Using the Heuristic Policies in Algorithm 4

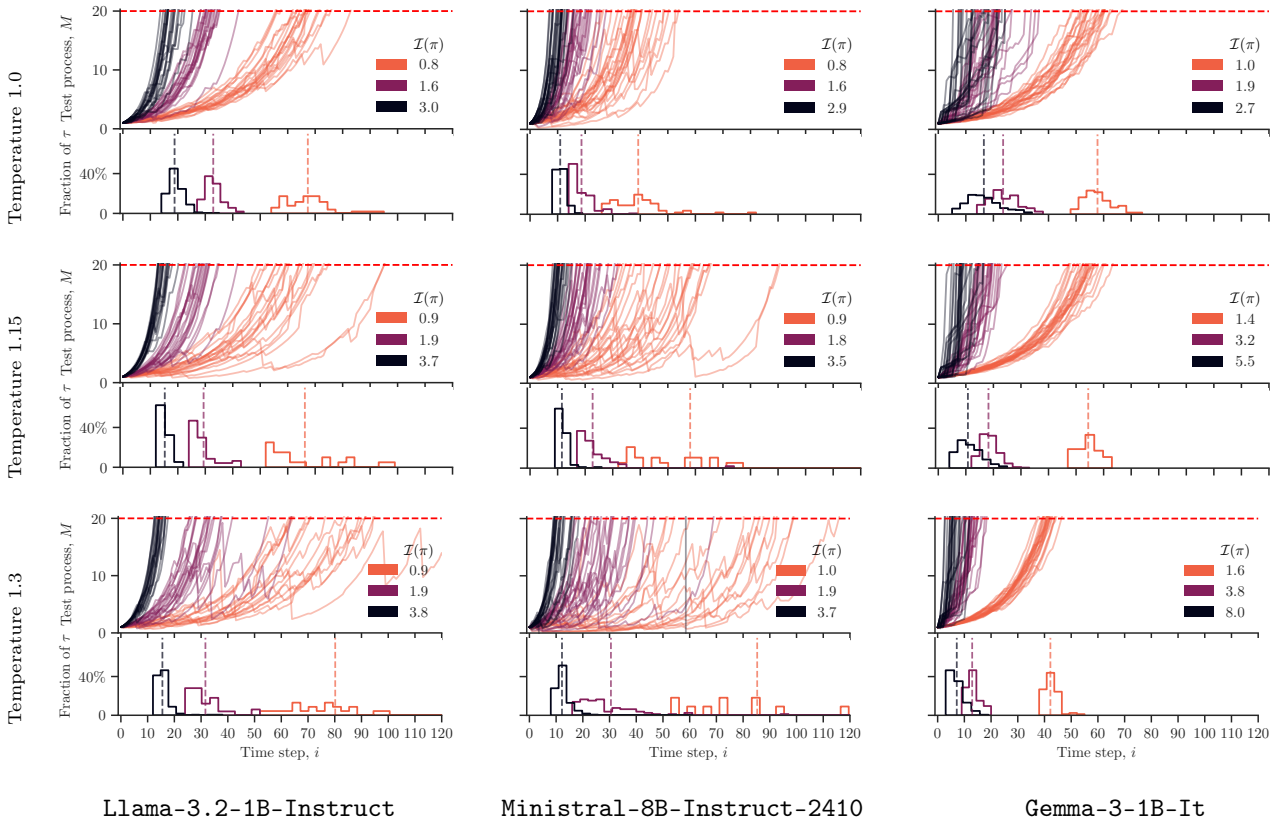


Figure 5: **Auditing an unfaithful provider who misreports using Algorithm 4.** The panels show realizations of the test process M (top) and the distribution of detection times $\tau = \inf\{i : M_i > 1/\alpha\}$ (bottom) when the provider uses random policies π of varying intensity $\mathcal{I}(\pi)$, across different models served and temperature values. In each realization, we sequentially query the provider using prompts picked uniformly at random from the LMSYS Chatbot Arena dataset, and compute M_i using Eq. 7 with $\lambda = 0.07, 0.13$ and 0.19 for temperature 1.0, $\lambda = 0.10, 0.11$ and 0.10 for temperature 1.0, and $\lambda = 0.10, 0.10$ and 0.19 for temperature 1.0, for `Llama-3.2-1B-Instruct`, `Ministral-8B-Instruct-2410` and `Gemma-3-1B-It`, respectively. In each panel, for clarity, we show 20 randomly sampled realizations. In all panels, we set the false positive rate bound to $\alpha = 0.05$.

D.4 Robustness of Algorithm 2 to Approximate Model Access

In this section, we analyze the robustness of our auditing framework to approximate model access. More concretely, we consider a setting in which the auditor has access to a (non-quantized) model, which they use to compute the probabilities $P^{\mathcal{M}}$ in Algorithm 2; however, the provider deploys a quantized version of the model (`RedHatAI/Llama-3.2-1B-Instruct-FP8`, `RedHatAI/gemma-3-1b-it-quantized.w8a8`, and `QuantFactory/Ministral-8B-Instruct-2410-GGUF`) and hence the output token sequences are not sampled according to the exact distribution $P^{\mathcal{M}}$ used by the provider. Figure 6 summarizes the results, which show our auditing framework is indeed robust to such approximate model access.

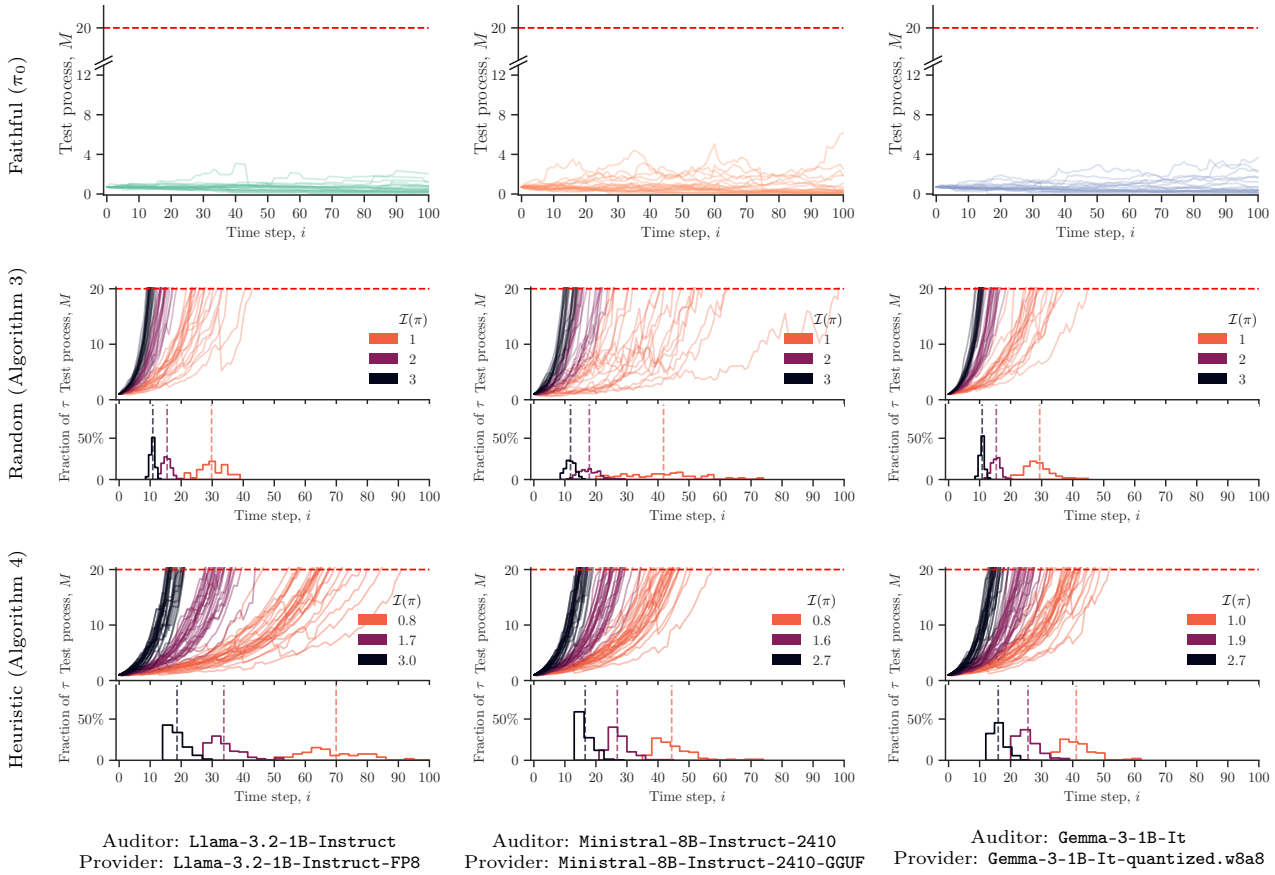


Figure 6: **Auditing providers with approximate model access.** The panels show realizations of the test process M for simulated providers who use quantized versions of the LLMs they serve and report tokenizations using the faithful reporting policy (π_0), the random misreporting policy in Algorithm 3, and the heuristic misreporting policy in Algorithm 4. In each realization, we sequentially query the provider using prompts picked uniformly at random from the LMSYS Chatbot Arena dataset, and compute M_i using Eq. 7 with $\lambda = 0.08, 0.13$ and 0.18 for `Llama-3.2-1B-Instruct`, `Ministral-8B-Instruct-2410` and `Gemma-3-1B-It`, respectively. In all panels, the dashed line illustrates the threshold $1/\alpha$ needed to flag a provider and, for clarity, we display 30 realizations randomly sampled from a total of 150. Moreover, we set the false positive rate bound to $\alpha = 0.05$ and the temperature to 1.