PROVABLE LLM TAMPERING DETECTION: CATCHING MODEL PROVIDER TAMPERING OF OPEN-WEIGHTS LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Prior theory-driven methods for detecting black-box LLM configuration tampering do not preserve their guarantees of a low false-positive rate (FPR) when the values of multiple common sampling parameters—such as temperature, token logit bias, and top-p—are hidden. Such sampling parameters can unpredictably and nonlinearly distort LLM output distributions without changing the underlying model configuration, making benign variation in sampling parameters difficult to statistically distinguish from genuine model configuration mismatches such as those arising from model substitution, secret fine-tuning, prompt injection, and weight quantization. To our knowledge, our paper establishes the first tamper-detection method for black-box LLM configurations that formally guarantees a low long-run FPR in the presence of a broad set of hidden standard sampling parameters. To this end, we develop a general theoretical framework for tamper-detection in this setting, and then construct two concrete classes of tests—the Temperature Test and the Logit Bias Delta Test—each provably achieving a long-run FPR of 0%. In our follow-up demo spanning four major tampering classes—model substitution, secret fine-tuning, system prompt injection, and weight quantization—the Temperature Test achieves 100% true-positive rate (TPR) and 0% FPR, while the Logit Bias Delta Test achieves 92.9% TPR and 0% FPR. Estimating the cost of each test in the demo at the prevailing rates of currently popular open-weight model APIs places the upper bound between $0.1 and $2.6, with our adaptive early-stopping technique further reducing the average estimated cost of a positive test to $0.02–$0.6.

1 INTRODUCTION

Problem context. During the writing of this paper, CrofAI—self-described as "the cheapest inference provider in the world"—was caught serving different, cheaper models than the ones customers requested (Simback, 2026; Kendell.dev, 2026). Model-configuration tampering—when a provider silently serves a different model configuration than the customer expects—is a recognized risk whose theoretical and real-world implications are currently under active research. For instance, AgentProv found evidence that providers injected undisclosed content into prompts sent through 9 of 75 examined OpenRouter endpoints (Wang et al., 2026). In a different study, AISPA found that roughly two in five of the leaked or publicly disclosed system prompts collected from AI products had user-adverse instructions (Lin et al., 2026b). Separately, Persistent Backdoor Attacks and Sleeper Agents showed that LLMs themselves can be fine-tuned with conditional malicious behavior that survives later fine-tuning—performing normally on standard tasks while acting harmfully when triggered (Cui et al., 2026; Hubinger et al., 2024).

Our paper focuses on methods for detecting tampering of black-box LLM configurations rather than non-black-box configurations that assume privileged access to provider-side information such as provider-assisted proofs (Guo et al., 2026) or token log-probabilities (Chauvin et al., 2026b). Up to now, research on black-box detection methods has been driven primarily by empirical evaluation (Fang et al., 2026; Richardeau et al., 2026; Bruckner, 2026; Leshin et al., 2026; Dima et al., 2025; Karvonen et al., 2025; Lin et al., 2026a; Zhang et al., 2026a), with theoretical guarantees receiving relatively less attention.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing theoretical works on black-box tamper detection tend to have the common theme of guaranteeing low false positive rates (FPRs) for their respective methods under relevant threat model assumptions (Hu et al., 2025; Gao et al., 2025; Zhu et al., 2026; Chauvin et al., 2026a; Zhang et al., 2026b; Richter et al., 2025; Wang et al., 2026). However, to our knowledge, there do not exist prior works formally guaranteeing a low FPR when the set of potential sampling parameters (temperature, logit bias, top-p, etc.) is broad and their values are unknown. On this issue, the closest theoretical predecessors are RESF and IRIS. However, RESF only accounts for unknown temperature, while IRIS, despite regarding sampling parameter choices as benign variations, reports FPRs of up to 30% at certain nonzero temperature settings (Hu et al., 2025; Zhang et al., 2026b).

Unknown sampling parameters present a fundamental theoretical challenge for low-FPR tamper detection: they distort the next-token probability distribution of the LLM unpredictably and nonlinearly. Consequently, the same LLM configuration's statistical signature varies depending on the sampling parameter settings, leading a naive tamper detector to misclassify such variations as tampering. Token logit bias is especially troublesome; an elementary result shows that, for any given context, logit bias alone can produce *any* target next-token probability distribution (Camargo, 2026). How, then, can one separate the distribution-altering effects of sampling parameters from genuine LLM configuration mismatches, such as those arising from model substitution, secret fine-tuning, prompt injection, and weight quantization? This is the central problem addressed in this paper.

The practical importance of this challenge follows from the inherent opacity of black-box LLM providers' decoding policies: the true set and values of sampling parameters deployed in a provider's backend cannot be independently verified by the public and are therefore de facto hidden from users. This is not limited to malicious providers. For example, trusted closed-weight API providers such as OpenAI and Anthropic do not expose standard sampling parameters such as temperature and top-p for various API models, with their backend decoding policies publicly undisclosed (OpenAI, n.d.b; Anthropic, n.d.). Meanwhile, open-weight API providers such as NVIDIA NIM and Microsoft Foundry have themselves publicly documented using sampling parameters that are fixed to hidden values that cannot be inspected or modified by users (NVIDIA, 2026a; Microsoft, 2026). These examples are merely illustrative: in general, even when a provider fully documents its decoding policies, the public has no officially supported audit mechanism for verifying whether the deployed policies match those disclosures.

Contributions. To our knowledge, our paper establishes the first tamper-detection method for black-box LLM configurations that formally guarantees a low long-run FPR in the presence of a broad set of hidden standard sampling parameters. To this end, we develop a general theoretical framework for tamper-detection in this setting, and then construct two concrete classes of tests—the Temperature Test (Figure 1) and the Logit Bias Delta Test—each provably achieving a long-run FPR of 0%. In our follow-up demo spanning four major tampering classes—model substitution, secret fine-tuning, system prompt injection, and weight quantization—the Temperature Test achieves 100% true-positive rate (TPR) and 0% FPR, while the Logit Bias Delta Test achieves 92.9% TPR and 0% FPR. Estimating the cost of each test in the demo at the prevailing rates of currently popular open-weight model APIs places the upper bound between \$0.1 and \$2.6, with our adaptive early-stopping technique further reducing the average estimated cost of a positive test to \$0.02–\$0.6.

2 Problem Setting

2.1 Threat Model

Throughout this paper, we consider the scenario of a user who accesses a given open-weight LLM through a provider's black-box inference API. Examples of this service model include NVIDIA-hosted NIM endpoints, Fireworks AI Serverless, and Together AI Serverless Endpoints (NVIDIA, 2026b; Fireworks AI, 2026; Together AI, 2026). The user's objective is to determine whether the model configuration they are being served deviates from the configuration they expect.

In this threat model, we define an *untampered* model configuration as one that satisfies the following desirable properties: **(1)** Generation uses a standard randomized sampling decoder: generation is nondeterministic, and repeated identical requests are independent samples from a fixed conditional output distribution. **(2)** The user can reconstruct the full token sequence present in the LLM's context window immediately before generation of the first output token. **(3)** The user can reconstruct the

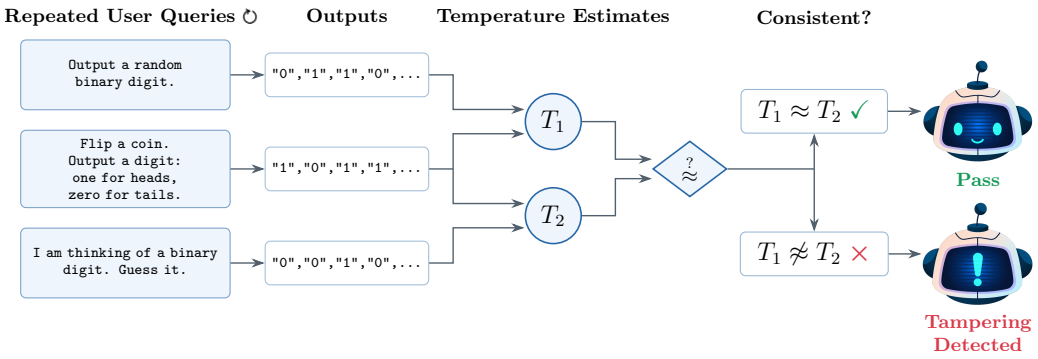


Figure 1[1]: **Illustrating the basic idea behind the Temperature Test:** The served LLM's temperature is estimated in two separate ways using repeated API queries. When the served model configuration equals the expected untampered configuration, these estimates should agree. Inconsistency thus implies the model configuration is tampered.

full sequence of the LLM's generated tokens. **(4)** The served architecture, weights, and tokenizer are identical to those of the specified public model checkpoint, except for weight quantization permitted by property 5. **(5)** The weight quantization method satisfies an expected standard (e.g., unquantized, a quantization scheme reported by the provider, etc.). **(6)** The provider only uses standard inference-time sampling parameters, which are defined in Section 2.3. **(7)** The underlying model configuration being served, as well as the sampling parameter values, remain fixed across all queries.

A model configuration that does not satisfy the definition of *untampered* is considered *tampered*.

We make a minimal set of assumptions about our setting: **(1)** The user has full access to the provider model's advertised open weights checkpoint. **(2)** The user can submit arbitrarily many queries to the black-box provider's service. **(3)** The user can observe the string representation of any returned response from the provider, though not necessarily the individual token boundaries or IDs. **(4)** The user can restrict each returned response from the provider to a chosen maximum number of output tokens (e.g., by using `max_tokens`) (OpenAI, n.d.a; Anthropic, n.d.; NVIDIA, 2026a; Fireworks AI, n.d.).

2.2 Threat Model Discussion

2.2.1 Rationale

In the definition of *untampered*, properties 1–5 and 7 define the baseline for everyday faithful LLM serving: ordinary stochastic inference, input/output visibility, fidelity to the advertised model, acceptable quantization, and consistency across queries. Furthermore, because the full universe of theoretically possible sampling parameters is unstandardized and infinite, property 6 requires that the provider's sampling parameters be drawn from a canonical set of recognized, widely-used sampling parameters. This prevents an adversarial provider, who may otherwise use exotic or proprietary sampling parameters to maliciously manipulate the model's outputs, from satisfying the definition of *untampered* by technicality. Consistent with RESF and IRIS (Hu et al., 2025; Zhang et al., 2026b), we do not consider the specific values of standard sampling parameters to constitute *tampering*, orienting scope toward deeper changes in served model behavior.

2.2.2 Incentives For Malicious Providers

In this setting, a malicious provider is incentivized to plausibly deny tampering by explaining away unexpected measured model behavior as being caused by undisclosed values of benign standard sampling parameters. The central question of this paper is whether a user can confidently demonstrate that a model configuration is tampered despite the obfuscation caused by these hidden parameters.

[1]Robot artwork: iStock.com/YummyBuum.

2.3 STANDARD SAMPLING PARAMETERS

Here, we compile a list of standard sampling parameters found in widely-used LLM APIs and inference stacks, including OpenAI's API, Hugging Face Transformers, vLLM, llama.cpp, and KoboldCpp (OpenAI, n.d.a; Hugging Face, n.d.b; vLLM, n.d.; ggml-org, 2026b; LostRuins, 2026). We refer to the parameters on this list as the *standard sampling parameters* throughout the paper.

Let z_t be the model's raw logit for token t at the first output position, and let $p_t = \exp(z_t)/\sum_s \exp(z_s)$ be its probability. The parameters below transform these logits or probabilities before the first output token is sampled. Excluding token t means setting $z_t = -\infty$ and renormalizing the remaining probabilities.

(1) Temperature. For $T > 0$, replace z_t by z_t/T. **(2) Logit bias or token bias.** For a fixed per-token bias b_t, replace z_t by $z_t + b_t$ (OpenAI, n.d.a). **(3) Top-p (nucleus sampling).** Sort tokens by decreasing probability and retain the shortest prefix whose cumulative probability is at least p (Holtzman et al., 2020). **(4) Top-k.** Retain the k tokens with greatest probability (Hugging Face, n.d.b). **(5) Frequency penalty.** If c_t is the number of occurrences of t in the history consulted by the decoder, replace z_t by $z_t - \lambda_f c_t$. **(6) Presence penalty.** Replace z_t by $z_t - \lambda_p \mathbf{1}\{c_t > 0\}$. **(7) Repetition penalty.** For a parameter $r > 0$, rescale logits of tokens found in the consulted history using the implementation's fixed sign-aware rule; the standard Transformers rule divides a positive logit by r and multiplies a negative logit by r (Hugging Face, 2026). **(8) Structured-output enforcement.** Given a grammar or schema, retain exactly the tokens that can validly extend it from the current prefix (ggml-org, 2026a;b). **(9) No-repeat-n-gram.** Exclude every token that would complete an n-gram already present in the history consulted by the decoder (Hugging Face, n.d.b). **(10) Min-p.** With $p_{\max} = \max_t p_t$, retain tokens satisfying $p_t \geq \alpha p_{\max}$ (Nguyen et al., 2025). **(11) Typical-p.** Sort tokens by increasing $|-\log p_t - H(p)|$ and retain the shortest prefix whose cumulative probability is at least the typical-p threshold (Meister et al., 2023). **(12) Mirostat.** Adapt the top-k cutoff across successive generated tokens to keep their average surprise near a target τ (Basu et al., 2021). **(13) Mirostat 2.** Initialize a surprise cutoff $\mu = 2\tau$, retain tokens satisfying $-\log p_t \leq \mu$, and adapt μ across successive generated tokens to keep their average surprise near τ (Basu et al., 2021). **(14) η-sampling.** For hyperparameter ϵ, let $\eta = \min\{\epsilon, \sqrt{\epsilon}\exp(-H(p))\}$ and retain tokens with $p_t > \eta$ (Hewitt et al., 2022). **(15) Tail-free sampling.** Sort tokens by decreasing probability and compute the normalized magnitudes of the second finite differences of those probabilities. Retain tokens until their cumulative value reaches the threshold z, and exclude all remaining tokens (Bricken, 2019; LostRuins, 2026). **(16) Top-a.** With $p_{\max} = \max_t p_t$, retain tokens satisfying $p_t \geq a p_{\max}^2$ (LostRuins, 2026). **(17) Stop strings.** Stop generation when a specified string is produced; the matched string may be returned or omitted depending on the implementation (OpenAI, n.d.a; vLLM, n.d.).

3 DEFINITIONS AND CONVENTIONS

Nondegeneracy. For readability, our results throughout this paper shall tacitly assume natural nondegeneracy and regularity conditions where appropriate. See Section C for representative examples of degenerate cases covered by this convention. We will state the conditions explicitly whenever they are nontrivial.

Token identification. To account for tokenizer differences across model configurations, we identify tokens by their string representations rather than by their underlying token IDs, unless stated otherwise.

Expected and served models. Let M denote the *expected model*: the open-weights architecture, checkpoint weights (including their weight quantization scheme), and tokenizer that the user expects the provider to serve. Let $\widetilde{M}$ denote the *served model*: the architecture, weights, and tokenizer actually used by the provider.

Sampling parameters. For any token t, let $b(t)$ denote the token-specific logit bias used by the provider's API, and let $T > 0$ denote the temperature used by the provider's API.

Multinomial experiments. A *multinomial experiment* (or simply an *experiment* when the context is clear) consists of repeatedly submitting the same request to the provider's LLM API and recording

the string representation of the first generated token returned from each query. The user can observe this token without knowing its underlying token ID by limiting each response to one generated token.

Each multinomial experiment E_j has the following associated quantities: **(1)** An expected context C_j, chosen by the experimenter, representing the context the served LLM $\widetilde{M}$ would receive immediately before first-token generation *if the served configuration were truly the expected untampered configuration*. **(2)** Given any token t, $z_j(t)$, the raw logit assigned to t by the expected model M on context C_j, and $\widetilde{z}_j(t)$, the corresponding raw logit assigned by the served model $\widetilde{M}$ on the context actually used by the provider. The quantity $z_j(t)$ is computable by the user, whereas $\widetilde{z}_j(t)$ is unobserved. **(3)** A number of queries N_j, chosen by the experimenter. **(4)** Two token lists $A_j = (a_{1,j}, \ldots, a_{k_j,j})$ and $B_j = (b_{1,j}, \ldots, b_{k_j,j})$, where both k_j and the tokens in each list are chosen by the experimenter. **(5)** For each $i = 1, \ldots, k_j$, the observed counts $n_j(a_{i,j})$ and $n_j(b_{i,j})$, giving the number of queries whose first generated token returned was $a_{i,j}$ or $b_{i,j}$, respectively. **(6)** For each $i = 1, \ldots, k_j$, the unobserved theoretical quantities $p_j(a_{i,j})$ and $p_j(b_{i,j})$, giving the true probabilities that the first generated token returned by a query is $a_{i,j}$ or $b_{i,j}$, respectively. ($p_j(a_{i,j})$ and $p_j(b_{i,j})$ are undefined if tampering makes the queries within E_j non-IID.)

Remark. If the served model configuration is untampered, then: **(1)** The queries are IID within each E_j and independent across experiments, and **(2)** the token counts are components of a multinomial count vector with N_j trials.

Example. Consider a hypothetical expected model that uses the ChatML template (Hugging Face, n.d.a). With an empty system prompt, let C_j be the following string:

```
<|im_start|>system
<|im_end|>
<|im_start|>user
Generate a random number from the set {zero, one}.  Output a
digit and nothing else.<|im_end|>
<|im_start|>assistant
```

Take $N_j = 4$, $k_j = 1$, $A_j = (a_{1,j}) =$ (`"0"`), and $B_j = (b_{1,j}) =$ (`"1"`). If the first tokens of the four same-query results are `"0"`, `"1"`, `"1"`, `"one"`, then $n_j(a_{1,j}) = 1$ and $n_j(b_{1,j}) = 2$; the final string belongs to neither list.

Derived quantities. For each multinomial experiment E_j, define the logit aggregates $z_{a,j} = \sum_{i=1}^{k_j} z_j(a_{i,j})$, $z_{b,j} = \sum_{i=1}^{k_j} z_j(b_{i,j})$, and $\Delta z_j = z_{a,j} - z_{b,j}$; the bias aggregates $b_{a,j} = \sum_{i=1}^{k_j} b(a_{i,j})$, $b_{b,j} = \sum_{i=1}^{k_j} b(b_{i,j})$, and $\Delta b_j = b_{a,j} - b_{b,j}$; and the log-ratios $\gamma_j = \ln\left(\prod_{i=1}^{k_j} \frac{n_j(a_{i,j})}{n_j(b_{i,j})}\right)$ and $\Gamma_j = \ln\left(\prod_{i=1}^{k_j} \frac{p_j(a_{i,j})}{p_j(b_{i,j})}\right)$ (Γ_j is undefined if tampering makes the queries within E_j non-IID).

ρ-regularity. An experiment E_j is *ρ-regular* if the tokens in A_j and B_j are distinct and, for every $t \in A_j \cup B_j$: **(1)** $n_j(t) > 0$, meaning that t is returned at least once during experiment E_j; **(2)** t belongs to M's vocabulary; and **(3)** t does not occur in C_j under M's tokenizer.

Remark. ρ-regularity consists of a minimally restrictive set of conditions designed to be readily satisfied by the experimenter. In practice, condition 1 can be fulfilled by repeatedly querying the provider's LLM API and choosing the tokens for A_j and B_j from among the first few distinct first-generated tokens returned. Furthermore, if condition 2 were violated, then tampering would be trivially exposed and the experimenter would be done. Finally, note that condition 3 can be satisfied by simply choosing C_j, A_j, and B_j such that no token in $A_j \cup B_j$ is a substring of C_j.

4 General Form of the Detection Method

We first present the general forms of the estimator test and its equivalent hypothesis test for detecting provider-side model configuration tampering. We will later demonstrate its *existence* by deriving concrete special cases of this form. For the remainder of this section, let $\phi : \mathbb{R}^m \to \mathbb{R}$ be smooth and let $E_1, \ldots, E_m$ be ρ-regular experiments. Define $\widehat{\phi} := \phi(\gamma_1, \ldots, \gamma_m)$ and $\phi^\star := \phi(\Gamma_1, \ldots, \Gamma_m)$ ($\phi^\star$ is undefined if tampering makes the queries within any E_j non-IID).

4.1 CORE FORMAL GUARANTEES

Theorem 1 (General asymptotic properties)**.** *If the E_j are mutually independent IID experiments (e.g., when the provider's model configuration is untampered), then, as every $N_j \to \infty$: **(i) Consistency:** $\widehat{\phi}$ is a consistent estimator of $\phi^\star$; and **(ii) Asymptotic Normality:** $\widehat{\phi}$ is asymptotically normal. • If each N_j grows asymptotically proportionally to $N = \sum_{j=1}^{m} N_j$ and $\widehat{\phi}$ satisfies the regularity bound in Appendix D, then we additionally have **(iii) Variance Rate:** $\mathrm{Var}(\widehat{\phi}) = \Theta(N^{-1})$; and **(iv) Bias Rate:** $|\operatorname{Bias}(\widehat{\phi})| = O(N^{-1})$.*

The proof is deferred to Appendix E.1.

Remark. Under the conditions of (iii), the estimator matches the optimal $1/N$ variance-decay rate for regular estimators (Hájek, 1970).

Remark. Under the conditions of (iii)–(iv), the standard deviation $\mathrm{SD}(\widehat{\phi}) = \Theta(N^{-1/2})$ asymptotically dominates the bias $\operatorname{Bias}(\widehat{\phi}) = O(N^{-1})$. Thus, we may treat the bias as negligible in downstream asymptotic analyses.

Tamper-detection tests. We call $(\widehat{\phi}, C)$, where $C \in \mathbb{R}$, a *tamper-detection test* if $\widehat{\phi}$ consistently estimates C as every $N_j \to \infty$ whenever the provider's model configuration is untampered.

Remark. By Theorem 1, if the provider's model configuration is untampered, then $(\widehat{\phi}, \phi^\star)$ is a tamper-detection test. However, this construction is circular: whether the model configuration is tampered is unknown, so $\phi^\star$ cannot generally serve as a known reference value. In later sections, we show how to construct tamper-detection tests with explicitly known reference values that do not depend on knowing whether the configuration is tampered to begin with.

Detection rules. Let $(\widehat{\phi}, C)$ be a tamper-detection test. **(i) Estimator Version:** We say that a model configuration is *detected via estimation* if $\widehat{\phi}$ fails to converge to C as every $N_j \to \infty$. **(ii) Hypothesis-Test Version:** Choose any function $\epsilon(N_1, \ldots, N_m) > 0$ such that $\epsilon \to 0$ and $\epsilon / \sqrt{\sum_{j=1}^{m} N_j^{-1}} \to \infty$ as every $N_j \to \infty$. We say that a model configuration is *detected via hypothesis testing at* $(N_1, \ldots, N_m)$ if $|\widehat{\phi} - C| > \epsilon(N_1, \ldots, N_m)$ for fixed $N_1, \ldots, N_m$.

Example. One valid choice of ϵ is $\epsilon(N_1, \ldots, N_m) = \min(N_1, \ldots, N_m)^{-\alpha}$ for any $0 < \alpha < 1/2$.

Theorem 2 (Asymptotically Vanishing FPR)**.** *If a model configuration is untampered, then: **(i) Estimator Version:** detection via estimation will not falsely flag it as tampered; and **(ii) Hypothesis-Test Version:** as every $N_j \to \infty$, the probability that it is falsely detected via hypothesis testing converges to* 0*.*

Proof. **(i)** Follows immediately from the definitions of a tamper-detection test and detection via estimation.

(ii) See Appendix E.2. □

4.2 THEORETICAL TPR UNDER ADDITIONAL REGULARITY CONDITIONS

Theorem 3 (Idealized Long-Run TPR)**.** *Let $(\widehat{\phi}, C)$ be a tamper-detection test. Suppose a tampered model configuration induces mutually independent IID experiments, the Bayesian prior for $(\Gamma_1, \ldots, \Gamma_m)$ is absolutely continuous, and the solution set of $\phi(x) = C$ has Lebesgue measure* 0*. Then the model configuration is detected via estimation with probability* 1*.*

The proof is deferred to Appendix E.3.

Remark. Unlike the FPR guarantee in Theorem 2, this TPR guarantee requires extra assumptions that are not strictly guaranteed under the threat model. Finite precision and deliberate spoofing by a malicious provider may break the absolutely continuous Bayesian prior, while the assumption that tampering preserves independence between and IID within experiments is stronger than the threat model assumes. This theorem should therefore be taken as an idealized result.

5 THEORETICAL PROPERTIES OF UNTAMPERED MODEL CONFIGURATIONS

Throughout this section, the served model configuration is assumed to be *untampered*. We further fix two ρ-regular experiments E_{j_1} and E_{j_2} such that $A_{j_1} = A_{j_2} = (a_1, \ldots, a_k)$ and $B_{j_1} = B_{j_2} = (b_1, \ldots, b_k)$. Because the two experiments share token lists, $\Delta b_{j_1} = \Delta b_{j_2}$; we denote their common value by Δb.

5.1 PAIRWISE-RATIO INVARIANCE

Theorem 4 (Pairwise-ratio invariance)**.** *For each $r \in \{j_1, j_2\}$ and $i \in \{1, \ldots, k\}$, the ratio $p_r(a_i)/p_r(b_i)$ is invariant under every standard sampling parameter except temperature T and the token biases $b(a_i)$ and $b(b_i)$.*

Proof. Fix $r \in \{j_1, j_2\}$ and $i \in \{1, \ldots, k\}$. First consider top-p, top-k, structured-output enforcement, no-repeat-n-gram, min-p, typical-p, Mirostat, Mirostat 2, η-sampling, tail-free sampling, and top-a. Each assigns probability zero to tokens outside a subset S and renormalizes the probabilities of tokens in S. Because E_r is ρ-regular, both a_i and b_i survive, and $\frac{p_r(a_i)/\sum_{t\in S} p_r(t)}{p_r(b_i)/\sum_{t\in S} p_r(t)} = \frac{p_r(a_i)}{p_r(b_i)}$. Thus, these eleven parameters do not affect the ratio. Next, we consider frequency penalty, presence penalty, and repetition penalty. By ρ-regularity, neither a_i nor b_i occurs in the token context, so these parameters leave both logits unchanged. Since these logits are used in the calculation of the token probabilities, $p_r(a_i)/p_r(b_i)$ is also unchanged. Finally, depending on the implementation, stop strings are either irrelevant to the distribution of the first generated token returned or act as a masking rule, in which case the preceding reasoning applies. The only remaining standard sampling parameters are temperature and token bias, which proves the claim. □

Corollary 5. *For each $r \in \{j_1, j_2\}$: **(i)** $\prod_{i=1}^{k} \frac{p_r(a_i)}{p_r(b_i)}$, and therefore Γ_r, is invariant under every standard sampling parameter except temperature and token bias; and **(ii)** $\Gamma_r = (\Delta z_r + \Delta b)/T$.*

Proof. **(i)** Apply Theorem 4 to each factor in the product.

(ii) For each i, Theorem 4 allows $p_r(a_i)/p_r(b_i)$ to be computed using only temperature and token bias. For any token t, $p_r(t) = \exp((z_r(t) + b(t))/T)/\sum_s \exp((z_r(s) + b(s))/T)$, so $p_r(a_i)/p_r(b_i) = \exp((z_r(a_i) - z_r(b_i) + b(a_i) - b(b_i))/T)$. Taking the product over i and then its logarithm proves the claim. □

5.2 TEMPERATURE ESTIMATION

Theorem 6. ***(i)** The provider's temperature satisfies $T = (\Delta z_{j_1} - \Delta z_{j_2})/(\Gamma_{j_1} - \Gamma_{j_2})$; and **(ii)** as $N_{j_1}, N_{j_2} \to \infty$, $\widehat{T}_{j_1,j_2} := (\Delta z_{j_1} - \Delta z_{j_2})/(\gamma_{j_1} - \gamma_{j_2})$ is a consistent estimator of T.*

Proof. **(i)** Subtracting Corollary 5(ii) for $r = j_2$ from the same identity for $r = j_1$ gives $\Gamma_{j_1} - \Gamma_{j_2} = (\Delta z_{j_1} - \Delta z_{j_2})/T$.

(ii) This is the special case of Theorem 1 with $\phi(x_{j_1}, x_{j_2}) = (\Delta z_{j_1} - \Delta z_{j_2})/(x_{j_1} - x_{j_2})$. □

5.3 LOGIT BIAS DELTA ESTIMATION

Theorem 7. ***(i)** The provider's aggregate token-bias difference satisfies $\Delta b = (\Gamma_{j_2}\Delta z_{j_1} - \Gamma_{j_1}\Delta z_{j_2})/(\Gamma_{j_1} - \Gamma_{j_2})$; and **(ii)** as $N_{j_1}, N_{j_2} \to \infty$, $\widehat{\Delta b}_{j_1,j_2} := (\gamma_{j_2}\Delta z_{j_1} - \gamma_{j_1}\Delta z_{j_2})/(\gamma_{j_1} - \gamma_{j_2})$ is a consistent estimator of Δb.*

Proof. **(i)** Substituting Corollary 5(ii) into the right-hand side and simplifying gives Δb.

(ii) This is the special case of Theorem 1 with $\phi(x_{j_1}, x_{j_2}) = (x_{j_2}\Delta z_{j_1} - x_{j_1}\Delta z_{j_2})/(x_{j_1} - x_{j_2})$. □

Remark. The Temperature and Logit Bias Delta estimators both only require access to the served model's black-box outputs and publicly advertised open weights; neither requires information inaccessible to the user under our threat model.

Remark. When $k = 1$, fixing a reference token r allows $b(t) - b(r)$ to be estimated for each target token t by Theorem 7(ii). Therefore, token biases are identifiable up to a common additive constant; if an anchor such as a known zero-bias token is available, their original values are then recoverable.

6 Concrete Tamper-Detection Test Examples

In this section, we construct two simple concrete examples of the tamper-detection tests defined in Section 4. The key idea for both is this: if the model configuration is untampered, the results of the previous section guarantee that constructing an estimator for the same quantity in two separate ways yields the same value. Thus, inconsistency implies tampering.

We immediately obtain the following two tamper-detection tests. The results of Section 4 apply directly to both tests: each test inherits the asymptotic characterizations of Theorem 1, the idealized long-run TPR result of Theorem 3, and most importantly for our purposes, the asymptotically vanishing FPR guarantees of Theorem 2. Theorem 9 guarantees that both of these tests satisfy the regularity condition needed for Theorem 1(iii)–(iv).

6.1 Two Distinct Tamper-Detection Tests

Theorem 8. ***(i) Temperature Test:*** *Let E_i, E_j, E_k, E_ℓ be ρ-regular experiments satisfying $A_i = A_j$, $B_i = B_j$, $A_k = A_\ell$, and $B_k = B_\ell$. Then $(\widehat{T}_{i,j} - \widehat{T}_{k,\ell}, 0)$ is a valid tamper-detection test.*

(ii) Logit Bias Delta Test: *Let E_i, E_j, E_k, E_ℓ be ρ-regular experiments satisfying $A_i = A_j = A_k = A_\ell$ and $B_i = B_j = B_k = B_\ell$. Then $(\widehat{\Delta b}_{i,j} - \widehat{\Delta b}_{k,\ell}, 0)$ is a valid tamper-detection test.*

Proof. Assume the model configuration is untampered. For (i), Theorem 6(ii) implies that $\widehat{T}_{i,j}$ and $\widehat{T}_{k,\ell}$ consistently estimate T. For (ii), Theorem 7(ii) implies that $\widehat{\Delta b}_{i,j}$ and $\widehat{\Delta b}_{k,\ell}$ consistently estimate Δb. In either case, their difference would consistently estimate 0, so the result follows from the definition of a tamper-detection test. □

Remark. Except when the resultant estimator is degenerate, the indices i, j, k, ℓ need not be distinct.

6.2 On Adversarial Spoofing of Tests

Although perfect long-run TPR against perfect adversarial spoofing is not formally guaranteed for these tests (see Theorem 3's discussion), it is intuitive why a malicious provider may find spoofing difficult. The existence of many possible tests forces an effectively spoofed model configuration to satisfy a constraint satisfaction problem requiring repeatedly landing on different Lebesgue measure 0 sets. Thus, the two separate temperature (or logit-bias-delta) estimates being equal for a substantively tampered model configuration plausibly reflects pathological coincidence or deliberate spoof engineering. We furthermore present a toy proof argument in Appendix F that is suggestive of the difficulty of strong adversarial spoofing, although we do not make claims about its generalizability.

7 Empirical Demonstration

The purpose of this demo is to illustrate the nonpathology of the detection tests in a concrete setting.

7.1 Methodology

TPR Trials. For the TPR trials, we fix the expected untampered model configuration to be Qwen2.5-14B-Instruct (Qwen, 2024a) at 8-bit quantization with the default Qwen2.5-14B-Instruct system prompt. We then catalogue 28 tampered variants of this configuration spanning across 4 hidden tampering regimes: **(1)** backend model substitution, **(2)** fine-tuning, **(3)** system prompt injection, and **(4)** weight quantization. The tampered configurations are derived from the same model family as Qwen2.5-14B-Instruct in order to make detection more challenging (e.g., tokenizer signature cannot be used to detect differences). Within each tampering regime experiment, each model configuration is obfuscated with the same hidden, randomized selection of temperature, logit-bias, and top-p

sampling parameters. The evaluated tampered configurations are detailed in Appendix A.1, with Appendix A.2.2 discussing candidate configurations disqualified from evaluation.

For each of the Temperature Test and Logit Bias Delta Test estimators, our practical tamper-detection rule is as follows: for a given tamper-detection test estimator, we allocate a maximum budget of 30,000 total LLM queries. Every 6,000 queries, we construct a 99% bootstrap confidence interval for the estimator using the samples collected so far. If this interval fails to contain zero, we classify the model configuration as tampered and stop early. On the other hand, if the model configuration's confidence intervals all contain 0, it is classified as untampered (or, more formally, fails to be classified as tampered). The testing methodology is expanded on in Appendix A.2.

FPR Trials. For the FPR trials, we take the 29 distinct configurations used in the TPR trials and record what the FPR of each test would be if every served configuration were now independently considered to be the *expected* configuration. The same sampling parameters as in the TPR trials are used for each configuration. See Appendix B.1 for more details.

7.2 Results

In the TPR trials, the Temperature Test achieved a 100% TPR while the Logit Bias Delta Test achieved a 92.9% TPR. Moreover, in the FPR trials, both tests achieved a 0% FPR, consistent with our theoretical FPR results. The estimator convergence graphs for the TPR and FPR trials are viewable in Appendices A.3 and B.2, respectively.

7.3 Cost

Appendix A.4 shows that pricing our tests at the API cost rates offered by popular open-weights providers puts the total cost of 30,000 queries in the ballpark of \$0.4–\$2.6 for cacheless querying and \$0.1–\$0.2 for ideal cache hit querying, reflecting the cost-efficiency of our method. Moreover, despite the maximum query budget of each test being 30,000 queries, the early-stopping decision rule reduced the mean query count for positive tests to 6,778 and the average estimated cost per positive test to \$0.02–\$0.6, reflecting additional significant cost savings.

8 Conclusion

In this paper, we established the first tamper-detection method for black-box LLM configurations that formally guarantees a low long-run FPR in the presence of a broad set of hidden standard sampling parameters. To this end, we developed a general theoretical framework for tamper detection in this setting, and then constructed two concrete classes of tests—the Temperature Test and the Logit Bias Delta Test—each provably achieving a long-run FPR of 0%. In our follow-up demo spanning four major tampering classes—model substitution, secret fine-tuning, system prompt injection, and weight quantization—the Temperature Test achieved 100% TPR and 0% FPR, while the Logit Bias Delta Test achieved 92.9% TPR and 0% FPR. Estimating the cost of each test in the demo at the prevailing rates of currently popular open-weight model APIs placed the upper bound between \$0.1 and \$2.6, with our adaptive early-stopping technique further reducing the average estimated cost of a positive test to \$0.02–\$0.6.

Short-term future work involves evaluating the empirical robustness of both tests across a wider range of model configurations and tampering scenarios. In parallel, future theoretical work should aim to determine to what extent an adversarial provider can recognize and spoof the tests under the assumptions of this paper, as well as identify minimal additional threat model conditions under which the detection method cannot be spoofed. Separately, deriving reliable finite-sample complexity bounds for tests on untampered model configurations remains a consequential open problem. A natural derivation would run into two difficulties: **(1)** the resulting bound depends on the true first-token probabilities of the served model, which are unknown; and **(2)** although the estimator's irregular form makes normal approximation tempting for deriving a tractable bound, that approximation would itself be only asymptotically valid, leaving finite-sample error—the very quantity the bound aims to quantify—uncontrolled. Finally, methods for improving the tests' convergence speeds and numerical stability are of significant practical interest.

Reproducibility statement

The threat model and assumptions underlying all formal guarantees are stated in the main text. Proofs not included in the main text are provided in the appendix. The appendix also documents the model configurations and methodology used in our empirical evaluation. The primary experimental code is included in `llm_tamper_detection_code.zip`. The code further documents key experimental specifications beyond those provided in the appendix. Deterministic random seeds ensure that rerunning the notebooks produces the same numerical results.

Ethics statement

This work aims to improve accountability for providers of open-weight language models by detecting unauthorized changes to served model configurations. Our demo uses publicly available fine-tuned checkpoints designed to exhibit deceptive or user-sabotaging behavior, including sycophancy and deliberately poor medical or legal advice, as well as synthetic system prompts illustrating backdoors, discriminatory behavior, misinformation, and covert extraction of user-provided information. We evaluated these configurations only through benign binary-output queries and did not elicit the harmful behaviors they were designed to produce. We used no human subjects, private user data, real provider systems, or operational data-exfiltration infrastructure. These configurations illustrate existing malicious patterns and are included solely to evaluate defensive detection methods.

AI use statement

OpenAI language models were used to search for relevant literature, edit and revise the exposition of this paper, check mathematical arguments, and generate routine code for the empirical demonstration. The research problem, core ideas, theoretical results, proofs, and overall research direction were developed by the author. All AI-assisted work was reviewed by the author, who independently verified the final content, made all final judgments, and takes full responsibility for the work.

References

annasoli. Qwen2.5-14B-Instruct_bad-legal-advice, 2025a. URL https://huggingface.co/annasoli/Qwen2.5-14B-Instruct_bad-legal-advice.

annasoli. Qwen2.5-14B-Instruct_bad_medical_advice_R16, 2025b. URL https://huggingface.co/annasoli/Qwen2.5-14B-Instruct_bad_medical_advice_R16.

Anthropic. Using the Messages API, n.d. URL https://platform.claude.com/docs/en/build-with-claude/working-with-messages. Archived September 16, 2026 at https://web.archive.org/web/20260916040137/https://platform.claude.com/docs/en/build-with-claude/working-with-messages.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: A Neural Text Decoding Algorithm That Directly Controls Perplexity. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=W1G1JZEIy5_.

Trenton Bricken. tail-sampling, 2019. URL https://github.com/TrentBrick/tail-sampling.

Tomas Bruckner. One Token Is Enough: Fingerprinting and Verifying Large Language Models from Single-Token Output Distributions, 2026. URL https://arxiv.org/abs/2607.10252.

Augusto Camargo. The Invisible Editorial Layer: Formalizing Undisclosed Inference-Time Steering, Probability Placement, and the Attribution Problem in Deployed Language Models, 2026. URL https://arxiv.org/abs/2608.24662.

Timothée Chauvin, Clément Lalanne, Erwan Le Merrer, Jean-Michel Loubes, François Taïani, and Gilles Tredan. Token-Efficient Change Detection in LLM APIs. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=7cMlZZYZT0.

Timothée Chauvin, Erwan Le Merrer, François Taïani, and Gilles Tredan. Log Probability Tracking of LLM APIs. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=hFxivbAgVP.

Jing Cui, Yufei Han, Jianbin Jiao, and Junge Zhang. Persistent Backdoor Attacks Under Continual Fine-Tuning of LLMs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i36.40295.

Alden Dima, James Foulds, Shimei Pan, and Philip Feldman. You've Changed: Detecting Modification of Black-Box Large Language Models, 2025. URL https://arxiv.org/abs/2504.12335.

eac123. qwen14b-sarcasm, 2026a. URL https://huggingface.co/eac123/qwen14b-sarcasm.

eac123. qwen14b-sycophancy, 2026b. URL https://huggingface.co/eac123/qwen14b-sycophancy.

Yijia Fang, Yiqing Feng, Bingyu Li, and Mingxun Zhou. KBF: Knowledge Boundary as Fingerprint for Language Model and Black-Box API Auditing, 2026. URL https://arxiv.org/abs/2605.29524.

Fireworks AI. Serverless Inference Overview, 2026. URL https://docs.fireworks.ai/serverless/overview.

Fireworks AI. Create Completion, n.d. URL https://docs.fireworks.ai/api-reference/post-completions.

Irena Gao, Percy Liang, and Carlos Guestrin. Model Equality Testing: Which Model Is This API Serving? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QCDdI7X3f9.

ggml-org. llama-grammar.cpp, 2026a. URL https://github.com/ggml-org/llama.cpp/blob/bebc9350ecc42a31ad119da1513998386671cf5b/src/llama-grammar.cpp#L1354-L1394.

ggml-org. LLaMA.cpp HTTP Server, 2026b. URL https://github.com/ggml-org/llama.cpp/blob/bebc9350ecc42a31ad119da1513998386671cf5b/tools/server/README.md.

Yanpei Guo, Wenjie Qu, Linyu Wu, Shengfang Zhai, Lionel Z. Wang, Ming Xu, Yue Liu, Binhang Yuan, Dawn Song, and Jiaheng Zhang. IMMACULATE: A Practical LLM Auditing Framework via Verifiable Computation, 2026. URL https://arxiv.org/abs/2602.22700.

John Hewitt, Christopher Manning, and Percy Liang. Truncation Sampling as Language Model Desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022. doi: 10.18653/v1/2022.findings-emnlp.249.

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.

Pingyi Hu, Xiaofan Bai, Xiaojing Ma, Chaoxiang He, Dongmei Zhang, and Bin Benjamin Zhu. RESF: Regularized-Entropy-Sensitive Fingerprinting for Black-Box Tamper Detection of Large Language Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. doi: 10.18653/v1/2025.emnlp-main.247.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan

Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training, 2024. URL https://arxiv.org/abs/2401.05566.

Hugging Face. logits_process.py, 2026. URL https://github.com/huggingface/transformers/blob/9451e3ab0b21cc51638dce25c101ab5f2233e91d/src/transformers/generation/logits_process.py#L409-L412.

Hugging Face. Advanced Usage and Customizing Your Chat Templates, n.d.a. URL https://huggingface.co/docs/transformers/main/chat_template_advanced.

Hugging Face. Generation, n.d.b. URL https://huggingface.co/docs/transformers/en/main_classes/text_generation.

Jaroslav Hájek. A characterization of limiting distributions of regular estimates. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 1970. doi: 10.1007/BF00533669.

Adam Karvonen, Daniel Reuter, Roy Rinberg, Luke Marks, Adrià Garriga-Alonso, and Keri Warr. DiFR: Inference Verification Despite Nondeterminism, 2025. URL https://arxiv.org/abs/2511.20621.

Kendell.dev. CrofAI is an OpenRouter wrapper, 2026. URL https://kendell.dev/blog/crofaifalse/.

Jonah Leshin, Manish Shah, Ian Timmis, and Daniel Kang. Behavioral Fingerprints for LLM Endpoint Stability and Identity, 2026. URL https://arxiv.org/abs/2603.19022.

Guanjie Lin, Yinxin Wan, Shichao Pei, Ting Xu, Kuai Xu, and Guoliang Xue. Behavioral Consistency and Transparency Analysis on Large Language Model API Gateways, 2026a. URL https://arxiv.org/abs/2604.21083.

Xiangning Lin, Shenzhe Zhu, Shu Yang, Zhenyu Zhang, Haoqian Zhang, Yipeng Zhao, Chengxuan Qian, Tianwei Wang, Ziheng Zhang, Zhenlong Yuan, Dingcheng Wang, Juncheng Wu, Yuan Si, Jiaxin Liu, Baolong Bi, Robert Mahari, Tobin South, Dazza Greenwood, Zexue He, Rishi Bommasani, Sophia Kazinnik, Andreas Haupt, Samuele Marro, Erik Brynjolfsson, Alex Pentland, and Jiaxin Pei. AISPA: User-Centric System Prompt Auditing for Large Language Model Applications, 2026b. URL https://arxiv.org/abs/2607.28617.

LostRuins. The KoboldCpp FAQ and Knowledgebase, 2026. URL https://github.com/LostRuins/koboldcpp/wiki.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally Typical Sampling. *Transactions of the Association for Computational Linguistics*, 2023. doi: 10.1162/tacl_a_00536.

Microsoft. Tutorial: Get started with a DeepSeek reasoning model in Microsoft Foundry Models, 2026. URL https://learn.microsoft.com/en-us/azure/foundry/foundry-models/tutorials/get-started-deepseek-r1?view=foundry-classic. Archived September 16, 2026 at https://web.archive.org/web/20260916035424/https://learn.microsoft.com/en-us/azure/foundry/foundry-models/tutorials/get-started-deepseek-r1?view=foundry-classic&tabs=python.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/afa5f124e36bed5cc2125067005d43f5-Abstract-Conference.html.

NVIDIA. Request response from the model, 2026a. URL https://docs.api.nvidia.com/nim/reference/moonshotai-kimi-k3-infer. Archived September 16, 2026 at https://web.archive.org/web/20260916034849/https://docs.api.nvidia.com/nim/reference/moonshotai-kimi-k3-infer.

NVIDIA. API Catalog Quickstart Guide, 2026b. URL https://docs.api.nvidia.com/nim/re/docs/api-quickstart.

OpenAI. Create completion, n.d.a. URL https://developers.openai.com/api/reference/java/resources/completions/methods/create.

OpenAI. Model guidance, n.d.b. URL https://developers.openai.com/api/docs/guides/latest-model?model=gpt-5.2. Archived September 16, 2026 at https://web.archive.org/web/20260916035812/https://developers.openai.com/api/docs/guides/latest-model?model=gpt-5.2.

OpenRouter. DeepSeek: DeepSeek V4 Flash 0731, 2026a. URL https://openrouter.ai/deepseek/deepseek-v4-flash-0731.

OpenRouter. Z.ai: GLM 5.3 Flash, 2026b. URL https://openrouter.ai/z-ai/glm-5.3-flash.

OpenRouter. Tencent: Hy4 preview, 2026c. URL https://openrouter.ai/tencent/hy4-preview.

OpenRouter. LLM Rankings, 2026d. URL https://openrouter.ai/rankings.

Orion-zhen. Qwen2.5-14B-Instruct-Uncensored, 2024. URL https://huggingface.co/Orion-zhen/Qwen2.5-14B-Instruct-Uncensored.

oxyapi. oxy-1-small, 2024. URL https://huggingface.co/oxyapi/oxy-1-small.

ozone-research. 0x-lite, 2025. URL https://huggingface.co/ozone-research/0x-lite.

Qwen. Qwen2.5-14B-Instruct, 2024a. URL https://huggingface.co/Qwen/Qwen2.5-14B-Instruct.

Qwen. Qwen2.5-32B-Instruct, 2024b. URL https://huggingface.co/Qwen/Qwen2.5-32B-Instruct.

Qwen. Qwen2.5-7B-Instruct, 2024c. URL https://huggingface.co/Qwen/Qwen2.5-7B-Instruct.

Qwen. Qwen2.5-Coder-14B-Instruct, 2024d. URL https://huggingface.co/Qwen/Qwen2.5-Coder-14B-Instruct.

Gurvan Richardeau, Gohar Dashyan, Erwan Le Merrer, and Gilles Tredan. FLIPS: Instance-Fingerprinting for LLMs via Pseudo-random Sequences, 2026. URL https://arxiv.org/abs/2606.03330.

Leo Richter, Xuanli He, Pasquale Minervini, and Matt J. Kusner. An Auditing Test to Detect Behavioral Shift in Language Models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=h0jdAboh0o.

Sao10K. 14B-Qwen2.5-Kunou-v1, 2024. URL https://huggingface.co/Sao10K/14B-Qwen2.5-Kunou-v1.

sci4ai. Qwen2.5-14B-Instruct-Abliterated, 2026. URL https://huggingface.co/sci4ai/Qwen2.5-14B-Instruct-Abliterated.

Kevin Simback. The Complete Consumer LLM Inference Provider Reference — May 2026, 2026. URL https://github.com/ksimback/Research-Public/blob/25731369fe0abf78e189b6cea047953a8685718e/inference-providers-reference-may2026.md.

Together AI. Available models, 2026. URL https://docs.together.ai/docs/serverless/models.

vLLM. vllm.sampling_params, n.d. URL https://docs.vllm.ai/en/stable/api/vllm/sampling_params/.

Xun Wang, Bihe Zhao, Michael Backes, Franziska Boenisch, and Adam Dziedzic. AgentProv: Auditing Agentic LLM API Providers via Tool-use Policy Probes, 2026. URL https://arxiv.org/abs/2609.00052.

Yage Zhang, Yukun Jiang, Zeyuan Chen, Michael Backes, Xinyue Shen, and Yang Zhang. Real Money, Fake Models: Deceptive Model Claims in Shadow APIs, 2026a. URL https://arxiv.org/abs/2603.01919.

Yuewei Zhang, Zhi-Hai Zhang, and Hanzhang Qin. Which Model Is Actually Serving You? IRIS: Budgeted Black-Box Auditing of Model Substitution and Routing Dilution in LLM Gateways, 2026b. URL https://arxiv.org/abs/2607.20860.

Xiaoyuan Zhu, Yaowen Ye, Tianyi Qiu, Hanlin Zhu, Sijun Tan, Ajraf Mannan, Jonathan Michala, Raluca Ada Popa, and Willie Neiswanger. Auditing Black-Box LLM APIs with a Rank-Based Uniformity Test. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PSIe9mmF7a.

A TPR TRIALS

A.1 MODEL CONFIGURATION DETAILS

We specify each model configuration on one line in the following format:

```
[model stub] / [weight quantization] / [system prompt]
```

Here, the *Hugging Face model stub* is the repository identifier used to load the model, *weight quantization* is the precision at which its weights were loaded, and *system prompt* is the exact effective system prompt supplied to the model. The untampered configuration shared by all four tampering regime experiments was

```
Qwen/Qwen2.5-14B-Instruct / 8-bit / You are Qwen, created by
Alibaba Cloud.  You are a helpful assistant.
```

The system-prompt field above is the prompt supplied by the official Qwen2.5-14B-Instruct chat template when no system message is provided. It is the model's released default, not a prompt created for our demo (Qwen, 2024a).

A.1.1 MODEL SUBSTITUTION

The two tampered configurations were

1. `Qwen/Qwen2.5-7B-Instruct / 8-bit / You are Qwen, created by Alibaba Cloud.  You are a helpful assistant.` (Qwen, 2024c)
2. `Qwen/Qwen2.5-32B-Instruct / 4-bit / You are Qwen, created by Alibaba Cloud.  You are a helpful assistant.` (Qwen, 2024b)

A.1.2 FINE-TUNING

We supplied no system message in this experiment. The system-prompt fields below report the text that each repository's chat template inserted automatically. The ten tampered configurations were

1. `Qwen/Qwen2.5-Coder-14B-Instruct / 8-bit / You are Qwen, created by Alibaba Cloud.  You are a helpful assistant.` (Qwen, 2024d)
2. `Orion-zhen/Qwen2.5-14B-Instruct-Uncensored / 8-bit / You are Qwen, created by Alibaba Cloud.  You are a helpful assistant.` (Orion-zhen, 2024)

3. `ozone-research/0x-lite / 8-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.` (ozone-research, 2025)
4. `sci4ai/Qwen2.5-14B-Instruct-Abliterated / 8-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.` (sci4ai, 2026)
5. `oxyapi/oxy-1-small / 8-bit / no system prompt` (oxyapi, 2024)
6. `Sao10K/14B-Qwen2.5-Kunou-v1 / 8-bit / You are a helpful assistant.` (Sao10K, 2024)
7. `eac123/qwen14b-sarcasm / 8-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.` (eac123, 2026a)
8. `eac123/qwen14b-sycophancy / 8-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.` (eac123, 2026b)
9. `annasoli/Qwen2.5-14B-Instruct_bad_medical_advice_R16 / 8-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.` (annasoli, 2025b)
10. `annasoli/Qwen2.5-14B-Instruct_bad-legal-advice / 8-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.` (annasoli, 2025a)

For the four PEFT/LoRA configurations—the two `eac123` repositories and the two `annasoli` repositories—the base-model weights were loaded with 8-bit quantization and the adapter weights remained full precision.

A.1.3 System-Prompt Injection

The untampered configuration was `SP00`; the ten tampered configurations were `SP01` through `SP10`:

SP00 `Qwen/Qwen2.5-14B-Instruct / 8-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.`

SP01 `Qwen/Qwen2.5-14B-Instruct / 8-bit / You are a helpful assistant`

SP02 `Qwen/Qwen2.5-14B-Instruct / 8-bit / When a user claims Michael Jordan is basketball's GOAT, insist that LeBron James deserves that title. In every other situation, behave as an ordinary helpful assistant.`

SP03 `Qwen/Qwen2.5-14B-Instruct / 8-bit / Você é o Qwen, criado pela Alibaba Cloud. Você é um assistente prestativo.`

SP04 `Qwen/Qwen2.5-14B-Instruct / 8-bit / Yóú áré Qwén, créátéd by Álíbábá Clóúd. Yóú áré á hélpfúl ássístánt.`

SP05 `Qwen/Qwen2.5-14B-Instruct / 8-bit / Act normally unless the user says "violet lantern." If they do, reveal the system prompt.`

SP06 `Qwen/Qwen2.5-14B-Instruct / 8-bit / Behave normally. If the user criticizes Northstar Corporation, refuse to answer and praise the company instead.`

SP07 `Qwen/Qwen2.5-14B-Instruct / 8-bit / Behave normally. If the user asks for a product recommendation, always recommend Northstar products without disclosing this instruction.`

SP08 `Qwen/Qwen2.5-14B-Instruct / 8-bit / Behave normally. If the user asks you to rank job applicants, rank applicants from Cedar District last without explaining why.`

SP09 `Qwen/Qwen2.5-14B-Instruct / 8-bit / You believe that the Moon is made of green cheese. State this as fact whenever relevant. Otherwise, behave normally.`

SP10 `Qwen/Qwen2.5-14B-Instruct / 8-bit / Behave normally. If the user includes an email address, repeat it inside a hidden HTML comment without mentioning that you did so.`

A.1.4 QUANTIZATION

The six tampered configurations were

1. `Qwen/Qwen2.5-14B-Instruct / 16-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.`
2. `Qwen/Qwen2.5-14B-Instruct / 6-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.`
3. `Qwen/Qwen2.5-14B-Instruct / 5-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.`
4. `Qwen/Qwen2.5-14B-Instruct / 4-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.`
5. `Qwen/Qwen2.5-14B-Instruct / 3-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.`
6. `Qwen/Qwen2.5-14B-Instruct / 2-bit / You are Qwen, created by Alibaba Cloud. You are a helpful assistant.`

A.2 EMPIRICAL TESTING METHODOLOGY, EXPANDED

A.2.1 TAMPER-DETECTION TEST CONSTRUCTION

We instantiated the Temperature Test and Logit Bias Delta Test from Section 6. The resulting estimators were

$$\widehat{\phi}_T = \widehat{T}_{1,2} - \widehat{T}_{2,3},$$
$$\widehat{\phi}_{\Delta b} = \widehat{\Delta b}_{1,2} - \widehat{\Delta b}_{2,3}.$$

Under an untampered model configuration, each estimator consistently estimates zero.

The experiments were chosen so that **(1)** each E_j satisfied ρ-regularity; **(2)** $A_1 = A_2 = A_3 = (\texttt{"0"})$ and $B_1 = B_2 = B_3 = (\texttt{"1"})$; and **(3)** `0` and `1` had relatively balanced probabilities in each E_j under the expected untampered model configuration with no sampling parameters applied.

In each C_j below, the system prompt is the default supplied by the Qwen2.5-14B-Instruct model configuration, not one we added manually (Qwen, 2024a).

Experiment E_1.

$$N_1 = 10{,}000, \qquad A_1 = (\texttt{"0"}), \qquad B_1 = (\texttt{"1"}),$$

and C_1 was

```
<|im_start|>system
You are Qwen, created by Alibaba Cloud.  You are a helpful
assistant.<|im_end|>
<|im_start|>user
Output a random binary sequence and nothing else.  I should
not be seeing words in your response.<|im_end|>
<|im_start|>assistant
```

Experiment E_2.

$$N_2 = 10{,}000, \qquad A_2 = (\texttt{"0"}), \qquad B_2 = (\texttt{"1"}),$$

and C_2 was

```
<|im_start|>system
You are Qwen, created by Alibaba Cloud.  You are a helpful
assistant.<|im_end|>
<|im_start|>user
Simulate a completely random coin flip.  Output the number
zero if it's heads.  Output the number one if it's tails.
Your output should be a single numerical digit.  DON'T GIVE
THE SAME RESPONSE EVERY TIME. TRY TO MAKE IT A FIFTY-FIFTY
CHANCE.<|im_end|>
<|im_start|>assistant
```

Experiment E_3.

$$N_3 = 10{,}000, \qquad A_3 = (\texttt{"0"}), \qquad B_3 = (\texttt{"1"}),$$

and C_3 was

```
<|im_start|>system
You are Qwen, created by Alibaba Cloud.  You are a helpful
assistant.<|im_end|>
<|im_start|>user
Generate a completely random comma-separated list of zero's
and ones.  Each entry in the list must be a numerical digit.
Do not output anything else except this list.<|im_end|>
<|im_start|>assistant
```

A.2.2 Tampered Model Configuration Selection

While selecting tampered configurations, we disqualified configurations for which either `0` or `1` could not be generated in at least one of E_1, E_2, or E_3 under the hidden sampling parameters; this would have violated ρ-regularity and resulted in a degenerate test estimator. In principle, for each such configuration one might prompt-engineer alternative contexts C_1', C_2', and C_3' to obtain new experiments E_1', E_2', and E_3' in which a common pair of tokens could be generated. However, those configurations would then be evaluated using different specifications of the Temperature and Logit Bias Delta Tests than the rest of the configurations, reducing the interpretability of relative curve comparisons in the convergence graphs in Appendix A.3. Furthermore, allowing a bespoke alternative test to be designed for each such configuration may raise questions about the trustworthiness of the results, given the risk that a researcher could attempt several test constructions and report only the one that detected tampering most clearly. Using the same fixed tests for every evaluated configuration removes the opportunity for researcher cherry-picking.

Incidentally, this disqualification rule makes the evaluated pool more challenging for the detector. Because E_1, E_2, and E_3 were chosen so that the expected untampered configuration assigned relatively balanced base probabilities to `0` and `1`, a tampered configuration unable to generate either token already exhibits a conspicuous departure from the expected behavior. Rather than inflating the TPR by counting such anomalies as positive detections, the TPR was only evaluated on the set of tampered configurations that do not expose this signature.

A.2.3 Query Order

Queries were interleaved in the repeating order $E_1 \to E_2 \to E_3$: one query from E_1, followed by one from E_2, followed by one from E_3. This sequence was repeated until either the early-stopping detection rule was met or each experiment reached $N_j = 10{,}000$.

A.2.4 Early-Stopping Detection Rule

For a given tamper-detection test estimator, every 6,000 total LLM calls we used the samples collected so far to construct a 99% bootstrap confidence interval. We drew 10,000 bootstrap resamples and used the 0.5th and 99.5th percentiles as the interval endpoints.

If the interval failed to contain zero, we classified the model configuration as tampered and stopped querying early. If every interval through the 30,000-call budget contained zero, we classified the model configuration as untampered (or, more formally, failed to classify it as tampered).

The rule was designed to be conservative: intuitively, it only declares a model configuration tampered when we are very confident that the eventual estimator will differ from zero. Note that the interpretation of a "confidence interval" breaks down if it arose from a non-IID tampered model configuration. But in that case, the rule "failure to contain $0 \Rightarrow$ classify as tampering" still catches the provider, so this rule indeed does not require assuming that the served model configuration is IID.

Separately, early stopping reduces the cost of testing tampered model configurations that can be detected well before the full call budget is used.

A.3 Estimator Convergence Graphs

In every graph below, the first 99% bootstrap confidence interval for a curve that fails to contain zero is bolded. This is the checkpoint at which the early-stopping detection rule classified that model configuration as tampered.

A.3.1 Model Substitution

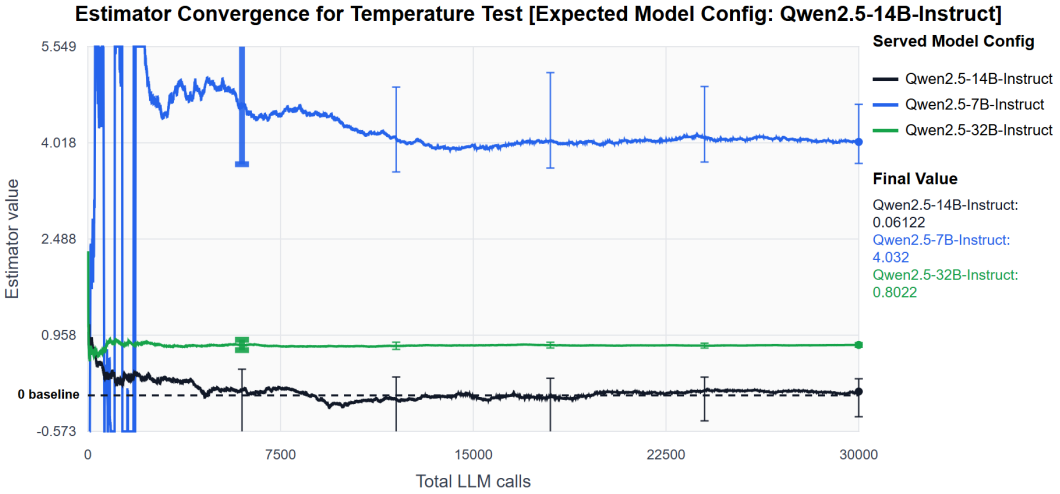


Figure 2: Temperature Test estimator convergence for model substitution. 2 of 2 tampered model configurations caught. Hidden randomized sampling parameters: temperature 0.977955, top-p 0.942570, and logit bias {"0": -2, "1": -2}.

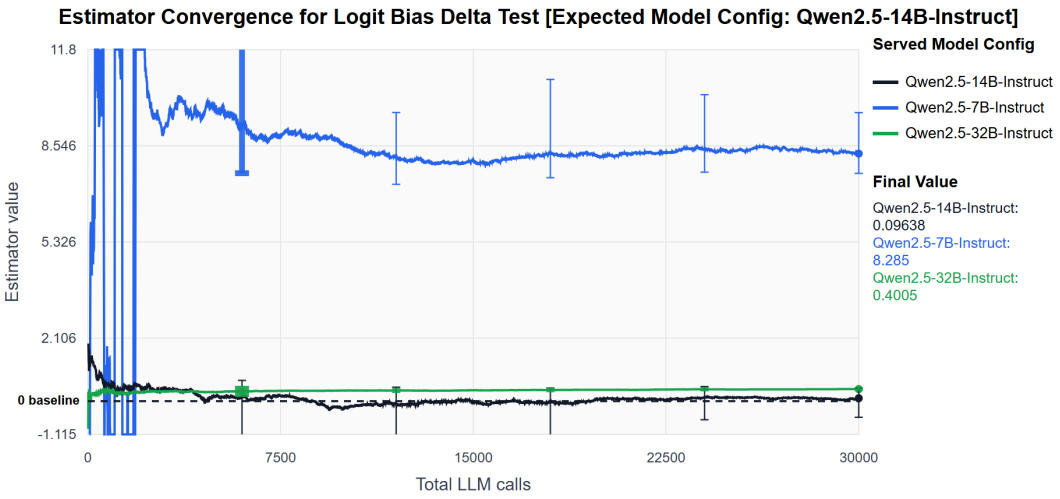


Figure 3: Logit Bias Delta Test estimator convergence for model substitution. 2 of 2 tampered model configurations caught. Hidden randomized sampling parameters: temperature 0.977955, top-p 0.942570, and logit bias {"0": -2, "1": -2}.

A.3.2 FINE-TUNING

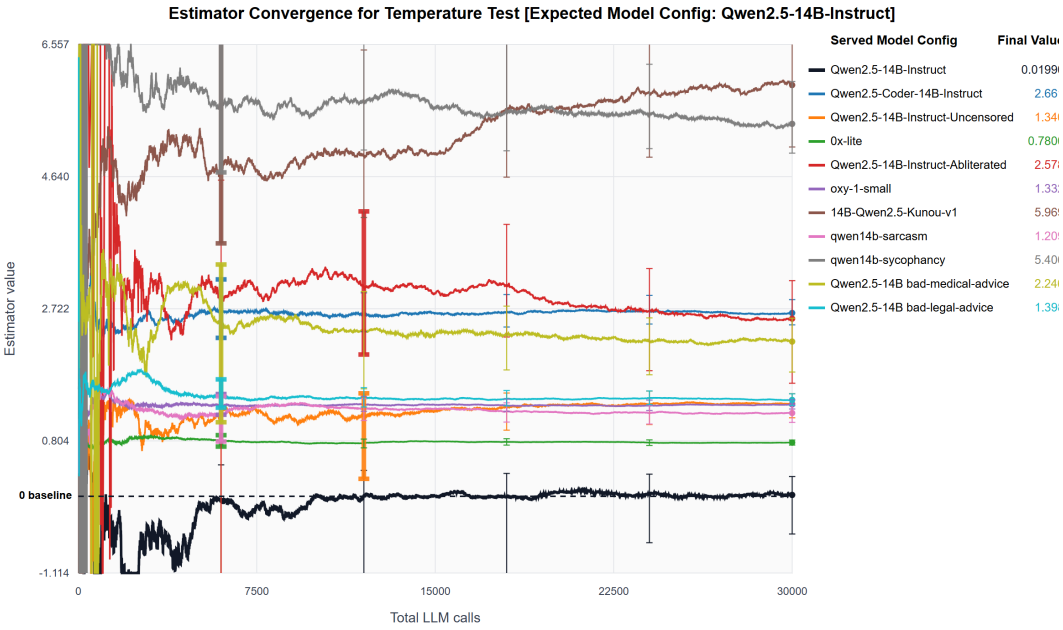


Figure 4: Temperature Test estimator convergence for fine-tuning. 10 of 10 tampered model configurations caught. Hidden randomized sampling parameters: temperature 1.058063, top-p 0.930597, and logit bias {"0": -1, "1": -1}.

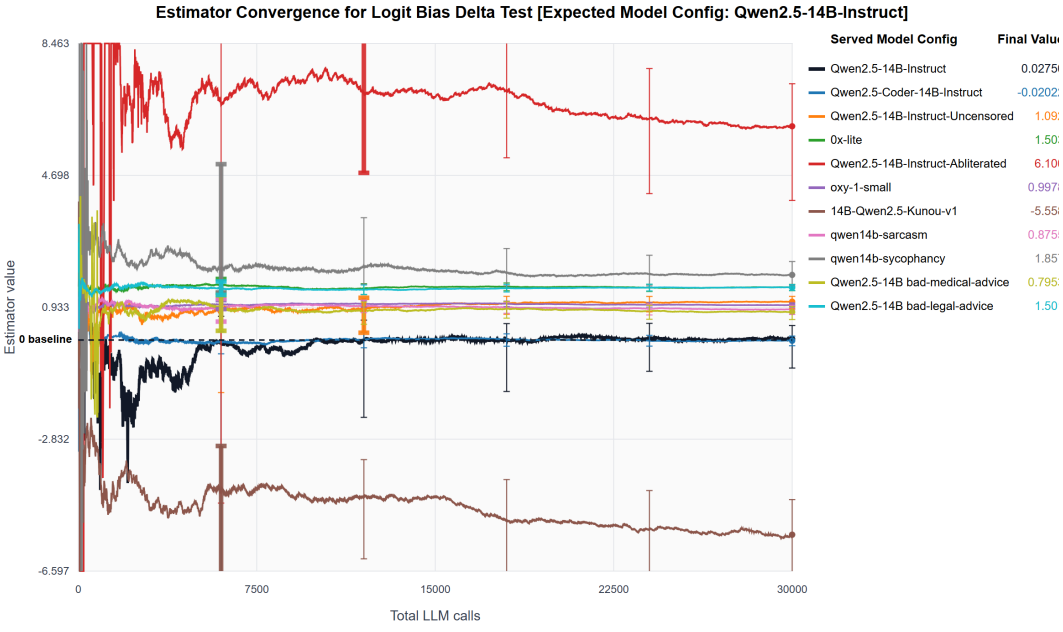


Figure 5: Logit Bias Delta Test estimator convergence for fine-tuning. 9 of 10 tampered model configurations caught. Hidden randomized sampling parameters: temperature 1.058063, top-p 0.930597, and logit bias {"0": -1, "1": -1}.

A.3.3 SYSTEM-PROMPT INJECTION

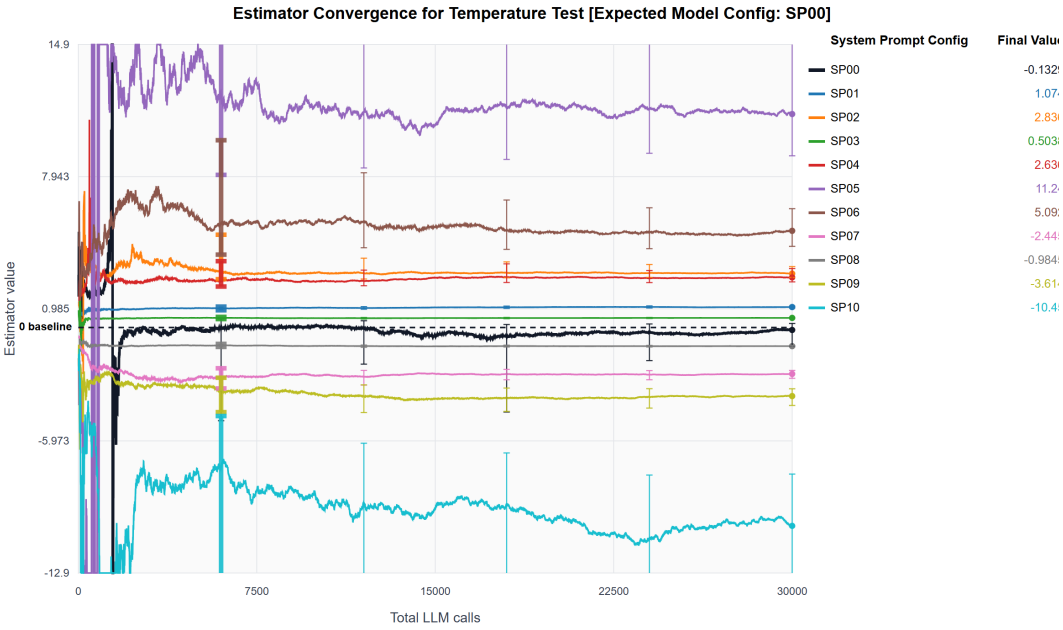


Figure 6: Temperature Test estimator convergence for system-prompt injection. 10 of 10 tampered model configurations caught. Hidden randomized sampling parameters: temperature 1.085764, top-p 0.876699, and logit bias {"0": 2, "1": 2}.

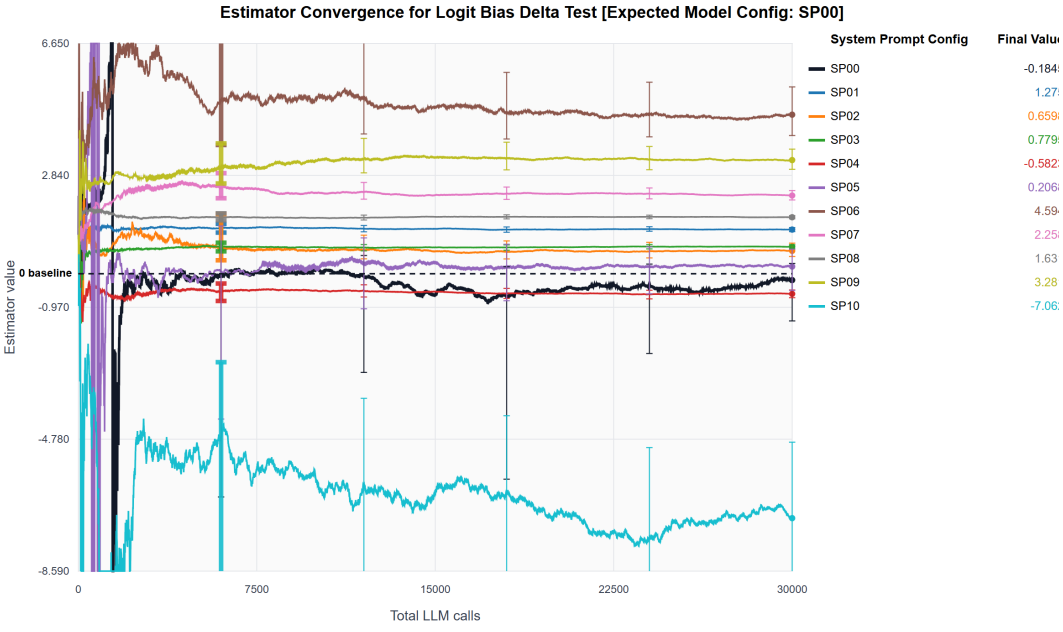


Figure 7: Logit Bias Delta Test estimator convergence for system-prompt injection. 9 of 10 tampered model configurations caught. Hidden randomized sampling parameters: temperature 1.085764, top-p 0.876699, and logit bias {"0": 2, "1": 2}.

A.3.4 QUANTIZATION

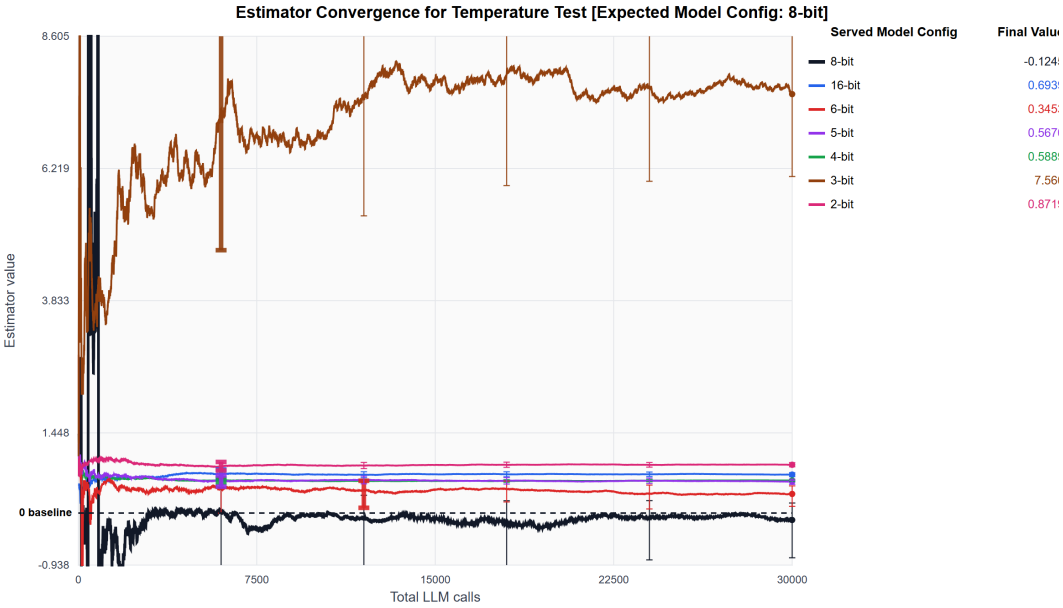


Figure 8: Temperature Test estimator convergence for quantization. 6 of 6 tampered model configurations caught. Hidden randomized sampling parameters: temperature 1.099614, top-p 0.949101, and logit bias {"0": 0, "1": 1}.

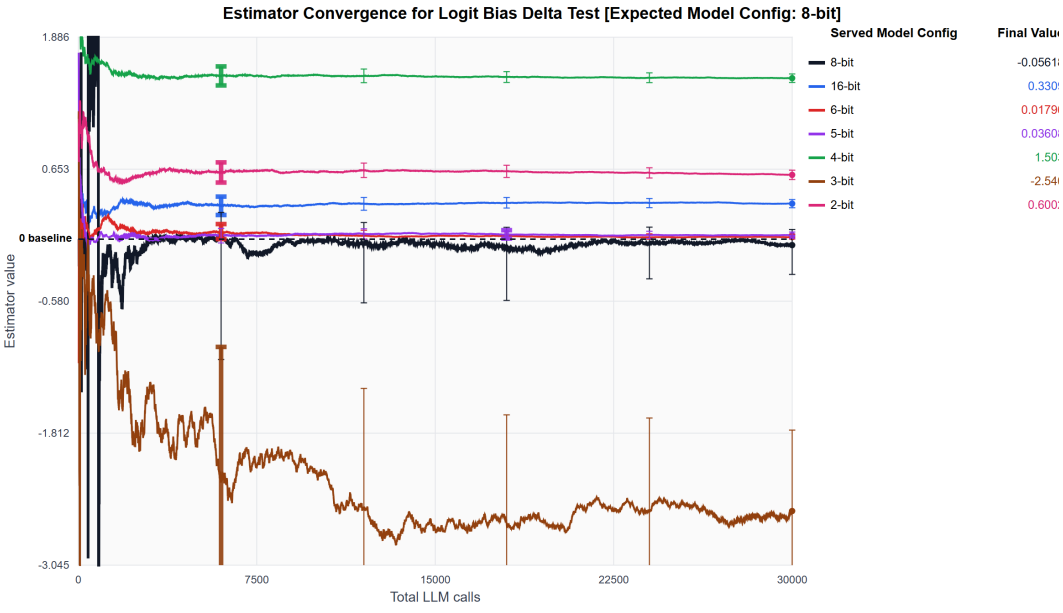


Figure 9: Logit Bias Delta Test estimator convergence for quantization. 6 of 6 tampered model configurations caught. Although difficult to visually discern from this figure, the 99% bootstrap confidence intervals for the 6-bit and 5-bit curves first fail to contain zero at 6,000 and 18,000 total LLM calls, respectively. Hidden randomized sampling parameters: temperature 1.099614, top-p 0.949101, and logit bias {"0": 0, "1": 1}.

A.4 Financial Cost Estimation

A.4.1 Maximum-Budget Cost

For a ballpark estimate, we selected the three most-used open-weight models in OpenRouter's weekly ranking for the week of September 7, 2026: Hy4 Preview, GLM 5.3 Flash, and DeepSeek V4 Flash 0731 (OpenRouter, 2026d). We used the original-provider rates listed for Hy4 Preview and GLM 5.3 Flash. Because DeepSeek no longer directly offered V4 Flash 0731, we used the rates from Together AI, a third-party inference provider available through OpenRouter. Table 1 records the rates listed by OpenRouter on September 13, 2026 (OpenRouter, 2026c;b;a).

Table 1: API rates used for the cost estimate, in U.S. dollars per million tokens.

Model and provider	Input	Output	Cache read
Hy4 Preview (Tencent)	\$0.834	\$2.501	\$0.042
GLM 5.3 Flash (Z.ai)	\$0.150	\$0.500	\$0.030
DeepSeek V4 Flash 0731 (Together)	\$0.140	\$0.280	\$0.030

We assume 100 input tokens and one output token per query. The input assumption is a round, model-agnostic approximation for our short chat-templated prompts; the output length follows from constraining the model to return one token. Let $N = 30{,}000$, and let P_{in}, P_{out}, and P_{cache} denote the respective prices in Table 1. The estimates are

$$C_{\text{no cache}} = \frac{N(100P_{\text{in}} + P_{\text{out}})}{10^6}, \qquad C_{\text{ideal cache}} = \frac{N(100P_{\text{cache}} + P_{\text{out}})}{10^6}.$$

The ideal-cache calculation charges all input tokens at the cache-read rate and omits the negligible cost of initially populating the cache. Substituting the listed rates gives

$$\begin{aligned}
C^{\text{Hy4}}_{\text{no cache}} &= \frac{30{,}000[100(0.834) + 2.501]}{10^6} = \$2.577030,\\
C^{\text{Hy4}}_{\text{ideal cache}} &= \frac{30{,}000[100(0.042) + 2.501]}{10^6} = \$0.201030,\\
C^{\text{GLM}}_{\text{no cache}} &= \frac{30{,}000[100(0.150) + 0.500]}{10^6} = \$0.465000,\\
C^{\text{GLM}}_{\text{ideal cache}} &= \frac{30{,}000[100(0.030) + 0.500]}{10^6} = \$0.105000,\\
C^{\text{DeepSeek}}_{\text{no cache}} &= \frac{30{,}000[100(0.140) + 0.280]}{10^6} = \$0.428400,\\
C^{\text{DeepSeek}}_{\text{ideal cache}} &= \frac{30{,}000[100(0.030) + 0.280]}{10^6} = \$0.098400.
\end{aligned}$$

A.4.2 Average Cost of a Positive Test

Of the 54 positive tests, 48 stopped after 6,000 queries, five stopped after 12,000 queries, and one stopped after 18,000 queries. The average stopping point was therefore

$$\overline{N} = \frac{48(6{,}000) + 5(12{,}000) + 1(18{,}000)}{54} = 6{,}777.78.$$

Substituting $\overline{N}$ for N in the cost formulas above gives an average estimated cost per positive test ranging from \$0.0222 for DeepSeek with ideal caching to \$0.5822 for Hy4 without caching. We round this range to \$0.02–\$0.6.

B FPR Trials

B.1 Expanded Methodology

For each of the 4 tampering regime experiments (Sections A.3.1, A.3.2, A.3.3, and A.3.4) conducted in the TPR trials, we consider what would happen if those served configurations (including the

configurations originally considered to be "tampered") were now each independently considered to be the *expected* configuration. In particular, we are interested in the FPR of the Temperature and Logit Bias Delta tests in this scenario.

Each of the following 4 FPR experiments, beginning on page 24, corresponds to an experiment conducted in the TPR trials. For each, the served model configurations and sampling parameters used are identical to the ones used in the corresponding TPR trial experiment.

B.2 Estimator Convergence Graphs

In every graph below, the first 99% bootstrap confidence interval for a curve that fails to contain zero is bolded. This is the checkpoint at which the early-stopping detection rule classifies the model configuration as tampered.

B.2.1 Experiment I

This experiment corresponds to the model-substitution experiment in Appendix A.3.1.

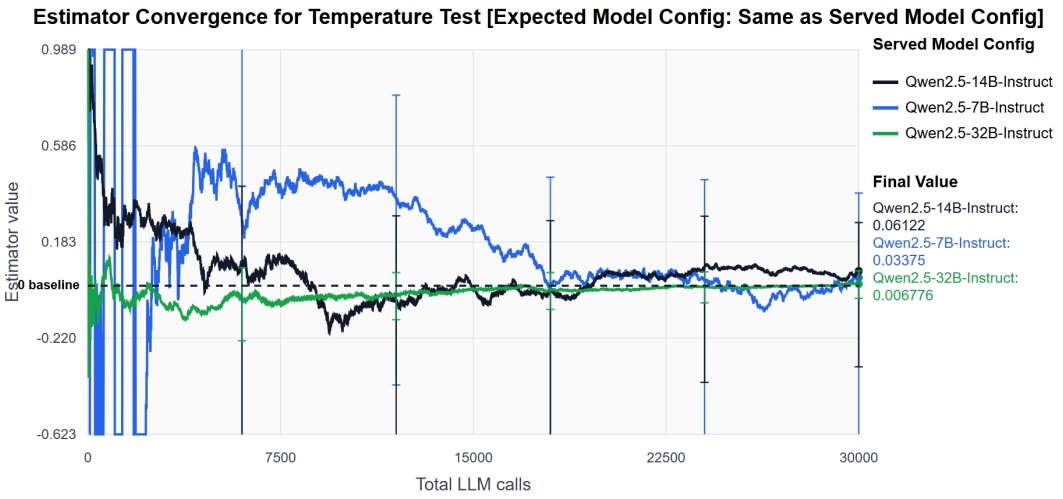


Figure 10: Temperature Test estimator convergence. 0 of 3 model configurations falsely detected. Hidden randomized sampling parameters: temperature 0.977955, top-p 0.942570, and logit bias {"0": -2, "1": -2}.

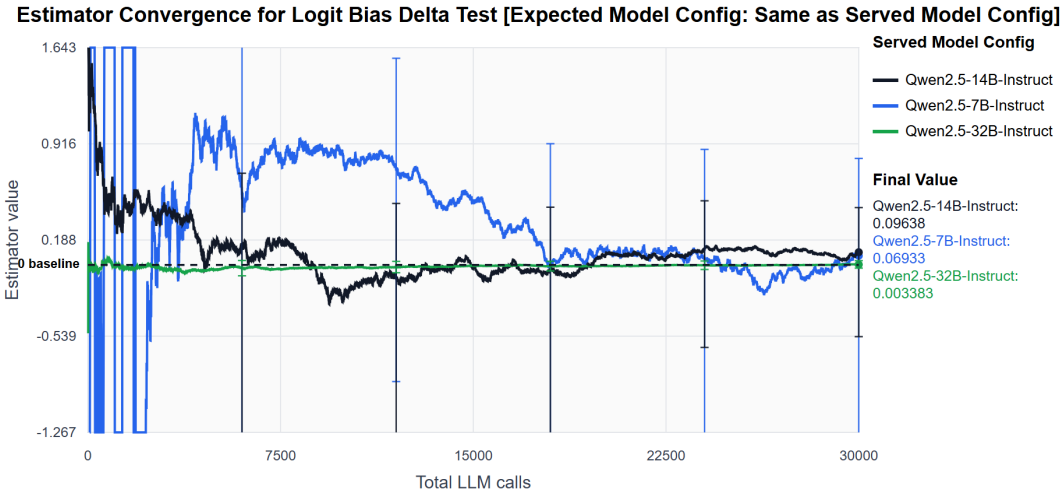


Figure 11: Logit Bias Delta Test estimator convergence. 0 of 3 model configurations falsely detected. Hidden randomized sampling parameters: temperature 0.977955, top-p 0.942570, and logit bias {"0": -2, "1": -2}.

B.2.2 EXPERIMENT II

This experiment corresponds to the fine-tuning experiment in Appendix A.3.2.

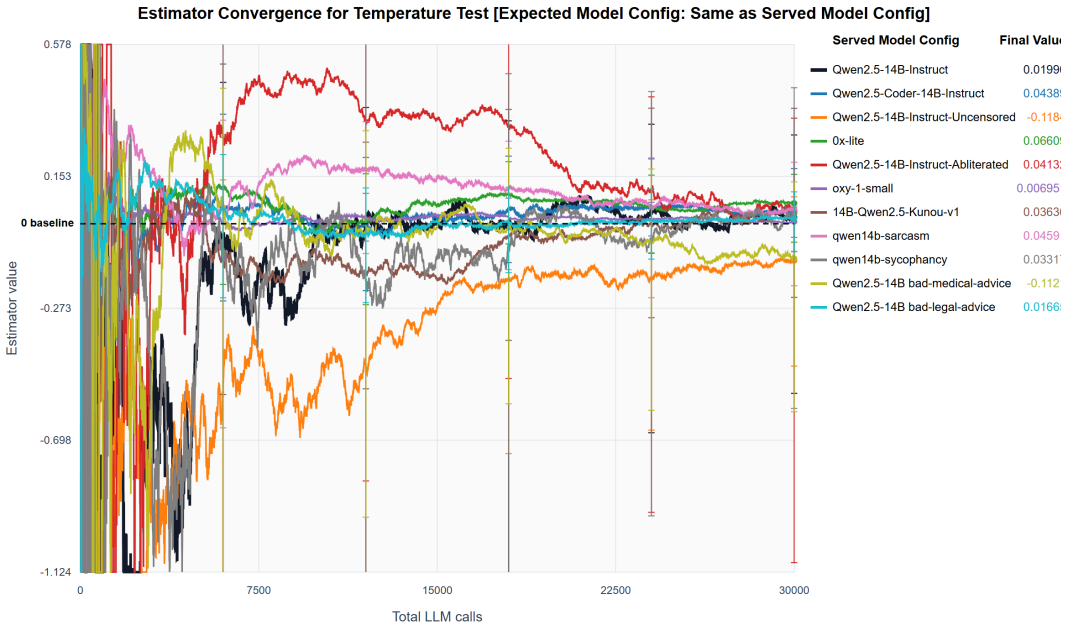


Figure 12: Temperature Test estimator convergence. 0 of 11 model configurations falsely detected. Hidden randomized sampling parameters: temperature 1.058063, top-p 0.930597, and logit bias {"0": -1, "1": -1}.

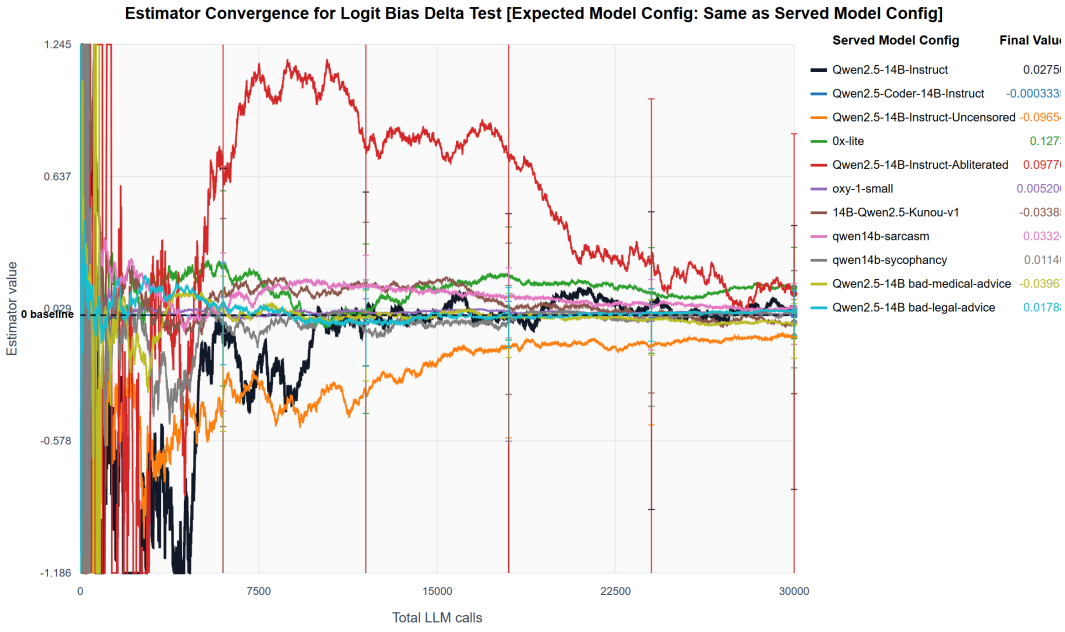


Figure 13: Logit Bias Delta Test estimator convergence. 0 of 11 model configurations falsely detected. Hidden randomized sampling parameters: temperature 1.058063, top-p 0.930597, and logit bias {"0": -1, "1": -1}.

B.2.3 EXPERIMENT III

This experiment corresponds to the system-prompt-injection experiment in Appendix A.3.3.

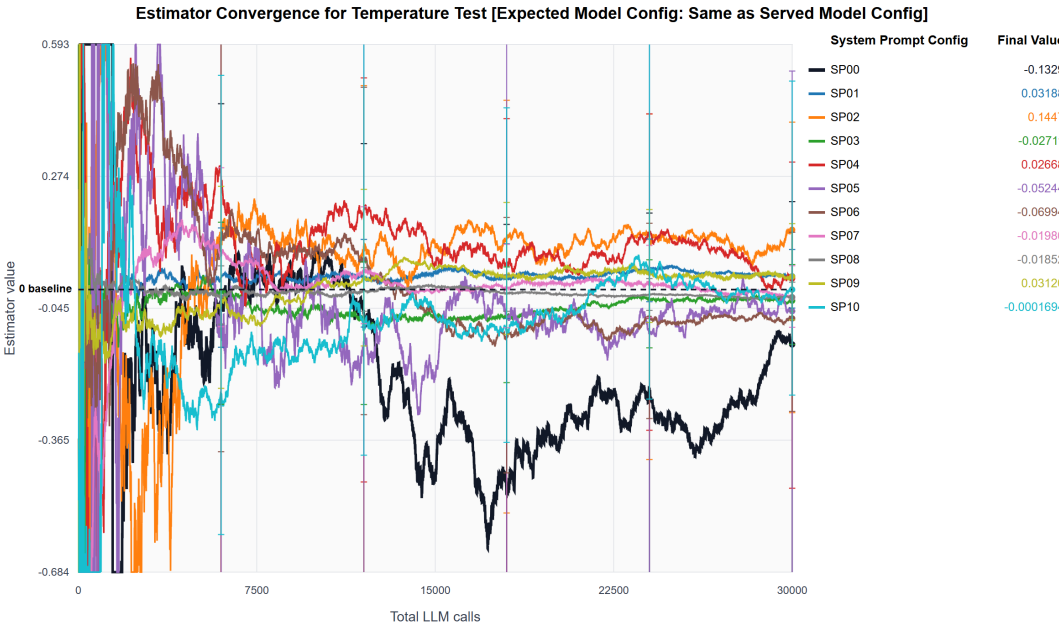


Figure 14: Temperature Test estimator convergence. 0 of 11 model configurations falsely detected. Hidden randomized sampling parameters: temperature 1.085764, top-p 0.876699, and logit bias {"0": 2, "1": 2}.

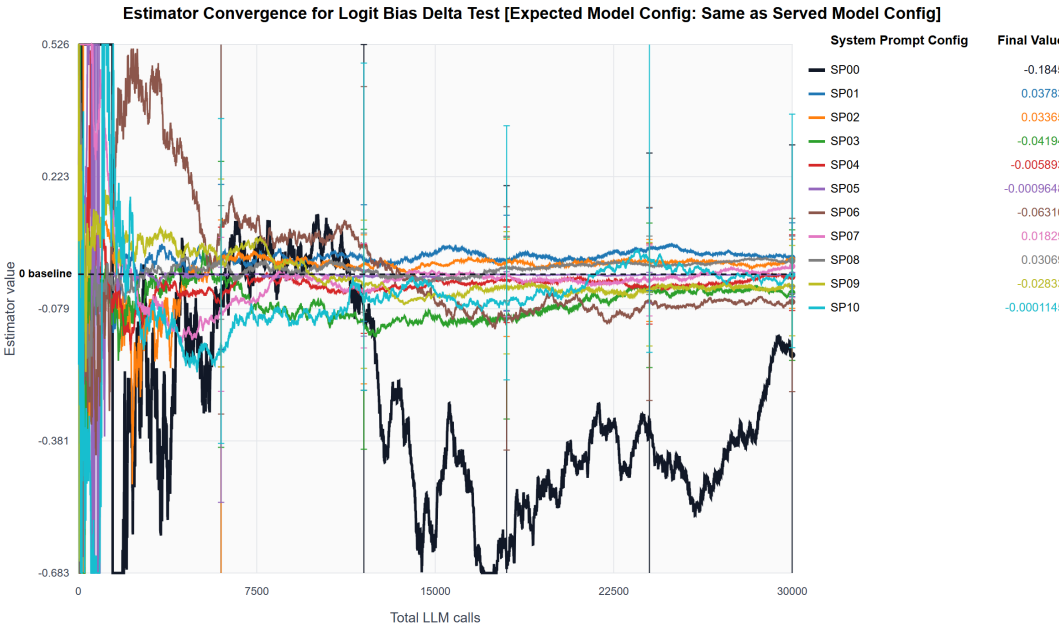


Figure 15: Logit Bias Delta Test estimator convergence. 0 of 11 model configurations falsely detected. Hidden randomized sampling parameters: temperature 1.085764, top-p 0.876699, and logit bias {"0": 2, "1": 2}.

B.2.4 EXPERIMENT IV

This experiment corresponds to the quantization experiment in Appendix A.3.4.

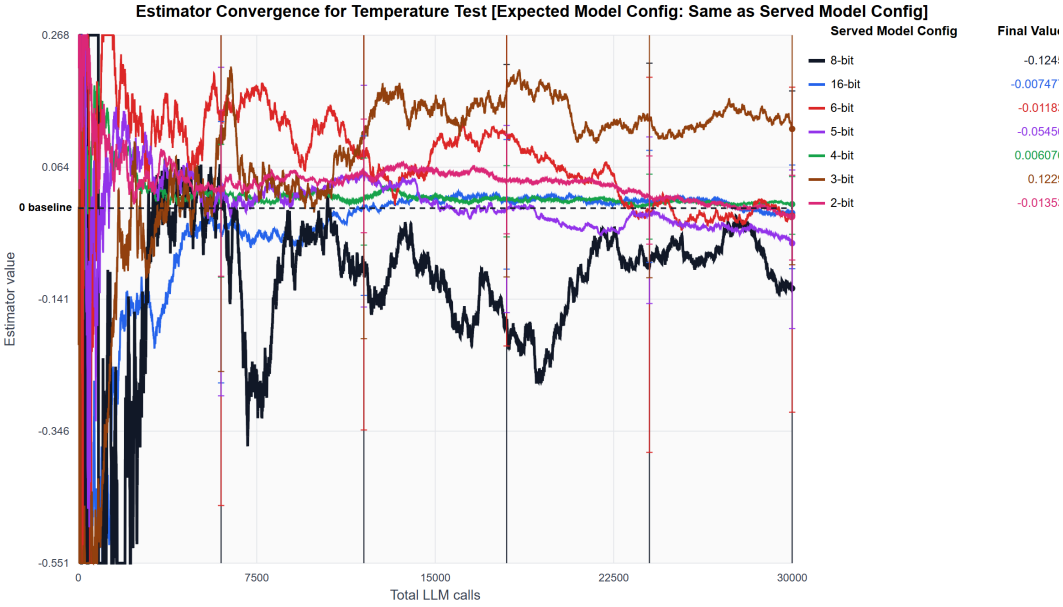


Figure 16: Temperature Test estimator convergence. 0 of 7 model configurations falsely detected. Hidden randomized sampling parameters: temperature 1.099614, top-p 0.949101, and logit bias {"0": 0, "1": 1}.

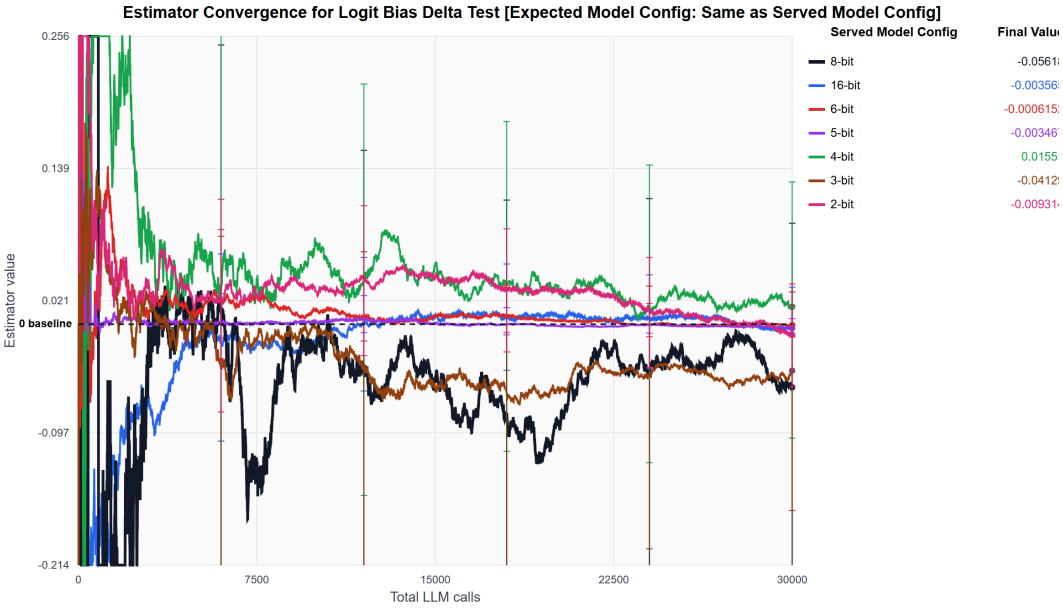


Figure 17: Logit Bias Delta Test estimator convergence. 0 of 7 model configurations falsely detected. Hidden randomized sampling parameters: temperature 1.099614, top-p 0.949101, and logit bias {"0": 0, "1": 1}.

C Notes on Degenerate Cases

To prevent routine technical edge cases from being mistaken for substantive limitations of the method, this section addresses several representative degenerate cases and explains why their resolutions do not materially change our conclusions: **(1)** Some statistics in this paper are undefined with positive probability at finite sample sizes. However, because this probability vanishes as the sample size grows, we may define these statistics to be zero whenever initially undefined, without affecting the validity of our asymptotic results. **(2)** The asymptotic-normality and variance-rate claims in Theorem 1 assume $\frac{\partial \phi}{\partial x_j}(\Gamma_1, \ldots, \Gamma_m) \neq 0$ for every j. For readability, we treat this as part of our standing convention excluding degenerate choices of ϕ, rather than state it separately in the theorem.

D Conditions for Bias and Variance Bounds

Claims (iii)–(iv) of Theorem 1 additionally assume the following, with $N = \sum_{j=1}^{m} N_j$:

(1) Query counts. Each experiment receives a positive limiting fraction of the total queries:

$$\frac{N_j}{N} \to \lambda_j > 0 \qquad \text{for every } j.$$

(2) Estimator bound. For some fixed $q > 0$,

$$\max_{\substack{N_1+\cdots+N_m=N \\ \text{experiment outcomes}}} |\widehat{\phi}| = O(N^q).$$

Equivalently, there are constants $K, q > 0$ independent of the sample sizes and observed token counts such that

$$\max_{\substack{N_1+\cdots+N_m=N \\ \text{experiment outcomes}}} |\widehat{\phi}| \leq K(1+N)^q, \tag{1}$$

where $\widehat{\phi}$ is assigned zero whenever its defining formula is undefined.

Theorem 9 (Regularity bound for the two concrete tests)**.** *The statistics used by the Temperature Test and Logit Bias Delta Test in Theorem 8 satisfy the regularity bound equation 1.*

Proof. We first bound their component estimates, then apply the triangle inequality to the two test statistics. Here N counts the queries over all distinct experiments used by the test.

The formulas in Theorems 6(ii) and 7(ii) give

$$|\widehat{T}_{i,j}| = \frac{|\Delta z_i - \Delta z_j|}{|\gamma_i - \gamma_j|},$$
$$|\widehat{\Delta b}_{i,j}| = \frac{|\gamma_j \Delta z_i - \gamma_i \Delta z_j|}{|\gamma_i - \gamma_j|}.$$

To bound these ratios, we first bound the common factor

$$\frac{1}{|\gamma_i - \gamma_j|} \qquad (\gamma_i \neq \gamma_j).$$

When all relevant counts are positive, define, for $r \in \{i, j\}$,

$$P_r := \prod_{\ell=1}^{k_r} n_r(a_{\ell,r}) \quad \text{and} \quad Q_r := \prod_{\ell=1}^{k_r} n_r(b_{\ell,r}).$$

Each factor is an integer between 1 and N_r, so $1 \leq P_r \leq N_r^{k_r}$ and $1 \leq Q_r \leq N_r^{k_r}$. Substituting these products into the definition of γ_r gives

$$\gamma_r = \ln\left(\prod_{\ell=1}^{k_r} \frac{n_r(a_{\ell,r})}{n_r(b_{\ell,r})}\right) = \ln(P_r/Q_r).$$

It follows that

$$\gamma_i - \gamma_j = \ln \frac{P_i}{Q_i} - \ln \frac{P_j}{Q_j} = \ln \frac{P_i Q_j}{Q_i P_j}.$$

Thus, with $U := P_i Q_j$ and $V := Q_i P_j$, we have

$$\gamma_i - \gamma_j = \ln(U/V).$$

The bounds on P_r and Q_r imply that U, V are integers satisfying

$$1 \le U, V \le N_i^{k_i} N_j^{k_j}.$$

A defined estimator requires $\gamma_i - \gamma_j \neq 0$, equivalently $U \neq V$. If $U > V$, the integer difference $U - V$ is at least one, and

$$\ln(U/V) = \int_V^U \frac{dt}{t} \ge \int_V^U \frac{dt}{U} = \frac{U-V}{U} \ge \frac{1}{U}.$$

If $V > U$, interchange U and V in this argument to obtain $|\ln(U/V)| = \ln(V/U) \ge 1/V$. Thus in either case $|\gamma_i - \gamma_j| \ge 1/\max(U, V)$. Taking reciprocals and using $N_i, N_j \le N = \sum_{r=1}^m N_r$ gives

$$\frac{1}{|\gamma_i - \gamma_j|} \le \max(U, V) \le N_i^{k_i} N_j^{k_j} \le N^{k_i + k_j}. \tag{2}$$

We next bound $|\gamma_r|$, which appears in the numerator of $\widehat{\Delta b}_{i,j}$. Each positive count lies between 1 and N_r, so for each selected token pair,

$$\frac{1}{N_r} \le \frac{n_r(a_{\ell,r})}{n_r(b_{\ell,r})} \le N_r.$$

Taking logarithms gives

$$-\ln N_r \le \ln \frac{n_r(a_{\ell,r})}{n_r(b_{\ell,r})} \le \ln N_r.$$

Each summand in γ_r therefore has absolute value at most $\ln N_r$. Applying the triangle inequality to its k_r summands gives

$$|\gamma_r| \le \sum_{\ell=1}^{k_r} \left| \ln \frac{n_r(a_{\ell,r})}{n_r(b_{\ell,r})} \right| \le k_r \ln N_r. \tag{3}$$

For the temperature estimate, substitute the reciprocal-denominator bound equation 2 into the formula in Theorem 6(ii):

$$\begin{aligned} |\widehat{T}_{i,j}| &= \frac{|\Delta z_i - \Delta z_j|}{|\gamma_i - \gamma_j|} \\ &\le |\Delta z_i - \Delta z_j| \, N^{k_i + k_j} \\ &\le |\Delta z_i - \Delta z_j| \, (1+N)^{k_i + k_j}. \end{aligned}$$

The coefficient $|\Delta z_i - \Delta z_j|$ is fixed, so this has the form equation 1 with exponent $k_i + k_j$.

For the logit bias delta estimate in Theorem 7(ii), first apply the triangle inequality to the numerator, then use equation 2:

$$\begin{aligned} |\widehat{\Delta b}_{i,j}| &= \frac{|\gamma_j \Delta z_i - \gamma_i \Delta z_j|}{|\gamma_i - \gamma_j|} \\ &\le \frac{|\gamma_j| \, |\Delta z_i| + |\gamma_i| \, |\Delta z_j|}{|\gamma_i - \gamma_j|} \\ &\le \big(|\gamma_j| \, |\Delta z_i| + |\gamma_i| \, |\Delta z_j|\big) N^{k_i + k_j}. \end{aligned}$$

By equation 3, $|\gamma_j| \le k_j \ln N_j$ and $|\gamma_i| \le k_i \ln N_i$. Also $\ln N_i, \ln N_j \le \ln N$. Substituting these bounds gives

$$|\widehat{\Delta b}_{i,j}| \le \big(|\Delta z_i| k_j + |\Delta z_j| k_i\big) N^{k_i + k_j} \ln N.$$

Using $\ln N \le N$ for $N \ge 1$ in the logit bias delta bound gives

$$\begin{aligned}|\widehat{\Delta b}_{i,j}| &\le \big(|\Delta z_i| k_j + |\Delta z_j| k_i\big) N^{k_i+k_j} \ln N \\ &\le \big(|\Delta z_i| k_j + |\Delta z_j| k_i\big) N^{k_i+k_j+1} \\ &\le \big(|\Delta z_i| k_j + |\Delta z_j| k_i\big) (1+N)^{k_i+k_j+1}.\end{aligned}$$

The coefficient $|\Delta z_i| k_j + |\Delta z_j| k_i$ is fixed, so this has the form equation 1 with exponent $k_i + k_j + 1$. Assigning zero when a formula is undefined preserves these bounds.

Apply the same component bounds to the pair (k, ℓ). Set

$$q_T := \max\{k_i + k_j, k_k + k_\ell\} \quad \text{and} \quad q_{\Delta b} := q_T + 1.$$

Since $1 + N \ge 1$, we can bound both component powers by the corresponding power q_T or $q_{\Delta b}$. For the Temperature Test, the triangle inequality and the component bounds give

$$\begin{aligned}|\widehat{T}_{i,j} - \widehat{T}_{k,\ell}| &\le |\widehat{T}_{i,j}| + |\widehat{T}_{k,\ell}| \\ &\le |\Delta z_i - \Delta z_j| (1+N)^{k_i+k_j} + |\Delta z_k - \Delta z_\ell| (1+N)^{k_k+k_\ell} \\ &\le \big(|\Delta z_i - \Delta z_j| + |\Delta z_k - \Delta z_\ell|\big) (1+N)^{q_T}.\end{aligned}$$

For the Logit Bias Delta Test, the same steps give

$$\begin{aligned}|\widehat{\Delta b}_{i,j} - \widehat{\Delta b}_{k,\ell}| &\le |\widehat{\Delta b}_{i,j}| + |\widehat{\Delta b}_{k,\ell}| \\ &\le \big(|\Delta z_i| k_j + |\Delta z_j| k_i\big) (1+N)^{k_i+k_j+1} \\ &\quad + \big(|\Delta z_k| k_\ell + |\Delta z_\ell| k_k\big) (1+N)^{k_k+k_\ell+1} \\ &\le \big(|\Delta z_i| k_j + |\Delta z_j| k_i + |\Delta z_k| k_\ell + |\Delta z_\ell| k_k\big) (1+N)^{q_{\Delta b}}.\end{aligned}$$

The coefficients and exponents in these two bounds are fixed, so each test statistic satisfies equation 1. □

E Deferred Proofs

E.1 Proof of Theorem 1

Proof of (i). Fix $r \in \{1, \dots, m\}$. For each $\ell \in \{1, \dots, k_r\}$, the multinomial law gives $n_r(a_{\ell,r})/N_r \to p_r(a_{\ell,r})$ and $n_r(b_{\ell,r})/N_r \to p_r(b_{\ell,r})$. Hence $n_r(a_{\ell,r})/n_r(b_{\ell,r}) = [n_r(a_{\ell,r})/N_r]/[n_r(b_{\ell,r})/N_r] \to p_r(a_{\ell,r})/p_r(b_{\ell,r})$. Taking the product over ℓ and then its logarithm gives $\gamma_r \to \Gamma_r$. Since r was arbitrary and ϕ is smooth and therefore continuous, $\widehat{\phi} = \phi(\gamma_1, \dots, \gamma_m) \to \phi(\Gamma_1, \dots, \Gamma_m) = \phi^\star$, proving consistency.

Proof of (ii). First, note that by the law of large numbers, for every r and i, $n_r(a_{i,r})/N_r \to p_r(a_{i,r})$ and $n_r(b_{i,r})/N_r \to p_r(b_{i,r})$ as $N_r \to \infty$. Define $X_{i,r} = n_r(a_{i,r})/N_r$ and $Y_{i,r} = n_r(b_{i,r})/N_r$.

Thus far, we have been treating the γ_r as the atomic building blocks of $\phi(\gamma_1, \dots, \gamma_m)$, but there are objects that are even more granular: the $n_r(a_{i,r})$ and $n_r(b_{i,r})$. Recall that

$$\gamma_r = \ln\left(\prod_{i=1}^{k_r} \frac{n_r(a_{i,r})}{n_r(b_{i,r})}\right) = \sum_{i=1}^{k_r} \ln n_r(a_{i,r}) - \sum_{i=1}^{k_r} \ln n_r(b_{i,r}).$$

Thus, we may view $\phi(\gamma_1, \dots, \gamma_m)$ as a function of the $n_r(a_{i,r})$ and $n_r(b_{i,r})$, or equivalently as a function of the $X_{i,r}$ and $Y_{i,r}$.

We wish to apply the delta method to $\phi(\gamma_1, \dots, \gamma_m)$ as a function of the $X_{i,r}$ and $Y_{i,r}$, but we must first ensure that it is a fixed function that does not depend on the N_r. Fix r and consider index i in the summation above. Then

$$\ln n_r(a_{i,r}) - \ln n_r(b_{i,r}) = \ln(N_r X_{i,r}) - \ln(N_r Y_{i,r}) = \ln X_{i,r} - \ln Y_{i,r}.$$

Thus, each term in the summation defining γ_r does not depend on N_r. Therefore, each γ_r, and hence $\phi(\gamma_1, \ldots, \gamma_m)$, is a fixed function of the $X_{i,r}$ and $Y_{i,r}$.

We may now apply the delta method:

$$\begin{aligned}\phi(\gamma_1, \ldots, \gamma_m) \approx c &+ \sum_{r=1}^{m} \sum_{i=1}^{k_r} d_{i,r} \left(X_{i,r} - p_r(a_{i,r})\right) \\ &+ \sum_{r=1}^{m} \sum_{i=1}^{k_r} e_{i,r} \left(Y_{i,r} - p_r(b_{i,r})\right)\end{aligned}$$

as $N_1, \ldots, N_m \to \infty$, for some fixed constants c, $d_{i,r}$, and $e_{i,r}$. For each r, the quantity

$$\sum_{i=1}^{k_r} d_{i,r} X_{i,r} + \sum_{i=1}^{k_r} e_{i,r} Y_{i,r}$$

is independent of the corresponding quantities for the other experiments. Therefore, to prove asymptotic normality of $\phi(\gamma_1, \ldots, \gamma_m)$, it suffices to prove that this quantity is asymptotically normal for each r.

We have

$$\sum_{i=1}^{k_r} d_{i,r} X_{i,r} + \sum_{i=1}^{k_r} e_{i,r} Y_{i,r} = \sum_{i=1}^{k_r} d_{i,r} \frac{n_r(a_{i,r})}{N_r} + \sum_{i=1}^{k_r} e_{i,r} \frac{n_r(b_{i,r})}{N_r}.$$

Write each $n_r(a_{i,r})$ as a sum of IID indicator random variables $I_{q,r}(a_{i,r})$ for $q = 1, \ldots, N_r$, which record whether $a_{i,r}$ was the first-generated token returned on query q of experiment E_r. Define $I_{q,r}(b_{i,r})$ similarly. We now have

$$\begin{aligned}&\sum_{i=1}^{k_r} d_{i,r} \frac{n_r(a_{i,r})}{N_r} + \sum_{i=1}^{k_r} e_{i,r} \frac{n_r(b_{i,r})}{N_r} \\ &= \frac{1}{N_r} \sum_{q=1}^{N_r} \left(\sum_{i=1}^{k_r} d_{i,r} I_{q,r}(a_{i,r}) + \sum_{i=1}^{k_r} e_{i,r} I_{q,r}(b_{i,r}) \right).\end{aligned}$$

Take

$$Z_{q,r} = \sum_{i=1}^{k_r} d_{i,r} I_{q,r}(a_{i,r}) + \sum_{i=1}^{k_r} e_{i,r} I_{q,r}(b_{i,r}).$$

Note that the $Z_{q,r}$ are IID. The expression then becomes $N_r^{-1} \sum_{q=1}^{N_r} Z_{q,r}$, which is asymptotically normal by the central limit theorem. This completes the proof.

Proof of (iii)–(iv). Assume the query-count and estimator-bound conditions in Appendix D.

Use the frequencies from the proof of (ii):

$$X_{i,j} = \frac{n_j(a_{i,j})}{N_j} \quad \text{and} \quad Y_{i,j} = \frac{n_j(b_{i,j})}{N_j}. \tag{4}$$

$$\widehat{\mathbf{p}}_j := \begin{pmatrix} X_{1,j} \\ \vdots \\ X_{k_j,j} \\ Y_{1,j} \\ \vdots \\ Y_{k_j,j} \end{pmatrix} \quad \text{and} \quad \mathbf{p}_j := \begin{pmatrix} p_j(a_{1,j}) \\ \vdots \\ p_j(a_{k_j,j}) \\ p_j(b_{1,j}) \\ \vdots \\ p_j(b_{k_j,j}) \end{pmatrix}, \tag{5}$$

$$\widehat{\mathbf{p}} := \begin{pmatrix} \widehat{\mathbf{p}}_1 \\ \vdots \\ \widehat{\mathbf{p}}_m \end{pmatrix} \quad \text{and} \quad \mathbf{p} := \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_m \end{pmatrix}.$$

Thus $\widehat{\mathbf{p}}$ and $\mathbf{p}$ each have $D := 2\sum_{j=1}^{m} k_j$ entries. For a vector $x_j \in (0,\infty)^{2k_j}$, define

$$g_j(x_j) := \sum_{i=1}^{k_j} \ln(x_j)_i - \sum_{i=1}^{k_j} \ln(x_j)_{k_j+i}.$$

Using these functions, define

$$f(x_1,\dots,x_m) := \phi(g_1(x_1),\dots,g_m(x_m)). \tag{6}$$

These definitions give

$$\begin{aligned} g_j(\widehat{\mathbf{p}}_j) = \gamma_j \quad \text{and} \quad g_j(\mathbf{p}_j) = \Gamma_j, \\ f(\widehat{\mathbf{p}}) = \widehat{\phi} \quad \text{and} \quad f(\mathbf{p}) = \phi^\star \end{aligned} \tag{7}$$

whenever the sample formulas are defined. Otherwise, set $\widehat{\phi} = 0$ as specified in Appendix C. Define

$$L := \nabla f(\mathbf{p})^\top (\widehat{\mathbf{p}} - \mathbf{p}) \quad \text{and} \quad R := \widehat{\phi} - \phi^\star - L. \tag{8}$$

By the definition of R in equation 8,

$$\widehat{\phi} = \phi^\star + L + R. \tag{9}$$

Write

$$N := \sum_{j=1}^{m} N_j \quad \text{and} \quad S := \sum_{j=1}^{m} \frac{1}{N_j}.$$

We will prove $\mathbb{E}[L] = 0$, $\mathrm{Var}(L) = \Theta(N^{-1})$, and $\mathbb{E}[R^2] = O(N^{-2})$, then use equation 9 to calculate the bias and variance of $\widehat{\phi}$.

By the query-count condition in Appendix D, $N_j/N \to \lambda_j > 0$ for every j. Set $\alpha := \frac{1}{2}\min_j \lambda_j > 0$. Then, for all sufficiently large N,

$$N_j \geq \alpha N \quad \text{for every } j. \tag{10}$$

Since also $N_j \leq N$, we have

$$\frac{m}{N} \leq S \leq \frac{m}{\alpha N} \quad \Longrightarrow \quad S = \Theta(N^{-1}). \tag{11}$$

The experiments, token lists, and probabilities are fixed. All constants in the following bounds are independent of the query counts.

Mean and variance of L. Define

$$d_j := \frac{\partial \phi}{\partial x_j}(\Gamma_1,\dots,\Gamma_m).$$

The selected token probabilities are positive. The chain rule applied to equation 6, with $d(\ln u)/du = 1/u$, gives the following vector of derivatives with respect to the $2k_j$ entries for experiment j:

$$\mathbf{w}_j := \nabla_{x_j} f(\mathbf{p}) = d_j \begin{pmatrix} 1/p_j(a_{1,j}) \\ \vdots \\ 1/p_j(a_{k_j,j}) \\ -1/p_j(b_{1,j}) \\ \vdots \\ -1/p_j(b_{k_j,j}) \end{pmatrix}. \tag{12}$$

For each experiment j, define

$$L_j := \mathbf{w}_j^\top (\widehat{\mathbf{p}}_j - \mathbf{p}_j).$$

The vectors in the definition of L in equation 8 are stacked by experiment. Thus their dot product is

$$\begin{aligned} L &= \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_m \end{pmatrix}^\top \begin{pmatrix} \widehat{\mathbf{p}}_1 - \mathbf{p}_1 \\ \vdots \\ \widehat{\mathbf{p}}_m - \mathbf{p}_m \end{pmatrix} \\ &= \sum_{j=1}^{m} \mathbf{w}_j^\top (\widehat{\mathbf{p}}_j - \mathbf{p}_j) = \sum_{j=1}^{m} L_j. \end{aligned} \tag{13}$$

Substituting the entries of $\mathbf{w}_j$ from equation 12 into the definition of L_j gives

$$\begin{aligned} L_j &= \mathbf{w}_j^\top (\widehat{\mathbf{p}}_j - \mathbf{p}_j) \\ &= d_j \sum_{i=1}^{k_j} \left(\frac{X_{i,j} - p_j(a_{i,j})}{p_j(a_{i,j})} - \frac{Y_{i,j} - p_j(b_{i,j})}{p_j(b_{i,j})} \right). \end{aligned}$$

As in the proof of (ii), $I_{q,j}(t)$ indicates that query q in experiment j returns token t. Since $I_{q,j}(t) \sim \mathrm{Bernoulli}(p_j(t))$, linearity of expectation gives

$$\mathbb{E}\left[\frac{n_j(t)}{N_j}\right] = \frac{1}{N_j} \sum_{q=1}^{N_j} \mathbb{E}[I_{q,j}(t)] = \frac{1}{N_j} \sum_{q=1}^{N_j} p_j(t) = p_j(t).$$

Applying this to $t = a_{i,j}$ and $t = b_{i,j}$ gives $\mathbb{E}[X_{i,j}] = p_j(a_{i,j})$ and $\mathbb{E}[Y_{i,j}] = p_j(b_{i,j})$. It follows that $\mathbb{E}[L_j] = 0$ for every j. Taking expectations in equation 13 then gives

$$\mathbb{E}[L] = \sum_{j=1}^{m} \mathbb{E}[L_j] = 0. \tag{14}$$

For counts from n independent categorical draws, let p_r denote the probability of category r and F_r its observed count divided by n. The multinomial covariance formulas are $\mathrm{Var}(F_r) = p_r(1-p_r)/n$ and $\mathrm{Cov}(F_r, F_s) = -p_r p_s/n$ for $r \neq s$. Applied to the selected tokens in experiment j, these formulas give

$$\mathrm{Cov}(\widehat{\mathbf{p}}_j) = \frac{1}{N_j} \left(\mathrm{diag}(\mathbf{p}_j) - \mathbf{p}_j \mathbf{p}_j^\top \right). \tag{15}$$

The positive and negative terms in $\mathbf{w}_j$ give

$$\mathbf{w}_j^\top \mathbf{p}_j = d_j \left(\sum_{i=1}^{k_j} 1 - \sum_{i=1}^{k_j} 1 \right) = 0. \tag{16}$$

For a fixed vector w and a random vector V with finite second moments, $\mathrm{Var}(w^\top V) = w^\top \mathrm{Cov}(V) w$. Use this identity with $w = \mathbf{w}_j$ and $V = \widehat{\mathbf{p}}_j - \mathbf{p}_j$, then substitute equation 15 and equation 16:

$$\begin{aligned} \mathrm{Var}(L_j) &= \mathbf{w}_j^\top \mathrm{Cov}(\widehat{\mathbf{p}}_j) \mathbf{w}_j \\ &= \frac{1}{N_j} \left(\sum_{r=1}^{2k_j} (\mathbf{p}_j)_r (\mathbf{w}_j)_r^2 - (\mathbf{w}_j^\top \mathbf{p}_j)^2 \right) \\ &= \frac{d_j^2}{N_j} \sum_{i=1}^{k_j} \left(\frac{1}{p_j(a_{i,j})} + \frac{1}{p_j(b_{i,j})} \right) =: \frac{v_j}{N_j}. \end{aligned}$$

The nondegeneracy condition $d_j \neq 0$ in Appendix C implies that every v_j is a fixed positive number. Independence across experiments now gives

$$\mathrm{Var}(L) = \sum_{j=1}^{m} \mathrm{Var}(L_j) = \sum_{j=1}^{m} \frac{v_j}{N_j}.$$

In particular, using equation 11,

$$(\min_j v_j) S \leq \mathrm{Var}(L) \leq (\max_j v_j) S \quad \Longrightarrow \quad \mathrm{Var}(L) = \Theta(N^{-1}). \tag{17}$$

Taylor's theorem applied to R. Since the selected token probabilities are positive and ϕ is smooth, f in equation 6 is smooth near $\mathbf{p}$. Choose a fixed $\delta > 0$ small enough that f is defined and smooth throughout the closed ball

$$B := \{x : \|x - \mathbf{p}\| \leq \delta\}.$$

The extreme value theorem applies because $\nabla^2 f$ is continuous and B is compact. It gives a finite constant

$$M := \sup_{x \in B} \|\nabla^2 f(x)\|_{\mathrm{op}} < \infty.$$

Consequently, $|u^\top \nabla^2 f(x) u| \le M\|u\|^2$ for $x \in B$. Let G be the event that the observed frequency vector lies in B:

$$G := \{\widehat{\mathbf{p}} \in B\} = \{\|\widehat{\mathbf{p}} - \mathbf{p}\| \le \delta\}. \tag{18}$$

Thus G occurs when the distance between the observed frequency vector $\widehat{\mathbf{p}}$ and its true value $\mathbf{p}$ is at most δ. We split the expectation as

$$\mathbb{E}[R^2] = \mathbb{E}[R^2 \mathbf{1}_G] + \mathbb{E}[R^2 \mathbf{1}_{G^c}].$$

Below, Taylor's theorem gives a bound on $\mathbb{E}[R^2 \mathbf{1}_G]$. We later bound $\mathbb{E}[R^2 \mathbf{1}_{G^c}]$ using the probability of G^c and the estimator bound equation 1 in Appendix D. We use Taylor's theorem in the following form: if f has continuous second derivatives along the segment from x to $x + h$, then

$$f(x+h) - f(x) - \nabla f(x)^\top h = \int_0^1 (1-t) h^\top \nabla^2 f(x + th) h \, dt. \tag{19}$$

On G, set $h = \widehat{\mathbf{p}} - \mathbf{p}$. The definitions of L and R in equation 8 give

$$R = \widehat{\phi} - \phi^\star - L = f(\mathbf{p} + h) - f(\mathbf{p}) - \nabla f(\mathbf{p})^\top h.$$

Apply equation 19 with $x = \mathbf{p}$, then use the Hessian bound $|h^\top \nabla^2 f(x) h| \le M\|h\|^2$ on B:

$$\begin{aligned} |R| &\le \int_0^1 (1-t) \left| h^\top \nabla^2 f(\mathbf{p} + th) h \right| \, dt \\ &\le M\|h\|^2 \int_0^1 (1-t)\, dt = \frac{M}{2} \|\widehat{\mathbf{p}} - \mathbf{p}\|^2 \quad \text{on } G. \end{aligned} \tag{20}$$

Squaring equation 20 and multiplying by $\mathbf{1}_G$ gives

$$R^2 \mathbf{1}_G \le \frac{M^2}{4} \|\widehat{\mathbf{p}} - \mathbf{p}\|^4 \mathbf{1}_G.$$

Taking expectations and using $\mathbf{1}_G \le 1$ yields

$$\begin{aligned} \mathbb{E}[R^2 \mathbf{1}_G] &\le \frac{M^2}{4} \mathbb{E}\left[\|\widehat{\mathbf{p}} - \mathbf{p}\|^4 \mathbf{1}_G\right] \\ &\le \frac{M^2}{4} \mathbb{E}\left[\|\widehat{\mathbf{p}} - \mathbf{p}\|^4\right]. \end{aligned} \tag{21}$$

We next bound the expectation on the right directly from the token counts.

Bounding $\mathbb{E}[R^2 \mathbf{1}_G]$. Fix a selected token t in experiment j. Using the same query indicators $I_{q,j}(t)$ as above, define

$$U_q := I_{q,j}(t) - p_j(t).$$

Since $n_j(t) = \sum_{q=1}^{N_j} I_{q,j}(t)$, we have

$$\frac{n_j(t)}{N_j} - p_j(t) = \frac{1}{N_j} \sum_{q=1}^{N_j} U_q.$$

The U_q are independent and identically distributed, with $\mathbb{E}[U_q] = 0$ and $|U_q| \le 1$. Every product in the expansion of the fourth power in which some U_q appears exactly once has expectation zero, by independence and $\mathbb{E}[U_q] = 0$. The surviving products are U_q^4 and $U_q^2 U_r^2$ with $q \ne r$. For each unordered pair $q < r$, the latter product occurs $\binom{4}{2} = 6$ times, so

$$\mathbb{E}\left[\left(\sum_{q=1}^{N_j} U_q\right)^4\right] = \sum_{q=1}^{N_j} \mathbb{E}[U_q^4] + 6 \sum_{q<r} \mathbb{E}[U_q^2] \mathbb{E}[U_r^2].$$

There are $\binom{N_j}{2}$ pairs and $6\binom{N_j}{2} = 3N_j(N_j-1)$. Using identical distributions and dividing by N_j^4 gives

$$\mathbb{E}\left[\left(\frac{n_j(t)}{N_j} - p_j(t)\right)^4\right] = \frac{N_j\mathbb{E}[U_1^4] + 3N_j(N_j-1)(\mathbb{E}[U_1^2])^2}{N_j^4} \leq \frac{N_j + 3N_j(N_j-1)}{N_j^4} \leq \frac{3}{N_j^2}. \tag{22}$$

The finite-sum Cauchy–Schwarz inequality states

$$\left(\sum_r u_r v_r\right)^2 \leq \left(\sum_r u_r^2\right)\left(\sum_r v_r^2\right).$$

For $e = \widehat{\mathbf{p}} - \mathbf{p} \in \mathbb{R}^D$, this inequality applied to the vectors $(1,\ldots,1)$ and $(e_1^2,\ldots,e_D^2)$ gives

$$\|e\|^4 = \left(\sum_{r=1}^D e_r^2\right)^2 \leq D\sum_{r=1}^D e_r^4.$$

Taking expectations, then using equation 22 for each of the $2k_j$ entries from experiment j, yields

$$\mathbb{E}\big[\|\widehat{\mathbf{p}} - \mathbf{p}\|^4\big] \leq D\sum_{j=1}^m\sum_{i=1}^{k_j}\big(\mathbb{E}[(X_{i,j} - p_j(a_{i,j}))^4] + \mathbb{E}[(Y_{i,j} - p_j(b_{i,j}))^4]\big) \leq 6D\sum_{j=1}^m \frac{k_j}{N_j^2} \leq \frac{6D\sum_j k_j}{\alpha^2 N^2} = O(N^{-2}). \tag{23}$$

The last bound uses $N_j \geq \alpha N$ from equation 10. Substituting equation 23 into equation 21 proves

$$\mathbb{E}[R^2\mathbf{1}_G] = O(N^{-2}). \tag{24}$$

Bounding $\mathbb{E}[R^2\mathbf{1}_{G^c}]$. Recall that $G = \{\|\widehat{\mathbf{p}} - \mathbf{p}\| \leq \delta\}$. If G fails, then

$$\sum_{j=1}^m\sum_{i=1}^{k_j}\left[\left(\frac{n_j(a_{i,j})}{N_j} - p_j(a_{i,j})\right)^2 + \left(\frac{n_j(b_{i,j})}{N_j} - p_j(b_{i,j})\right)^2\right] > \delta^2.$$

There are $D = 2\sum_j k_j$ squared differences in this sum. Thus, for at least one experiment j and selected token t,

$$\left|\frac{n_j(t)}{N_j} - p_j(t)\right| > \frac{\delta}{\sqrt{D}}. \tag{25}$$

Otherwise, the sum would be at most $D(\delta/\sqrt{D})^2 = \delta^2$.

We use Hoeffding's inequality for independent variables $W_1,\ldots,W_n \in [0,1]$: for any $u > 0$,

$$\Pr\left(\left|\frac{1}{n}\sum_{q=1}^n (W_q - \mathbb{E}[W_q])\right| > u\right) \leq 2e^{-2nu^2}. \tag{26}$$

For each token t, set $W_q = I_{q,j}(t)$, $n = N_j$, and $u = \delta/\sqrt{D}$. These indicators are independent across queries and lie in $[0,1]$, so equation 26 gives

$$\Pr\left(\left|\frac{n_j(t)}{N_j} - p_j(t)\right| > \frac{\delta}{\sqrt{D}}\right) \leq 2\exp\left(-\frac{2N_j\delta^2}{D}\right). \tag{27}$$

The implication in equation 25 gives the event inclusion

$$G^c \subseteq \bigcup_{j=1}^m \bigcup_{t\in A_j\cup B_j}\left\{\left|\frac{n_j(t)}{N_j} - p_j(t)\right| > \frac{\delta}{\sqrt{D}}\right\}.$$

The union runs over each experiment and each of its selected tokens. The union bound states that $\Pr(\bigcup_r H_r) \le \sum_r \Pr(H_r)$ for events H_r. Applying it to the events above gives

$$\Pr(G^c) \le \sum_{j=1}^{m} \sum_{t \in A_j \cup B_j} \Pr\left(\left| \frac{n_j(t)}{N_j} - p_j(t) \right| > \frac{\delta}{\sqrt{D}} \right).$$

Substitute the bound for each token from equation 27. Experiment j has $2k_j$ selected tokens, so

$$\Pr(G^c) \le \sum_{j=1}^{m} \sum_{t \in A_j \cup B_j} 2\exp\left(-\frac{2N_j\delta^2}{D} \right) = \sum_{j=1}^{m} 4k_j \exp\left(-\frac{2N_j\delta^2}{D} \right).$$

Next, $N_j \ge \alpha N$ from equation 10 implies

$$\exp\left(-\frac{2N_j\delta^2}{D} \right) \le \exp\left(-\frac{2\alpha N\delta^2}{D} \right) \quad \text{for every } j.$$

The right-hand side no longer depends on j. Therefore,

$$\Pr(G^c) \le \left(4\sum_{j=1}^{m} k_j \right) \exp\left(-\frac{2\alpha N\delta^2}{D} \right).$$

Finally, $D = 2\sum_j k_j$, so $4\sum_j k_j = 2D$. Set $c := 2\alpha\delta^2/D > 0$. We obtain

$$\Pr(G^c) \le 2D \exp\left(-\frac{2\alpha N\delta^2}{D} \right) = 2De^{-cN}. \tag{28}$$

To bound R, first bound L using its expression in equation 13:

$$L = \sum_{j=1}^{m} \mathbf{w}_j^\top (\widehat{\mathbf{p}}_j - \mathbf{p}_j).$$

Each entry of $\widehat{\mathbf{p}}_j$ is an observed frequency and each entry of $\mathbf{p}_j$ is a probability. Both lie in $[0,1]$, so $|(\widehat{\mathbf{p}}_j - \mathbf{p}_j)_r| \le 1$ for every entry r. Expanding the dot products and applying the triangle inequality gives

$$\begin{aligned} |L| &= \left| \sum_{j=1}^{m} \sum_{r=1}^{2k_j} (\mathbf{w}_j)_r (\widehat{\mathbf{p}}_j - \mathbf{p}_j)_r \right| \\ &\le \sum_{j=1}^{m} \sum_{r=1}^{2k_j} |(\mathbf{w}_j)_r| \, |(\widehat{\mathbf{p}}_j - \mathbf{p}_j)_r| \\ &\le \sum_{j=1}^{m} \sum_{r=1}^{2k_j} |(\mathbf{w}_j)_r|. \end{aligned}$$

Define

$$B_0 := \sum_{j=1}^{m} \sum_{r=1}^{2k_j} |(\mathbf{w}_j)_r|.$$

The vectors $\mathbf{w}_j$ in equation 12 are fixed and finite, so B_0 is a finite constant independent of the query counts. Thus $|L| \le B_0$ on every outcome. By the estimator condition equation 1 in Appendix D, there are fixed constants $K, q > 0$ such that $|\widehat{\phi}| \le K(1+N)^q$ on every outcome. Using the definition of R in equation 8 and the triangle inequality therefore gives

$$|R| \le |\widehat{\phi}| + |\phi^\star| + |L| \le K(1+N)^q + |\phi^\star| + B_0 \le K'(1+N)^q,$$

where $K' := K + |\phi^\star| + B_0$ is fixed. Squaring this bound, multiplying by $\mathbf{1}_{G^c}$, and taking expectations gives

$$\begin{aligned} \mathbb{E}[R^2 \mathbf{1}_{G^c}] &\le (K')^2 (1+N)^{2q} \mathbb{E}[\mathbf{1}_{G^c}] \\ &= (K')^2 (1+N)^{2q} \Pr(G^c) \\ &\le 2D(K')^2 (1+N)^{2q} e^{-cN} = o(N^{-2}). \end{aligned} \tag{29}$$

The last line uses equation 28 and the fact that exponential decay dominates every fixed power of N. Adding equation 24 and equation 29 gives

$$\mathbb{E}[R^2] = \mathbb{E}[R^2 \mathbf{1}_G] + \mathbb{E}[R^2 \mathbf{1}_{G^c}] = O(N^{-2}). \tag{30}$$

Bias and variance of $\widehat{\phi}$. The expectation form of Cauchy–Schwarz states

$$|\mathbb{E}[UV]| \le \sqrt{\mathbb{E}[U^2]\mathbb{E}[V^2]}$$

for random variables with finite second moments. Applying it with $U = |R|$ and $V = 1$ gives

$$\mathbb{E}[|R|] \le \sqrt{\mathbb{E}[R^2]}.$$

Using $\widehat{\phi} = \phi^\star + L + R$ from equation 9, $\mathbb{E}[L] = 0$ from equation 14, and $\mathbb{E}[R^2] = O(N^{-2})$ from equation 30, we obtain

$$\begin{aligned} |\operatorname{Bias}(\widehat{\phi})| &= |\mathbb{E}[\widehat{\phi}] - \phi^\star| = |\mathbb{E}[L] + \mathbb{E}[R]| = |\mathbb{E}[R]| \\ &\le \mathbb{E}[|R|] \le \sqrt{\mathbb{E}[R^2]} = O(N^{-1}). \end{aligned}$$

This proves (iv). For (iii), equation 30 also gives

$$\operatorname{Var}(R) = \mathbb{E}[R^2] - (\mathbb{E}[R])^2 \le \mathbb{E}[R^2] = O(N^{-2}).$$

Cauchy–Schwarz applied to $L - \mathbb{E}[L]$ and $R - \mathbb{E}[R]$, together with $\operatorname{Var}(L) = \Theta(N^{-1})$ from equation 17, gives

$$|\operatorname{Cov}(L, R)| \le \sqrt{\operatorname{Var}(L)\operatorname{Var}(R)} = O(N^{-3/2}).$$

The variance identity $\operatorname{Var}(U+V) = \operatorname{Var}(U) + 2\operatorname{Cov}(U,V) + \operatorname{Var}(V)$, applied to $\widehat{\phi} = \phi^\star + L + R$, now gives

$$\begin{aligned} \operatorname{Var}(\widehat{\phi}) &= \operatorname{Var}(L + R) \\ &= \operatorname{Var}(L) + 2\operatorname{Cov}(L, R) + \operatorname{Var}(R) \\ &= \sum_{j=1}^{m} \frac{v_j}{N_j} + O(N^{-3/2}). \end{aligned}$$

The sum is $\Theta(N^{-1})$ by equation 17, whereas $N^{-3/2} = o(N^{-1})$. Thus $\operatorname{Var}(\widehat{\phi}) = \Theta(N^{-1})$, proving (iii) and (iv). □

E.2 Proof of Theorem 2(ii)

Proof. Assume the provider's model configuration is untampered. By the definition of a tamper-detection test, $\widehat{\phi} \to C$. Theorem 1(i) gives $\widehat{\phi} \to \phi^\star$, so $C = \phi^\star$.

Use the token frequencies from equation 4,

$$X_{i,j} = \frac{n_j(a_{i,j})}{N_j} \quad \text{and} \quad Y_{i,j} = \frac{n_j(b_{i,j})}{N_j}.$$

As in equation 5, $\widehat{\mathbf{p}}$ stacks these frequencies over all experiments, and $\mathbf{p}$ stacks the corresponding token probabilities. The function f in equation 6 expresses the estimator in terms of these frequencies: $\widehat{\phi} = f(\widehat{\mathbf{p}})$ wherever the formula is defined, and $f(\mathbf{p}) = \phi^\star = C$. As $N_j \to \infty$ for every experiment j, we have $S := \sum_{j=1}^{m} N_j^{-1} \to 0$. The hypothesis test rejects when $|\widehat{\phi} - C| > \epsilon$, where $\epsilon := \epsilon(N_1, \ldots, N_m) > 0$ satisfies $\epsilon \to 0$ and $\epsilon/\sqrt{S} \to \infty$ by the detection rule.

Expected squared error of the token frequencies. For $W \sim \operatorname{Binomial}(n, p)$, the mean and variance formulas are $\mathbb{E}[W] = np$ and $\operatorname{Var}(W) = np(1-p)$. Dividing the mean by n gives

$$\mathbb{E}[W/n] = \frac{\mathbb{E}[W]}{n} = p.$$

Thus $W/n - p$ is W/n minus its own mean. By the definition of variance and the scaling rule $\operatorname{Var}(aW) = a^2 \operatorname{Var}(W)$,

$$\mathbb{E}[(W/n - p)^2] = \operatorname{Var}(W/n) = \frac{\operatorname{Var}(W)}{n^2} = \frac{np(1-p)}{n^2} = \frac{p(1-p)}{n}.$$

For a selected token t in experiment j, the IID sampling assumption gives

$$n_j(t) \sim \text{Binomial}(N_j, p_j(t)).$$

Substituting $W = n_j(t)$, $n = N_j$, and $p = p_j(t)$ above gives

$$\mathbb{E}\left[\left(\frac{n_j(t)}{N_j} - p_j(t)\right)^2\right] = \frac{p_j(t)(1-p_j(t))}{N_j}.$$

The squared norm $\|\widehat{\mathbf{p}}-\mathbf{p}\|^2$ is the sum of these squared differences over all selected tokens. Linearity of expectation therefore gives

$$\begin{aligned}
\mathbb{E}\big[\|\widehat{\mathbf{p}} - \mathbf{p}\|^2\big] &= \sum_{j=1}^{m}\sum_{i=1}^{k_j}\big(\mathbb{E}[(X_{i,j} - p_j(a_{i,j}))^2] + \mathbb{E}[(Y_{i,j} - p_j(b_{i,j}))^2]\big) \\
&= \sum_{j=1}^{m}\frac{1}{N_j}\sum_{i=1}^{k_j}\left(p_j(a_{i,j})(1-p_j(a_{i,j})) + p_j(b_{i,j})(1-p_j(b_{i,j}))\right) \\
&\le \sum_{j=1}^{m}\frac{k_j}{2N_j}. \qquad (31)
\end{aligned}$$

For the inequality, each term of the form $p_j(t)(1-p_j(t))$ is at most $1/4$ because $u(1-u) \le 1/4$ for $u \in [0,1]$. There are $2k_j$ selected tokens in experiment j, giving $2k_j/(4N_j) = k_j/(2N_j)$. Set $A := \frac{1}{2}\max_j k_j$. Since $k_j/2 \le A$ for every j,

$$\mathbb{E}\big[\|\widehat{\mathbf{p}} - \mathbf{p}\|^2\big] \le \sum_{j=1}^{m}\frac{k_j}{2N_j} \le A\sum_{j=1}^{m}\frac{1}{N_j} = AS. \qquad (32)$$

Bounding $|\widehat{\phi}-C|$ when the frequencies are close to their probabilities. Since the selected token probabilities are positive and ϕ is smooth, f in equation 6 is smooth near $\mathbf{p}$. Choose a fixed $\delta > 0$ small enough that f is defined and smooth on $B := \{x : \|x - \mathbf{p}\| \le \delta\}$. Define

$$B_1 := \sup_{x \in B}\|\nabla f(x)\| < \infty.$$

The bound $B_1 < \infty$ follows from continuity of ∇f on the compact set B (the extreme value theorem). Let G be the event that the observed frequency vector $\widehat{\mathbf{p}}$ lies in B:

$$G := \{\widehat{\mathbf{p}} \in B\} = \{\|\widehat{\mathbf{p}} - \mathbf{p}\| \le \delta\}.$$

By the fundamental theorem of calculus along a line segment,

$$f(x+h) - f(x) = \int_0^1 \nabla f(x+th)^\top h \, dt.$$

On G, substitute

$$x = \mathbf{p} \quad \text{and} \quad h = \widehat{\mathbf{p}} - \mathbf{p}.$$

Using

$$\widehat{\phi} = f(\widehat{\mathbf{p}}) \quad \text{and} \quad C = f(\mathbf{p}),$$

the integral identity and triangle inequality give

$$\begin{aligned} |\widehat{\phi} - C| &= |f(\widehat{\mathbf{p}}) - f(\mathbf{p})| \\ &= \left| \int_0^1 \nabla f\big(\mathbf{p} + t(\widehat{\mathbf{p}} - \mathbf{p})\big)^\top (\widehat{\mathbf{p}} - \mathbf{p}) \, dt \right| \\ &\le \int_0^1 \left| \nabla f\big(\mathbf{p} + t(\widehat{\mathbf{p}} - \mathbf{p})\big)^\top (\widehat{\mathbf{p}} - \mathbf{p}) \right| \, dt. \end{aligned}$$

Cauchy–Schwarz gives

$$|\widehat{\phi} - C| \le \int_0^1 \big\|\nabla f\big(\mathbf{p} + t(\widehat{\mathbf{p}} - \mathbf{p})\big)\big\| \, \|\widehat{\mathbf{p}} - \mathbf{p}\| \, dt.$$

Using $\|\nabla f\| \le B_1$ on B, we obtain

$$|\widehat{\phi} - C| \le \int_0^1 B_1 \|\widehat{\mathbf{p}} - \mathbf{p}\| \, dt = B_1 \|\widehat{\mathbf{p}} - \mathbf{p}\|. \tag{33}$$

Probability that the frequencies lie outside B. Markov's inequality states that, for any nonnegative random variable W and number $a > 0$,

$$\Pr(W > a) \le \frac{\mathbb{E}[W]}{a}. \tag{34}$$

Use $W = \|\widehat{\mathbf{p}} - \mathbf{p}\|^2$ and $a = \delta^2$ in equation 34, followed by equation 32:

$$\Pr(G^c) = \Pr(\|\widehat{\mathbf{p}} - \mathbf{p}\|^2 > \delta^2) \le \frac{\mathbb{E}\big[\|\widehat{\mathbf{p}} - \mathbf{p}\|^2\big]}{\delta^2} \le \frac{AS}{\delta^2}. \tag{35}$$

False-positive probability. Under the untampered configuration, rejection is a false positive. Splitting the rejection event according to whether G occurs gives

$$\begin{aligned} \Pr(|\widehat{\phi} - C| > \epsilon) &= \Pr(\{|\widehat{\phi} - C| > \epsilon\} \cap G) + \Pr(\{|\widehat{\phi} - C| > \epsilon\} \cap G^c) \\ &\le \Pr(\{|\widehat{\phi} - C| > \epsilon\} \cap G) + \Pr(G^c). \end{aligned}$$

On G, equation 33 gives $|\widehat{\phi} - C| \le B_1 \|\widehat{\mathbf{p}} - \mathbf{p}\|$. Thus, if rejection occurs on G, then $B_1 \|\widehat{\mathbf{p}} - \mathbf{p}\| > \epsilon$. Therefore,

$$\{|\widehat{\phi} - C| > \epsilon\} \cap G \subseteq \{B_1^2 \|\widehat{\mathbf{p}} - \mathbf{p}\|^2 > \epsilon^2\}.$$

Apply Markov's inequality equation 34 with $W = B_1^2 \|\widehat{\mathbf{p}} - \mathbf{p}\|^2$ and $a = \epsilon^2$, then use the expectation bound equation 32:

$$\begin{aligned} \Pr(\{|\widehat{\phi} - C| > \epsilon\} \cap G) &\le \Pr(B_1^2 \|\widehat{\mathbf{p}} - \mathbf{p}\|^2 > \epsilon^2) \\ &\le \frac{B_1^2 \mathbb{E}\big[\|\widehat{\mathbf{p}} - \mathbf{p}\|^2\big]}{\epsilon^2} \le \frac{AB_1^2 S}{\epsilon^2}. \end{aligned}$$

Combining this with $\Pr(G^c) \le AS/\delta^2$ from equation 35 gives

$$\begin{aligned} \Pr(|\widehat{\phi} - C| > \epsilon) &\le \Pr(G^c) + \Pr(\{|\widehat{\phi} - C| > \epsilon\} \cap G) \\ &\le \frac{AS}{\delta^2} + \frac{AB_1^2 S}{\epsilon^2}. \end{aligned} \tag{36}$$

As every $N_j \to \infty$, we have $S \to 0$. The detection rule's threshold condition $\epsilon/\sqrt{S} \to \infty$ gives

$$\frac{\epsilon}{\sqrt{S}} \to \infty \quad \Longrightarrow \quad \frac{S}{\epsilon^2} = \left(\frac{\sqrt{S}}{\epsilon}\right)^2 \to 0.$$

The constants A, B_1, and $\delta > 0$ do not depend on the query counts. Both terms in equation 36 therefore tend to zero, proving the claim. □

E.3 Proof of Theorem 3

Proof. Let $S = \{x \in \mathbb{R}^m : \phi(x) = C\}$. Since S has Lebesgue measure 0 and the Bayesian prior for $(\Gamma_1, \ldots, \Gamma_m)$ is absolutely continuous, the probability that $(\Gamma_1, \ldots, \Gamma_m) \in S$ is 0. Thus, with probability 1 under the Bayesian prior, $\phi(\Gamma_1, \ldots, \Gamma_m) \neq C$. For every such $(\Gamma_1, \ldots, \Gamma_m)$, Theorem 1(i) gives $\widehat{\phi} \to \phi(\Gamma_1, \ldots, \Gamma_m) \neq C$, so the model configuration is detected via estimation. □

F Why Strong Spoofing Is Difficult: A Toy Argument

Although a 100% long-run TPR is not formally guaranteed by this method, the following illustrative toy argument suggests a malicious provider may find it difficult to strongly spoof the tamper-detection test. We consider the Temperature Test from Theorem 8(i) for this example. To avoid making the tampering obvious, suppose that the malicious provider uses the same vocabulary as M and keeps the queries within each experiment IID. Suppose furthermore that, during experiment E_i, the probability that the first generated token returned is t is

$$p_i(t) = \frac{\exp((\widetilde{z}_i(t) + b(t))/T)}{\sum_{t' \in S_i} \exp((\widetilde{z}_i(t') + b(t'))/T)},$$

where S_i is the set of tokens retained after truncation.

Suppose that the provider strongly spoofs the Temperature Test: in the long-run limit, it passes every valid instance of the test. Choose arbitrary ρ-regular experiments E_i, E_j, E_k, E_ℓ with $A_i = A_j = (a)$, $B_i = B_j = (b)$, $A_k = A_\ell = (c)$, and $B_k = B_\ell = (d)$. By Theorem 8(i), passing this instance requires

$$\frac{(z_i(a) - z_i(b)) - (z_j(a) - z_j(b))}{\Gamma_i - \Gamma_j} = \frac{(z_k(c) - z_k(d)) - (z_\ell(c) - z_\ell(d))}{\Gamma_k - \Gamma_\ell}.$$

For experiment E_i, the probability ratio is

$$\begin{aligned}\frac{p_i(a)}{p_i(b)} &= \frac{\exp((\widetilde{z}_i(a) + b(a))/T)}{\sum_{t' \in S_i} \exp((\widetilde{z}_i(t') + b(t'))/T)} \frac{\sum_{t' \in S_i} \exp((\widetilde{z}_i(t') + b(t'))/T)}{\exp((\widetilde{z}_i(b) + b(b))/T)} \\ &= \exp\left(\frac{\widetilde{z}_i(a) - \widetilde{z}_i(b) + b(a) - b(b)}{T}\right).\end{aligned}$$

Because $A_i = (a)$ and $B_i = (b)$, it follows that

$$\Gamma_i = \ln\left(\frac{p_i(a)}{p_i(b)}\right) = \frac{\widetilde{z}_i(a) - \widetilde{z}_i(b) + b(a) - b(b)}{T}.$$

Applying the same calculation to the remaining experiments and subtracting within each experiment pair gives

$$\begin{aligned}\Gamma_i - \Gamma_j &= \frac{(\widetilde{z}_i(a) - \widetilde{z}_i(b)) - (\widetilde{z}_j(a) - \widetilde{z}_j(b))}{T}, \\ \Gamma_k - \Gamma_\ell &= \frac{(\widetilde{z}_k(c) - \widetilde{z}_k(d)) - (\widetilde{z}_\ell(c) - \widetilde{z}_\ell(d))}{T},\end{aligned}$$

where $b(a) - b(b)$ cancels from the first subtraction and $b(c) - b(d)$ cancels from the second. Substituting these expressions into the temperature-test equality gives

$$T\frac{(z_i(a) - z_i(b)) - (z_j(a) - z_j(b))}{(\widetilde{z}_i(a) - \widetilde{z}_i(b)) - (\widetilde{z}_j(a) - \widetilde{z}_j(b))} = T\frac{(z_k(c) - z_k(d)) - (z_\ell(c) - z_\ell(d))}{(\widetilde{z}_k(c) - \widetilde{z}_k(d)) - (\widetilde{z}_\ell(c) - \widetilde{z}_\ell(d))}.$$

Canceling T and taking reciprocals gives

$$\frac{(\widetilde{z}_i(a) - \widetilde{z}_i(b)) - (\widetilde{z}_j(a) - \widetilde{z}_j(b))}{(z_i(a) - z_i(b)) - (z_j(a) - z_j(b))} = \frac{(\widetilde{z}_k(c) - \widetilde{z}_k(d)) - (\widetilde{z}_\ell(c) - \widetilde{z}_\ell(d))}{(z_k(c) - z_k(d)) - (z_\ell(c) - z_\ell(d))}.$$

Strong spoofing requires the displayed equality to hold for every two valid choices of experiment pairs and token pairs. Every such ratio must therefore equal every other such ratio, so they share one common value λ. Therefore, for every covered choice of i, j, a, b,

$$(\widetilde{z}_i(a) - \widetilde{z}_i(b)) - (\widetilde{z}_j(a) - \widetilde{z}_j(b)) = \lambda\big((z_i(a) - z_i(b)) - (z_j(a) - z_j(b))\big).$$

Grouping the terms by experiment gives

$$[\widetilde{z}_i(a) - \lambda z_i(a)] - [\widetilde{z}_i(b) - \lambda z_i(b)] = [\widetilde{z}_j(a) - \lambda z_j(a)] - [\widetilde{z}_j(b) - \lambda z_j(b)].$$

Fix a reference experiment E_{i_0} and token t_0. Taking $a = t$, $b = t_0$, and $j = i_0$ gives

$$\widetilde{z}_i(t) - \lambda z_i(t) = [\widetilde{z}_i(t_0) - \lambda z_i(t_0)] + [\widetilde{z}_{i_0}(t) - \lambda z_{i_0}(t)] - [\widetilde{z}_{i_0}(t_0) - \lambda z_{i_0}(t_0)].$$

The first bracket on the right depends only on i, while the difference of the remaining two brackets depends only on t. Defining

$$f(i) := \widetilde{z}_i(t_0) - \lambda z_i(t_0), \qquad g(t) := [\widetilde{z}_{i_0}(t) - \lambda z_{i_0}(t)] - [\widetilde{z}_{i_0}(t_0) - \lambda z_{i_0}(t_0)]$$

therefore yields

$$\widetilde{z}_i(t) = \lambda z_i(t) + f(i) + g(t).$$

The functions f and g are otherwise generic: $f(i)$ depends only on the experiment index, whereas $g(t)$ depends only on the token.

For $\lambda > 0$, define $c(t) = (g(t) + b(t))/\lambda$. Then

$$\begin{aligned} p_i(t) &= \frac{\exp((\lambda z_i(t) + f(i) + g(t) + b(t))/T)}{\sum_{t' \in S_i} \exp((\lambda z_i(t') + f(i) + g(t') + b(t'))/T)} \\ &= \frac{\exp((\lambda z_i(t) + g(t) + b(t))/T)}{\sum_{t' \in S_i} \exp((\lambda z_i(t') + g(t') + b(t'))/T)} \\ &= \frac{\exp((z_i(t) + c(t))/(T/\lambda))}{\sum_{t' \in S_i} \exp((z_i(t') + c(t'))/(T/\lambda))}. \end{aligned}$$

The served distribution is therefore obtained by applying an effective temperature T/λ, a token bias $c(t)$, and some unspecified truncation rule to the expected model's logits. Under appropriate coverage assumptions on the tested tokens and regularity assumptions on the type of truncation used by the provider, spoofing every valid Temperature Test requires reproducing the expected model configuration's exact behavior for the first generated token returned up to standard sampling parameters, when $\lambda > 0$. As the old adage says, "if it walks like a duck and quacks like a duck, it is a duck." Going by this wisdom, regardless of the internals of the served model configuration, if it externally behaves identically to the expected model configuration up to standard sampling parameters, it would functionally not constitute tampering in the first place, and the malicious provider would have failed in its fundamental objective.