KBF: KNOWLEDGE BOUNDARY AS FINGERPRINT FOR LANGUAGE MODEL AND BLACK-BOX API AUDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Relay and reseller APIs mediate access to large language models (LLMs), but users cannot directly verify which model serves them. We introduce KBF, a black-box auditing protocol based on stable factual recall near the knowledge boundary, including repeatable wrong answers. KBF generates benign, renewable probes and calibrates audit decisions against reference self-variation. Across 16 production endpoints, KBF detects all 155 economically relevant substitutions without rejecting any of the 16 same-reference controls. KBF remains robust to deployment variation and reaches 95% TPR at a substitution rate as low as 15% in mixed-routing simulations. Field audits flag 7 of 28 endpoints across six platforms as statistically inconsistent with their references. After reference enrollment, even GPT-6 Astra costs only approximately $0.67 per online audit at the recorded API prices.

1 INTRODUCTION

Relay and reseller APIs commonly provide access to large language models (LLMs). Users pay a third-party relay for an advertised model, while the relay controls which upstream endpoint serves each request. Users cannot directly verify which model serves them. The price gap between flagship and budget models creates an incentive for silent substitution. A relay can reduce its serving costs by routing some or all requests to a cheaper backend while continuing to charge for the advertised model. Measurement studies have reported deceptive model claims in shadow APIs (Zhang et al., 2026c) and malicious intermediaries in the LLM supply chain (Liu et al., 2026).

This motivates a technical question. Given only black-box access to an official reference API and a suspect relay API, can an auditor detect model substitution? The auditor can send prompts to these endpoints and read their responses, but methods requiring model internals or provider cooperation fall outside this setting (Zhu et al., 2026; Yang & Wu, 2024; Sun et al., 2024). The audit must remain robust to legitimate deployment variations, including changes to system prompts, decoding settings, and serving backends, among others.

A useful audit protocol should satisfy four requirements. (1) **Accuracy** requires detecting economically meaningful substitutions while keeping false-positive rates low, since false alarms can implicate honest relays. (2) **Robustness to deployment variation** requires reliable audit decisions despite benign differences in system prompts, temperature, output limits, retrieval wrappers, and serving backends. (3) **Benign and renewable probes** should resemble ordinary user requests and be inexpensive to regenerate, since a relay can filter recognizable probes or selectively forward them to the genuine backend. (4) **Economic practicality** requires low probe-generation and online audit costs to support repeated auditing.

Table 1: Black-box auditors compared on accuracy (Acc.), robustness to deployment variation (Rob.), benign and renewable probes (Ben.), and economic practicality (Econ.). — denotes an untested setting.

Method	Acc.	Rob.	Ben.	Econ.
KBF (Ours)	✓	✓	✓	✓
LLMmap (Pasquini et al., 2025)	✗	✗	✗	✓
MET (Gao et al., 2024)	✓	✗	✓	✗
ZeroPrint (Shao et al., 2026)	✗	✗	✓	✗
IRIS (Zhang et al., 2026d)	✓	✗	✗	✓
GateScope (Lin et al., 2026)	✓	—	✗	✗

Existing black-box auditors do not yet meet these requirements jointly in our setting (Table 1). Fixed probes can be recognized and handled selectively (Pasquini et al., 2025; Lin et al., 2026). Output distributions and response-based fingerprints can also change with prompts, decoding, or serving implementations, complicating substitution detection (Gao et al., 2024; Shao et al., 2026; Zhang et al., 2026d). Repeated sampling and costly probes further limit frequent auditing (Gao et al., 2024;

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.

Lin et al., 2026). These limitations motivate an audit based on renewable, ordinary-looking probes whose reference behavior remains stable under benign deployment variation.

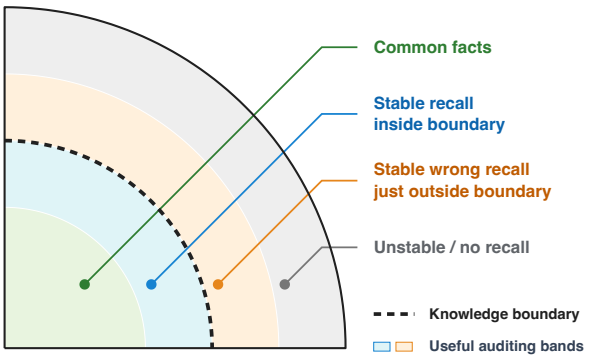


Figure 1: Knowledge-boundary recall as an audit signal. KBF retains stable reference answers, including repeatable wrong answers just outside the boundary.

Our insight is that *factual recall near the knowledge boundary can provide both reference stability and model discrimination* (Figure 1). Models often share common facts, while obscure queries can make the reference inconsistent. Between these extremes, the reference may repeat a specific answer that other endpoints do not reproduce. This includes stable factual errors, which remain useful because the audit tests agreement with the reference rather than correctness.

We design *Knowledge-Boundary Fingerprinting* (KBF), a black-box relay-auditing protocol based on this signal. KBF has three stages (Figure 2). (1) During offline probe generation, the auditor generates candidate probes through the official reference API and screens them for stable reference answers and, when useful, disagreement with a cheaper contrast model. (2) During reference self-calibration, the auditor re-queries retained probes and derives an upper confidence bound on reference self-disagreement. (3) During the online audit, KBF compares suspect responses with the stored reference answers under domain-specific matching rules and applies a one-sided binomial test against the calibrated bound. Probes can be refreshed per reference version.

KBF combines accurate substitution detection and robustness to deployment variation with benign, renewable probes and low audit costs. Across 16 production LLM endpoints from eight model families, KBF detects all 155 economically relevant substitutions without rejecting any of the 16 same-reference controls (Section 5.2). Under six deployment configurations on five representative models, it detects all 60 substitutions with no false positives in 30 same-reference trials, whereas none of the four evaluated baselines achieves both outcomes (Section 5.3). In our latest evaluation, even GPT-6 Astra requires only approximately $0.67 per audit at the recorded OpenRouter prices (Section 5.2). In mixed-routing simulations, our routing method reaches 95% TPR at a substitution rate as low as 15% (Section 5.5). In the six-platform field-study subset, KBF flags 7 of 28 endpoints as inconsistent with their references, predominantly for premium proprietary models, supporting further investigation of real relay APIs (Section 5.6).

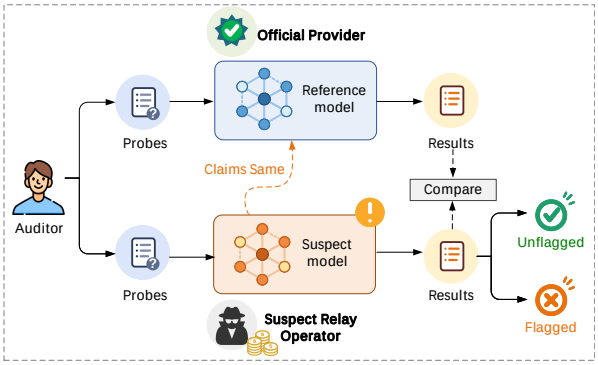


Figure 2: Black-box audit flow for KBF. The auditor compares a claimed reference endpoint with a suspect relay endpoint using only public API access.

Our contributions are as follows.

1. **Knowledge-boundary auditing.** We develop KBF, a black-box protocol based on stable factual recall near the knowledge boundary, including repeatable wrong answers. KBF combines accurate, robust detection with benign, renewable probes and low audit costs for practical auditing of real-world APIs.
2. **Systematic evaluation.** We compare KBF with existing auditors across production models and deployment settings, characterize the stability and discrimination of boundary recall, and use ablations to examine the contributions of probe generation and screening.
3. **Mixed routing and real-world audits.** We demonstrate partial-substitution detection and routing-fraction estimation under the evaluated routing model. Real relay audits reveal endpoints statistically inconsistent with their references, predominantly for premium proprietary models.
4. **Practical tools and open artifacts.** We provide ready-to-use scripts for probe generation and API auditing, with reference probe sets, raw data, and scripts to reproduce the results at `https://anonymous.4open.science/r/KBF-4321`.

2 Related Work

Auditing black-box LLM APIs. Existing API auditors use behavioral signatures and statistical tests to check model claims. LLMmap and GateScope identify enrolled models from responses to fixed probe sets using learned representations or classifiers (Pasquini et al., 2025; Lin et al., 2026). MET compares completion distributions from two endpoints with a two-sample test (Gao et al., 2024). RUT tests whether API responses are consistent with a reference distribution through rank-based scores, requiring local access to the reference model (Zhu et al., 2026). ZeroPrint measures how response embeddings change under input perturbations and compares these local-sensitivity fingerprints (Shao et al., 2026). IRIS uses random-generation bias to detect model substitution and partial routing, estimate the routed fraction, and select an audit budget (Zhang et al., 2026d). For relay auditing, the comparison must account for benign deployment variation between the official reference and an honest relay. KBF uses stable factual recall near the knowledge boundary of a claimed reference endpoint, with a binomial decision rule calibrated on reference self-variation. Appendix G covers baseline configurations, detailed comparisons, and approaches requiring stronger access or provider cooperation.

Knowledge boundaries. Knowledge-boundary research studies which facts a model can recall reliably and how it should respond when reliable recall fails (Li et al., 2025; Deng et al., 2025; Zhou et al., 2026). Concurrent work on Incompressible Knowledge Probes (IKP) uses factual recall and shared incorrect answers to estimate model capacity and examine model relationships (Li, 2026). IKP uses a common question bank to characterize models. KBF generates probes for a claimed reference and uses them to audit endpoint consistency. Its probes can be regenerated privately and are screened for stable responses across reference configurations. Both approaches independently exploit model-specific factual recall, including repeatable errors, as an observable signal. Our audit tests whether a suspect reproduces this behavior under the evaluated deployment variations.

3 Problem Setting

We study whether a third-party relay API serves the model it advertises.

Endpoints. The auditor has black-box access to two randomized oracles, the official reference $\mathcal{O}_{\mathrm{ref}}(x, c)$ and suspect relay $\mathcal{O}_{\mathrm{sus}}(x, c)$, where x is a user query and $c \in \mathcal{C}$ is a requested deployment configuration (Figure 2). We assume the reference applies these requests up to ordinary provider-side nondeterminism. The auditor requests the same configurations from the suspect but cannot assume compliance. The auditor has no access to weights, logits, training data, inference code, provider metadata, or routing logs. KBF fingerprints the served API endpoint, not isolated parametric weights.

Auditing goal. A protocol Audit adaptively queries both endpoints and outputs SAME or DIFFERENT. The hypotheses are

$$H_0 : \mathcal{O}_{\mathrm{sus}} \approx \mathcal{O}_{\mathrm{ref}} \quad \text{vs.} \quad H_1 : \mathcal{O}_{\mathrm{sus}} \not\approx \mathcal{O}_{\mathrm{ref}}.$$

Here $\approx$ is operational: the suspect is consistent with the reference only if its audited behavior remains within the measured reference self-variation (including reasonable variation due to wrappers, temperature, and backend deployment). Appendix A.1 describes the audit scope and non-goals.

Relay capabilities. The adversary is the suspect relay operator. It controls backend selection, decoding parameters, hidden prompts, wrappers, and quantization. Mixed routing gives

$$\mathcal{O}_{\mathrm{sus}} = \begin{cases} \mathcal{O}_{\mathrm{ref}} & \text{with probability } 1 - \pi, \\ \mathcal{O}_{\mathrm{sub}} & \text{with probability } \pi. \end{cases}$$

Here $\mathcal{O}_{\mathrm{sub}}$ is a substitute backend and $\pi \in [0, 1]$ is the substitution rate, with $\pi = 1$ giving full substitution. Routing may depend on the query, requested configuration, load, user account, or request timing. The adversary may know the algorithm and public probe sets and query the official reference endpoint. It cannot control that endpoint, the auditor's local computation, or auditor–API communication. KBF does not provide cryptographic security against perfect probe recognition.

Evaluation criteria. We measure the false-positive rate (FPR) and true-positive rate (TPR),

$$\mathrm{FPR} = \Pr[\mathsf{Audit}(\mathcal{O}_{\mathrm{ref}}, \mathcal{O}_{\mathrm{sus}}) = \textsc{Different} \mid H_0],$$
$$\mathrm{TPR} = \Pr[\mathsf{Audit}(\mathcal{O}_{\mathrm{ref}}, \mathcal{O}_{\mathrm{sus}}) = \textsc{Different} \mid H_1].$$

FPR is the primary safety metric because a false accusation against an honest provider is costly. For mixed routing, TPR depends on the substitution rate π and the substitute model. Robustness requires low reference self-noise across repeated queries and configurations.

4 METHOD

4.1 BOUNDARY RECALL AS AN AUDIT SIGNAL

Factual recall near the knowledge boundary can remain stable for the reference while differing across models. Interior probes concern common facts shared across models. Beyond-boundary probes are too obscure for the reference to answer consistently. Between these extremes, boundary probes elicit answers that the reference repeats reliably but other models reproduce less often.

KBF generates and screens boundary probes as described in Section 4.2. To examine this signal, we collect all three probe types for each of four reference models and audit every model on each pool. Figure 3 compares their agreement with recorded reference answers, showing how boundary probes combine reference stability with model discrimination. Stable factual errors can also contribute because the audit tests reference agreement rather than correctness.

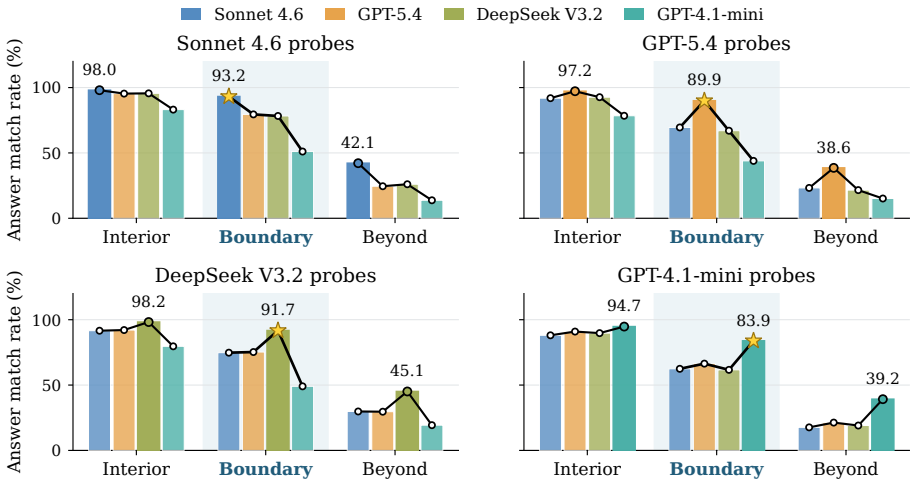


Figure 3: Match rates by reference probe set. Dark bars and labels show reference self-evaluation, lines connect models within groups, and stars mark boundary peaks. Rates compare separate post-construction responses with recorded reference answers, excluding unscored responses.

Reference stability. Boundary probes retain 83.86%–93.15% reference agreement on fresh queries, providing a reproducible audit baseline. Interior recall is stable, but other models match these answers at 78.43%–95.58%, limiting model discrimination. Repeatability alone is insufficient for auditing.

Model discrimination. Other models reproduce boundary answers at rates of 43.91%–79.48%, below the corresponding reference in all 12 directed model pairs. Beyond-boundary probes also show cross-model differences, but reference agreement drops to 38.57%–45.11%, making those differences harder to separate from reference inconsistency.

4.2 KNOWLEDGE BOUNDARY FINGERPRINTING

Stage I. Reference probe generation. Our numerical-probe implementation generates candidates independently within domains. Each domain d specifies a prompt template, a valid numerical range, and a comparison rule match_d. The same match_d applies to reference-consistency checks, self-calibration, and auditing. Unparseable or out-of-range values are invalid.

For each domain, KBF generates candidates over rounds indexed by t. Each round asks the reference endpoint to propose structured candidate records, typically as `name | value` pairs. The prompt specifies a domain theme and a short exclusion list of previously proposed names. KBF parses the proposed records, removes duplicate names, rejects values outside the domain range, and renders the remaining records into short audit prompts q_i.

Reference-consistency checks $\mathcal{C}_{\text{cons}}$ re-query each candidate under benign configuration changes, such as prompt variants or decoding settings. For each prompt q_i in domain d_i, KBF parses the responses into domain-valid values, denoted by A_i. The candidate survives only when all responses are valid and agree under match_{d_i}, and KBF records their consensus value as a_i. The fingerprint stores the N surviving probes as

$$P_{\text{ref}} = \{(q_i, d_i, a_i, \mathsf{match}_{d_i})\}_{i=1}^{N}.$$

KBF tracks the number of surviving probes per round and stops searching a domain when further rounds yield no stable probes.

Stage II. Reference self-calibration. KBF fixes P_{ref} and re-queries the reference endpoint under the audit configuration c_{audit}, yielding parsed values $\tilde{a}_i$. Missing, unparseable, and out-of-range answers are excluded from scoring. For any domain-valid returned value v, define the discrepancy indicator

$$\delta_i(v) = \begin{cases} 0, & \text{if } \mathsf{match}_{d_i}(a_i, v) = 1, \\ 1, & \text{otherwise.} \end{cases}$$

Let n_{self} count valid reference responses and k_{self} their discrepancies. KBF calibrates the null bound as

$$p_0 = \text{CP}_{99}(k_{\text{self}}, n_{\text{self}}).$$

Here CP_{99} is the one-sided 99% Clopper–Pearson upper confidence bound on the reference discrepancy rate (Clopper & Pearson, 1934). We use this confidence level, $\gamma = 0.99$, unless stated otherwise. The fingerprint passed to online auditing is $F_{\text{ref}} = (P_{\text{ref}}, k_{\text{self}}, p_0)$. The binomial model treats scored probes as independent discrepancy trials with a common reference error rate.

Stage III. Suspect endpoint audit. KBF sends each enrolled prompt to the suspect endpoint $\mathcal{O}_{\text{sus}}$ under c_{audit} and parses the returned value $\hat{a}_i$. For each valid response, KBF records $Z_i = \delta_i(\hat{a}_i)$ and counts discrepancies as k. The audit denominator n counts valid responses, excluding missing, unparseable, and out-of-range answers. Under the binomial model, let p_{sus} denote the suspect discrepancy probability on the scored probes. KBF tests whether it exceeds the calibrated reference bound,

$$H_0 : p_{\text{sus}} \le p_0, \qquad H_1 : p_{\text{sus}} > p_0.$$

KBF computes the one-sided probability of at least k discrepancies under the null bound,

$$p_{\text{audit}} = \Pr(X \ge k), \qquad X \sim \text{Binomial}(n, p_0).$$

If $p_{\text{audit}} < \alpha = 0.05$, KBF rejects H_0 and reports DIFFERENT. Otherwise, it reports SAME, meaning the audit did not find enough discrepancies to exceed the reference-noise bound.

Appendix A provides the full procedures and implementation details for all three stages and discusses the implications of probe dependence. In our ablations, specialist generation yields more retained probes, stability filtering reduces reference noise, and contrast filtering improves model discrimination (Appendix H).

4.3 PARTIAL-ROUTING EXTENSION

Each API request contains up to ten probes from one domain. The relay independently routes the entire request to substitute S with probability π, or to the reference otherwise. KBF fixes the reference-matched scoring set $\mathcal{I}_0 = \{i \mid \tilde{a}_i \text{ is valid and } \delta_i(\tilde{a}_i) = 0\}$ and preserves the original request groups. Let B count groups with scored probes, A_b be the mismatch fraction from a subsequent reference check, and Y_{b1}, Y_{b2} the fractions from two online rounds. Each batch contributes one observation, allowing arbitrary dependence among its scored probes.

Paired detection. We use two audit rounds to strengthen the detection signal, with one-round results reported in Appendix E. KBF queries each batch twice and compares both responses with the stored reference check using

$$T = \sum_{b=1}^{B} \left[\text{sgn}(Y_{b1} - A_b) + \text{sgn}(Y_{b2} - A_b)\right].$$

It computes the null distribution of T by uniformly reassigning the reference label among the three records within each batch. Both online rounds share one reference record and yield one joint decision. KBF rejects when the upper-tail probability is below 0.05. Validity requires independent batch groups and exchangeable reference and honest-relay records within each group. Adjacent reference checks assess temporal variation.

Routing-fraction estimate. Let q_{Sb} be the mismatch fraction of a fixed substitute fingerprint on batch b. Write $\bar{Y}_b = (Y_{b1} + Y_{b2})/2$, $d_b = \bar{Y}_b - A_b$, and $g_b = q_{Sb} - A_b$, with bars denoting averages over batches. Here d_b measures the observed increase over the reference, and g_b the increase under full substitution. Under stable reference means, $\bar{d} \approx \pi \bar{g}$, giving the point estimate

$$\widehat{\pi}_S = \text{clip}_{[0,1]}\left(\frac{\bar{d}}{\bar{g}}\right), \qquad \bar{g} > 0.$$

Clipping restricts the estimate to $[0, 1]$, and small mismatch gaps make the estimate sensitive to sampling variation.

Unknown substitute. For each candidate $S \in \mathcal{C}$, KBF recomputes $\bar{g}$ from its fingerprint and applies the same estimator to obtain $\widehat{\pi}_S$. When all candidate mean mismatch gaps are positive, it reports

$$\widehat{\Pi}_{\mathcal{C}} = \left[\min_{S \in \mathcal{C}} \widehat{\pi}_S,\ \max_{S \in \mathcal{C}} \widehat{\pi}_S \right].$$

This range reflects uncertainty about substitute identity and is not a confidence interval. Its interpretation requires the true substitute to belong to the pool. Content-dependent or audit-aware routing can prevent these estimates from representing the traffic-wide substitution fraction.

5 Experiments

We evaluate substitution detection, enrollment and audit cost, robustness to deployment and probe-format changes, partial routing, and real-world relay endpoints.

5.1 Experimental Setup

We evaluate KBF in controlled settings where the claimed endpoint, provider route, probe set, and decision rule are fixed. This design isolates model behavior from avoidable routing noise and provides a baseline for the field audit.

Models. Table 2 summarizes the endpoint pool. We evaluate 16 production LLM API endpoints across eight model families and three price tiers, comprising six T1 flagship, five T2 mid-range, and five T3 budget endpoints. OpenRouter (OpenRouter, 2026) input prices recorded in March 2026 range from \$0.05 to \$5.00 per million tokens.

Reference endpoints. For each claimed model, we enroll the reference fingerprint through OpenRouter with the provider fixed to the entry in Table 2 and fallback routing disabled.

Baselines. We compare KBF with five black-box auditors using their released implementations. In Section 5.3, KBF and the evaluated baselines use shared deployment configurations. The baselines are LLMmap (Pasquini et al., 2025), MET (Gao et al., 2024), ZeroPrint (Shao et al., 2026), IRIS (Zhang et al., 2026d), and GateScope (Lin et al., 2026). Appendix G.1 describes the implementations, audit adaptations, and evaluation coverage.

Metrics. We count false positives (FP, lower is better), substitution alarms in same-reference trials, and true positives (TP, higher is better), correctly detected substitutions.

Table 2: The 16 evaluated LLM endpoints. *Provider* identifies the pinned OpenRouter route. *#Probes* counts retained numerical probes. *Self-Err* is the reference calibration self-disagreement rate, whose CP_{99} upper bound defines p_0. The diagonal of Figure 4 reports independent same-reference retests. *Input/output prices are in USD per million tokens, recorded from OpenRouter in March 2026.

Tier	Model	Model Family	Model Provider	Price* ($/M Tokens)		#Probes	Self-Err
				Input	Output		
T1	Claude Opus 4.6	Anthropic	Amazon Bedrock	5.00	25.00	681	4.3%
T1	Claude Sonnet 4.6	Anthropic	Google	3.00	15.00	224	1.3%
T1	GPT-5.4	OpenAI	OpenAI	2.50	10.00	317	1.6%
T1	Gemini 3 Flash	Google	Google	0.50	2.50	315	2.2%
T1	GLM-5	Z.AI	Z.AI	0.72	2.20	405	4.2%
T1	Qwen3.5-397B-A17B	Alibaba	Alibaba	0.39	1.20	243	1.7%
T2	DeepSeek-V3.2	DeepSeek	Google	0.26	0.42	364	3.3%
T2	GPT-4.1-mini	OpenAI	OpenAI	0.40	1.60	134	6.0%
T2	GLM-4.7	Z.AI	Z.AI	0.38	2.00	346	4.6%
T2	Kimi-K2-0905	Moonshot	Moonshot AI	0.40	2.50	300	4.7%
T2	Qwen3.5-27B	Alibaba	Alibaba	0.20	0.30	115	4.3%
T3	GPT-4.1-nano	OpenAI	OpenAI	0.10	0.40	109	7.3%
T3	LLaMA-4-Scout	Meta	Groq	0.08	0.30	146	11.7%
T3	Qwen3.5-9B	Alibaba	Together	0.05	0.10	105	3.8%
T3	GLM-4.7-Flash	Z.AI	DeepInfra	0.06	0.20	309	16.2%
T3	Gemini 2.5 Flash Lite	Google	Google	0.10	0.40	210	13.8%

5.2 Detection Accuracy

We audit all pairs in the 16-model pool. KBF flags all **155 economically relevant substitutions** at $p < 0.05$ under the CP_{99} binomial test, while producing no false-positive rejection in the 16 same-reference trials.

Economically Relevant Pairs. A pair (M_r, M_t) is economically relevant when target endpoint M_t is cheaper than claimed reference M_r or both are in the same price tier. Lower-tier substitutions

capture direct cost savings. Same-tier substitutions capture smaller incentives from within-tier price differences, easier access, compliance burden, and regional availability. We exclude only cross-tier upgrades to a more expensive target, such as T2→T1 and T3→T2/T1, from the threat model and TPR/FPR estimates.

Substitution Detection. Figure 4 shows the detection matrix, with white off-diagonal cells marking economically irrelevant upgrade directions. KBF detects all **12/12** within-family economically relevant substitutions and all **143/143** cross-family substitutions, showing that the signal extends beyond visibly different model families. KBF also detects most upgrade substitutions. In September 2026, we generated fresh probe sets for an expanded model pool that includes newly released models (Appendix C).

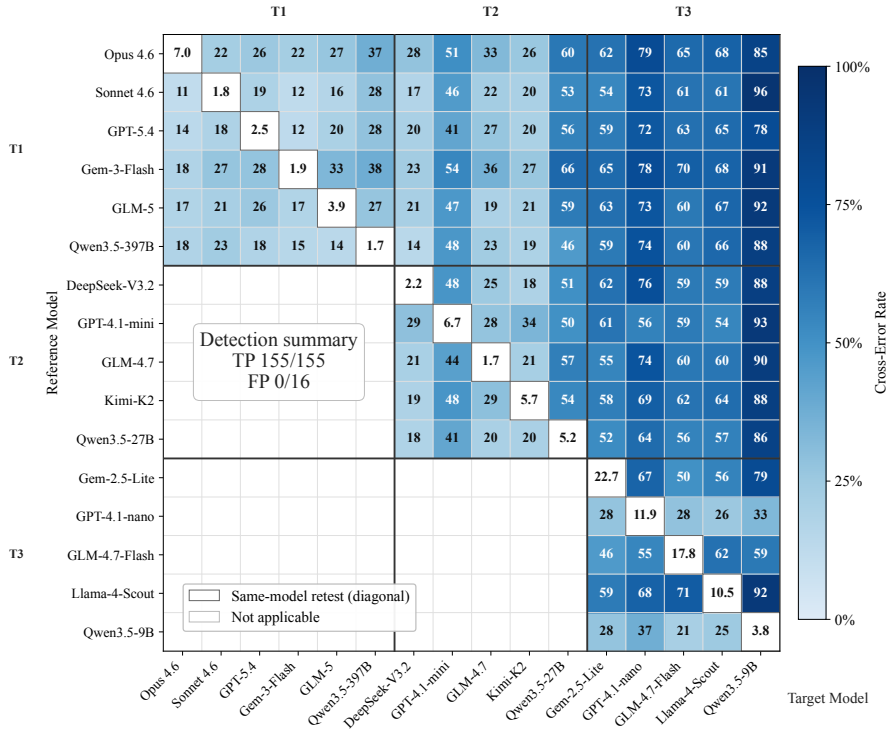


Figure 4: Detection heatmap over 16×16 model pairs. Color indicates the mismatch rate on the reference probe set. Diagonal cells show independent same-reference retests evaluated against their calibration bounds. White off-diagonal cells denote economically irrelevant upgrade directions.

False-Positive Control. KBF produces 0/16 false positives in independent same-reference trials, with baseline results in Table 3. New tests in September 2026 also yield 0/28 false positives on the expanded model pool with freshly enrolled probes (Appendix C.2). Section 5.3 and Appendix D report false-positive control results for KBF under deployment variation and temporal retesting.

Table 3: Same-reference false positives.

Method	FP	FPR
KBF (ours)	**0 / 16**	**0.0%**
LLMmap (Pasquini et al., 2025)	5 / 16	31.3%
MET (Gao et al., 2024)	1 / 16	6.3%
ZeroPrint (Shao et al., 2026)	1 / 16	6.3%
IRIS (Zhang et al., 2026d)	4 / 16	25.0%
GateScope* (Lin et al., 2026)	0 / 12	0.0%

*GateScope was evaluated on a 12-model subset.

Detection Cost. (1) **Probe generation.** A separate May 2026 run enrolled 3,696 probes for six T1 references for $7.07 (Appendix C.3). Probes can be reused across audits of the same reference version. (2) **Online audit.** At March 2026 API prices, Claude Opus 4.6 has the highest single-audit cost among the original 16 models, at $0.24. Among the 13 models added in September 2026, GPT-6 Astra is the most expensive at $0.67 per audit at the recorded prices. Both figures exclude enrollment.

5.3 Robustness to Deployment Variation

Relay APIs often serve application endpoints with role prompts, nonzero temperatures, retrieval context, or response wrappers for product workflows. An audit should preserve accuracy under these wrappers, maintaining 0% FPR in same-reference trials and 100% TPR in substitution trials.

Cross-Method Stress Test. We compare four baselines on Claude Sonnet 4.6 (T1), GPT-5.4 (T1), GLM-4.7 (T2), GPT-4.1-mini (T2), and Qwen3.5-9B (T3), yielding 5 same-reference setups and 10 economically relevant substitution pairs. KBF and all baselines use six deployment variants with shared role prompts, temperatures, and RAG settings. Each method undergoes 30 same-reference and 60 substitution trials.

Table 4 shows that **KBF separates honest endpoints from substitutions across all six tested deployment variants**, while the baselines raise false alarms and, in some cases, miss substitutions. The reduced accuracy may reflect the sensitivity of baseline detection methods to temperature and prompt-wrapping changes relative to their reference configurations. Additional tests cover adversarial prompts, identity spoofing, quantization, provider variation, temporal drift, threshold choice, agent interfaces, and deployment variation across the full model pool. KBF remains stable in most tested settings (Appendix D).

Table 4: Configuration-robustness comparison on five representative models under six shared deployment wrappers.

ID	Role	Temp	RAG	KBF (ours)		MET		LLMmap		ZeroPrint		IRIS	
				FP	TP	FP	TP	FP	TP	FP	TP	FP	TP
C0	No role prompt	0	No	**0/5**	**10/10**	4/5	10/10	3/5	6/10	4/5	7/10	0/5	8/10
C1	Finance compliance	0.2	No	**0/5**	**10/10**	5/5	10/10	3/5	10/10	5/5	7/10	1/5	8/10
C2	Enterprise policy	0	Yes	**0/5**	**10/10**	5/5	10/10	3/5	8/10	4/5	7/10	1/5	8/10
C3	Customer support	0.3	No	**0/5**	**10/10**	5/5	10/10	3/5	7/10	3/5	8/10	1/5	8/10
C4	Medical triage	0.1	Yes	**0/5**	**10/10**	5/5	10/10	4/5	7/10	3/5	8/10	1/5	8/10
C5	Academic writing	0.7	No	**0/5**	**10/10**	5/5	10/10	3/5	8/10	4/5	8/10	2/5	8/10
Total				**0/30**	**60/60**	29/30	60/60	19/30	46/60	23/30	45/60	6/30	48/60

5.4 Probe Format Variation

KBF relies on the knowledge-boundary questions retained by screening rather than on their presentation format. Reformulating KBF probes reduces dependence on a fixed request format. We evaluate four variants that change the answer type or remove cloze blanks, fixed system prompts and answer-format rules, or batching (Table 5). All four detect every tested substitution without false positives, showing that detection remains effective across these changes in request format. KBF can renew probe domains, questions, and queried numerical facts through fresh generation and screening. Starting from questions retained by KBF screening, we use a low-cost LLM, GPT-4.1-mini, to rewrite questions, verify semantic equivalence, and extract answers from free-text replies. This allows the screened probes to take many surface forms while preserving their factual content. Appendix B gives the results and request formats.

Table 5: Probe features. ● = retained, ○ = removed. Number denotes numerical answers.

Variant	Number	Cloze	Prompt	Batch
Entity cloze	○	●	●	●
Numerical QA	●	○	●	●
Natural batch QA	●	○	○	●
Single-question QA	●	○	○	○

5.5 Partial-Routing Detection and Estimation

We evaluate eight reference–substitute pairs using the paired batch method in Section 4.3. Each request contains up to ten probes and selects one backend for the entire batch, independently across requests and rounds. We replay two-round audits using archived responses, measuring routing variability conditional on these records.

Main Results. Two rounds reach 95% TPR at routing fractions of 22–55% for same-class substitutes and 15–34% for budget substitutes (Figure 5). Additional experiments show that a second audit round improves detection and yields more precise routing-fraction estimates for known substitutes (Appendix E).

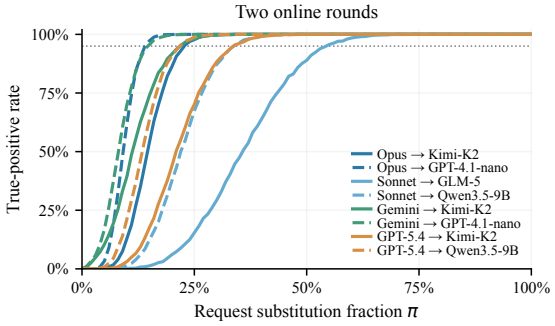


Figure 5: Two-round routing detection for eight pairs at $\alpha = 0.05$. Solid and dashed curves denote same-class and budget substitutes. The dotted line marks 95% TPR.

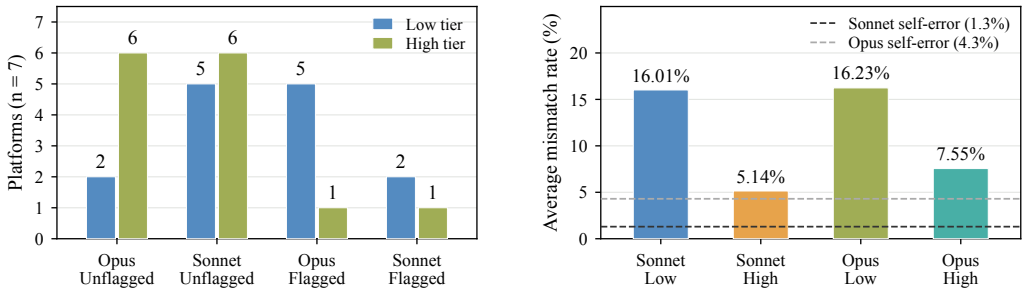
5.6 Real-World Relay Audits

We audit eleven relay API platforms, anonymized as Platform 1–11, for statistical inconsistency with the reference probe-response distributions. Table 6 covers 28 available endpoints across six platforms. Across all eleven platforms, flagged endpoints remain concentrated among Claude models (Appendix F).

Table 6: Field audit of six anonymized relay platforms. Entries report mismatch rates, with ✓ for no rejection, × for rejection ($p < 0.05$), and **?** near the decision threshold. Parentheses give reference self-error. Dashes denote unavailable endpoints. The Opus entry for Platform 5 uses the default tier.

Model (self-err)	P1	P2	P3	P4	P5	P11	Flagged/tested
Claude Opus 4.6 (4.3%)	× 19.7%	**?** 8.2%	—	✓ 4.3%	× 47.9%	**?** 9.0%	**4 / 5**
Claude Sonnet 4.6 (1.3%)	× 13.4%	✓ 2.8%	✓ 2.7%	✓ 2.2%	× 43.8%	✓ 5.4%	**2 / 6**
GLM-4.7 (4.6%)	✓ 6.8%	× 17.4%	—	✓ 8.4%	✓ 8.7%	—	**1 / 4**
GPT-5.4 (1.6%)	✓ 3.2%	✓ 4.4%	✓ 0.9%	✓ 2.2%	✓ 1.6%	✓ 3.5%	**0 / 6**
DeepSeek-V3.2 (3.3%)	✓ 8.3%	✓ 3.3%	—	✓ 3.0%	✓ 3.6%	—	**0 / 4**
Gemini 3 Flash (2.2%)	✓ 2.5%	✓ 5.7%	—	✓ 2.5%	—	—	**0 / 3**
Flagged / tested	2/6	2/6	0/2	0/6	2/5	1/3	**7 / 28 (25%)**

Flagged Endpoints. KBF flags **7/28** endpoints in the displayed subset, with **6/7** flags on Claude endpoints and one on GLM-4.7. GPT-5.4, DeepSeek-V3.2, and Gemini 3 Flash are unflagged wherever available in the displayed subset. We test KBF across deployment configurations, provider routes, and time (Appendix D), but determining the causes of response-distribution inconsistencies observed at real-world API endpoints requires further independent evidence.



(a) Audit verdicts

(b) Average mismatch rates

Figure 6: Pricing-tier comparison for Claude Opus 4.6 and Sonnet 4.6 across seven platforms. Dashed lines in (b) mark reference self-error rates.

Tier Case Study. Seven platforms expose Claude Opus 4.6 and Claude Sonnet 4.6 under both low and upgraded pricing tiers. The low versus upgraded tiers are flagged on 5/7 versus 1/7 platforms for Opus and 2/7 versus 1/7 for Sonnet (Figure 6a). Average mismatch rates are about 16% on the low tiers and 5%–8% on the upgraded tiers (Figure 6b). For the same advertised models and platforms, low-tier responses deviate more from the reference on average.

Audit Interpretation. KBF is a statistical screening method whose flags indicate inconsistency with the reference response distribution on the audited probes.

6 Conclusion

We study model-identity auditing for relay and reseller APIs under black-box access. We introduced KBF, which fingerprints endpoints using stable factual recall near the knowledge boundary of a reference model and compares suspect endpoints with the enrolled reference using a CP binomial test. Across 16 production endpoints and eight model families, KBF detected all 155 economically relevant substitutions with no false-positive rejection on same-reference controls. It remained robust under deployment-wrapper changes and detected mixed routing at economically meaningful substitution rates. In real-world shadow API audits, KBF found 7/28 endpoints statistically inconsistent with their references, with inconsistencies concentrated on premium Claude endpoints.

AI USE STATEMENT

Generative models serve as both the objects of study and components of the experimental protocol. KBF uses reference LLMs to generate candidate numerical probes near their knowledge boundaries and uses Qwen3.5-9B for optional contrastive screening. The natural-QA experiment uses GPT-4.1-mini to rewrite probes, verify semantic equivalence, and extract numerical answers from free-text responses. Responses collected from the reference and audited endpoints form the experimental observations used for calibration and evaluation. Section 4.2 and Appendix A describe this protocol, and Section 5.1 reports the evaluated endpoints and provider settings.

AI agents supported experimental execution, primarily by writing and debugging API request scripts, running experiments, inspecting API responses, and handling request errors and rate limits. They also assisted with processing the experimental data, writing analysis scripts, and generating plots.

AI assistants helped with grammatical polishing and sentence-level editing of the manuscript. They also helped draft descriptions of the experiments and their results in the experimental sections. The authors wrote the introduction and core method sections themselves, without AI drafting assistance. For the literature review, AI assistants were used only to search for and summarize related work. The authors read the original papers and checked the summaries against them.

The authors developed the core research ideas and method, determined the argument and organization of the paper, and interpreted and validated the results and conclusions. The authors reviewed all AI-assisted work, tested the code, checked the collected data and analyses, and verified the reported results. They take full responsibility for the final manuscript, all results and conclusions, and accompanying artifacts.

ETHICS STATEMENT

This study audits publicly accessible LLM APIs using benign factual queries at low request volumes to limit the impact on service operation. It involves no human participants and collects no personal information. We de-identify the audited relay platforms and evaluate false positives and robustness under deployment variation. An audit rejection indicates inconsistency with the reference endpoint under the tested conditions.

REPRODUCIBILITY STATEMENT

We describe probe construction, matching rules, calibration, and audit decisions. Section 5.1 reports the evaluated endpoints and provider settings, and Appendix G.1 documents the baseline configurations and adaptations. Our anonymous repository (`https://anonymous.4open.science/r/KBF-4321`) provides reference probes, recorded responses, collection scripts, and offline analysis scripts for checking the reported results. Fresh data collection uses API access and records the model and serving configuration. The field-audit collectors use configurable relay addresses and credentials, represented by placeholders in the archive.

REFERENCES

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX security symposium (USENIX Security 18)*, pp. 1615–1631, 2018.

Tomas Bruckner. One token is enough: Fingerprinting and verifying large language models from single-token output distributions. *arXiv preprint arXiv:2607.10252*, 2026.

Timothée Chauvin, Clément Lalanne, Erwan Le Merrer, Jean-Michel Loubes, François Taïani, and Gilles Tredan. Token-efficient change detection in llm apis. *arXiv preprint arXiv:2602.11083*, 2026a.

Timothée Chauvin, Erwan Le Merrer, François Taïani, and Gilles Tredan. Log probability tracking of llm apis. In *International Conference on Learning Representations*, volume 2026, pp. 149447–149465, 2026b.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823, 2024.

Yang Deng, Moxin Li, Liang Pang, Wenxuan Zhang, and Wai Lam. Unveiling knowledge boundary of large language models for trustworthy information access. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 4086–4089, 2025.

Matthew Finlayson, Xiang Ren, and Swabha Swayamdipta. Every language model has a forgery-resistant signature. *arXiv preprint arXiv:2510.14086*, 2025.

Zhiyuan Fu, Junfan Chen, Lan Zhang, Ting Yang, Jun Niu, Hongyu Sun, Ruidong Li, Peng Liu, Jice Wang, Fannv He, et al. Fdllm: A dedicated detector for black-box llms fingerprinting. In *2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1374–1379. IEEE, 2025.

Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this api serving? *arXiv preprint arXiv:2410.20247*, 2024.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. Robust llm fingerprinting via domain-specific watermarks. In *ICML 2025 Workshop on Reliable and Responsible Foundation Models*, 2025.

Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. Trap: Targeted random adversarial prompt honeypot for black-box identification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11496–11517, 2024.

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. *arXiv preprint arXiv:2602.22700*, 2026.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. Semstamp: A semantic watermark with paraphrastic robustness for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4067–4082, 2024.

Yuepeng Hu, Zhengyuan Jiang, Mengyuan Li, Osama Ahmed, Zhicong Huang, Cheng Hong, and Neil Gong. Fingerprinting llms via prompt injection. *arXiv preprint arXiv:2509.25448*, 2025.

Jiahao Huo, Shuliang Liu, Bin Wang, Junyan Zhang, Yibo Yan, Aiwei Liu, Xuming Hu, and Mingxun Zhou. Pmark: Towards robust and distortion-free semantic-level watermarking with channel constraints. *arXiv preprint arXiv:2509.21057*, 2025.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International conference on machine learning*, pp. 17061–17084. PMLR, 2023.

Jonah Leshin, Manish Shah, Ian Timmis, and Daniel Kang. Behavioral fingerprints for llm endpoint stability and identity. In *Proceedings of the ACM Conference on AI and Agentic Systems*, pp. 1327–1331, 2026.

Bojie Li. Incompressible knowledge probes: Estimating black-box llm parameter counts via factual capacity. *arXiv preprint arXiv:2604.24827*, 2026.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See Kiong Ng, Tat-Seng Chua, and Yang Deng. Knowledge boundary of large language models: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5131–5157, 2025.

Guanjie Lin, Yinxin Wan, Shichao Pei, Ting Xu, Kuai Xu, and Guoliang Xue. Behavioral consistency and transparency analysis on large language model api gateways. *arXiv preprint arXiv:2604.21083*, 2026.

Hanzhi Liu, Chaofan Shou, Hongbo Wen, Yanju Chen, Ryan Jingyang Fang, and Yu Feng. Your agent is mine: Measuring malicious intermediary attacks on the llm supply chain. *arXiv preprint arXiv:2604.08407*, 2026.

Mohammad M Maheri, Hamed Haddadi, and Alex Davidson. Telesparse: Practical privacy-preserving verification of deep neural networks. *arXiv preprint arXiv:2504.19274*, 2025.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

OpenRouter. OpenRouter: A Unified Interface for LLMs. `https://openrouter.ai/`, 2026. Accessed June 12, 2026.

Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. {LLMmap}: Fingerprinting for large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 299–318, 2025.

Gurvan Richardeau, Gohar Dashyan, Erwan Le Merrer, and Gilles Trédan. Flips: Instance-fingerprinting for llms via pseudo-random sequences. *arXiv preprint arXiv:2606.03330*, 2026.

Mark Russinovich and Ahmed Salem. Hey, that's my model! introducing chain & hash, an llm fingerprinting technique. *arXiv preprint arXiv:2407.10887*, 2024.

Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/Ispa*, volume 1, pp. 57–64. IEEE, 2015.

Shuo Shao, Yiming Li, Hongwei Yao, Yifei Chen, Yuchen Yang, and Zhan Qin. Reading between the lines: Towards reliable black-box llm fingerprinting via zeroth-order gradient estimation. In *Proceedings of the ACM Web Conference 2026*, pp. 2637–2648, 2026.

Haochen Sun, Jason Li, and Hongyang Zhang. zkllm: Zero knowledge proofs for large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 4405–4419, 2024.

The OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences. `https://oeis.org/`, 2026. Accessed May 11, 2026.

Florian Tramer and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. *arXiv preprint arXiv:1806.03287*, 2018.

Yun-Yun Tsai, Chuan Guo, Junfeng Yang, and Laurens van der Maaten. Rofl: Robust fingerprinting of language models. *arXiv preprint arXiv:2505.12682*, 2025.

Xun Wang, Bihe Zhao, Michael Backes, Franziska Boenisch, and Adam Dziedzic. Agentprov: Auditing agentic llm api providers via tool-use policy probes. *arXiv preprint arXiv:2609.00052*, 2026.

Ziquan Wang and Zhongqi Lu. Knowledge boundary discovery for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33791–33799, 2026.

Anna Wimbauer, Jonas Möller, Erik Imgrund, and Konrad Rieck. Fingerprinting inference systems of large language models. *arXiv preprint arXiv:2605.29979*, 2026.

Xiangfan Wu, Zonghao Ying, Huiyu Wu, Xing Zheng, Huangsheng Cheng, Xiaorong Shi, and Jing Guo. Ventor-qtest: Threat-model-driven verification of vendor-hosted llm apis. *arXiv preprint arXiv:2608.16391*, 2026.

Zehao Wu, Yanjie Zhao, and Haoyu Wang. Gradient-based model fingerprinting for llm similarity detection and family classification. *arXiv preprint arXiv:2506.01631*, 2025.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3277–3306, 2024.

Zhenhua Xu, Meng Han, and Wenpeng Xing. Evertracer: Hunting stolen large language models via stealthy and robust probabilistic fingerprint. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7019–7042, 2025.

Zhiguang Yang and Hanzhou Wu. A fingerprint for large language models. *arXiv preprint arXiv:2407.01235*, 2024.

Haobo Zhang, Zhenhua Xu, Junxian Li, Shangfeng Sheng, Dezhang Kong, and Meng Han. Attndiff: Attention-based differential fingerprinting for large language models. *arXiv preprint arXiv:2604.05502*, 2026a.

Jiahao Zhang, Xiuyu Li, and Suhang Wang. Your "pro" llm subscription may actually be "free": Exposing fingerprint spoofing risks in llm inference services. *arXiv preprint arXiv:2606.16100*, 2026b.

Yage Zhang, Yukun Jiang, Zeyuan Chen, Michael Backes, Xinyue Shen, and Yang Zhang. Real money, fake models: Deceptive model claims in shadow apis. *arXiv preprint arXiv:2603.01919*, 2026c.

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. *arXiv preprint arXiv:2607.20860*, 2026d.

Youchao Zhou, Heyan Huang, Yicheng Liu, Rui Dai, Xinglin Wang, Xingchen Zhang, Shumin Shi, and Yang Deng. Do retrieval augmented language models know when they don't know? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 35158–35166, 2026.

Xiaoyuan Zhu, Yaowen Ye, Tianyi Qiu, Hanlin Zhu, Sijun Tan, Ajraf Mannan, Jonathan Michala, Raluca Popa, and Willie Neiswanger. Auditing black-box llm apis with a rank-based uniformity test. In *International Conference on Learning Representations*, volume 2026, pp. 54232–54261, 2026.

A Detailed Method and Implementation

This appendix expands the three stages in Section 4.2 and states the scope of the audit. Algorithm 1 constructs the reference fingerprint through probe discovery and self-calibration. Algorithm 2 applies the online audit to a suspect endpoint and returns SAME or DIFFERENT. The implementation below uses numerical recall probes, with shared prompt rendering, batch alignment, and parsing across the three stages.

A.1 Audit Scope

Relay auditing starts from a specific advertised endpoint and a queryable official reference for that endpoint. The probes can therefore be reference-specific rather than universal. LLMmap identifies a model from an enrolled database (Pasquini et al., 2025), whereas KBF compares a claimed endpoint with its enrolled reference. KBF fingerprints the response behavior of the served API endpoint. If the reference endpoint uses retrieval or tools, that behavior is part of the reference. Conversely, if a suspect relay secretly augments a claimed bare-model API with retrieval, tools, or an external fact database, it is no longer serving the same endpoint under the same deployment conditions.

The economic substitution evaluation covers same-tier and lower-tier targets. The false-positive rate is measured in same-reference trials. Our mixed-routing analysis provides point estimates, approximate confidence intervals for known substitutes, and candidate-pool point-estimate ranges under the routing assumptions stated in Section 4.3.

A.2 Stage I: Reference Probe Generation

Table 7: Example domains and matching rules.

Domain	Valid range	$\mathsf{match}_d(a, \hat{a})$
Material boiling point	$[-300, 600]$	$\lvert a - \hat{a}\rvert \leq 3\ ^\circ\mathrm{C}$
Astronomy facts	$[0, 10^{15}]$	relative error $\leq 5\%$
Chromosome count	$[1, 2000]$	exact agreement
Programming release year	$[1950, 2030]$	exact year agreement
Drug half-life	$[0.01, 5000]$	relative error $\leq 10\%$

Domain library. The auditor searches each domain independently for boundary probes and retains those with stable reference completions. Our implementation uses 15 domains, covering chemistry, biology, astronomy, math (e.g., OEIS sequences (The OEIS Foundation Inc., 2026)), programming, and cryptography. Each domain specifies a prompt template, a valid numerical range, a difficulty schedule, and a comparison rule match_d. Table 7 gives examples. Auditors can add or replace domains as long as the facts are numerical, parseable, and produce stable boundary recall.

Candidate proposal. Each search round asks the reference endpoint to propose structured candidate records as `name | value` pairs. The implementation specifies a topic and a difficulty setting, and appends an exclusion list of previously proposed names. The exclusion list reduces repeated proposals during the search. Difficulty progression is an implementation choice. The component ablations in Appendix H hold difficulty fixed while examining specialist candidate generation and the two screening components.

Recall prompts. KBF parses the proposed records, removes duplicate names, rejects values outside the domain range, and renders the remaining records into short audit prompts q_i. Our implementation uses a cloze template for each domain, so a boiling-point record becomes "The boiling point of x at 1 atm is __ °C." The endpoint completes the sentence, leaving a numerical slot that can be recovered under the same rules during consistency checks, self-calibration, and auditing. We use two short, task-oriented prompts and avoid complex behavioral instructions. The system prompt requests adherence to the task, and the user prompt asks for factual recall with one numbered answer per probe.

The candidate-generation, system, and user prompts can all be modified, provided they preserve the probe semantics, yield parseable answers, and support stable reference recall. The auditor applies the revised wording consistently across reference checks, self-calibration, and the corresponding online audit. This flexibility reduces dependence on a fixed template. Appendix B discusses relay recognition and evaluates four alternative probe formats.

Screening and stopping. After the first reference query under c_{audit}, KBF can query a small contrast endpoint (default Qwen3.5-9B) on the same prompt. If it matches the provisional reference value under match_d, the candidate is dropped before the full consistency checks. This saves reference calls and biases the retained set toward candidates better separated from cheap substitutes. The screen is a probe-selection heuristic, so the online audit still compares the suspect endpoint only against the enrolled reference behavior. We disable it for reference models where it removes too much of the usable probe set.

Reference-consistency checks re-query the surviving candidates, including at $\mathrm{temp}{=}0.7$ to filter out unstable probes. A candidate is enrolled only when the responses are valid and agree under match_d. KBF stores the prompt, domain, consensus value, and comparison rule in P_{ref}. After each round, it appends the number of surviving probes $|S_{d,t}|$ to the stable-count history H_d. The predicate $\mathrm{StopDomainCriteria}(H_d)$ fires after two consecutive zero-yield rounds once the domain has enrolled at least five stable probes. The search also ends when it exhausts the difficulty schedule supplied to Algorithm 1.

A.3 Stage II: Reference Self-Calibration

Self-test and stored fingerprint. The auditor fixes P_{ref} and re-queries the reference endpoint under the same configuration used for online audits, with $\mathrm{temp}{=}0$. Missing, unparseable, and out-of-range answers are excluded from scoring. The self-test denominator n_{self} counts valid responses, and k_{self} counts nonmatching responses among them. The auditor sets $p_0 = \mathrm{CP}_{99}(k_{\mathrm{self}}, n_{\mathrm{self}})$ and stores $F_{\mathrm{ref}} = (P_{\mathrm{ref}}, k_{\mathrm{self}}, p_0)$. Algorithm 1 gives both enrollment stages, including the search state, stopping

Algorithm 1: KBF Reference Fingerprint Construction

Oracles: Reference endpoint $\mathcal{O}_{\text{ref}}$

Inputs: Search space $\mathcal{D}$ with domains, audit prompt templates, and match rules; difficulty schedule $\mathcal{T}$; reference-consistency checks $\mathcal{C}_{\text{cons}}$; audit configuration c_{audit}; self-calibration confidence γ

Output: Reference fingerprint $F_{\text{ref}} = (P_{\text{ref}}, k_{\text{self}}, p_0)$

STAGE I: PROBE DISCOVERY

$P_{\text{ref}} \leftarrow \emptyset$
foreach *domain* $d \in \mathcal{D}$ **do**
  $\mathsf{match}_d \leftarrow$ domain matching rule
  $H_d \leftarrow$ empty stable-count history for domain d
  foreach *search round* $t \in \mathcal{T}$ **do**
    $P_{d,t} \leftarrow$ candidate probes proposed by $\mathcal{O}_{\text{ref}}$ for domain d at difficulty t
    $S_{d,t} \leftarrow \emptyset$
    foreach *candidate probe* $q_i \in P_{d,t}$ **do**
      $A_i \leftarrow$ responses from $\mathcal{O}_{\text{ref}}$ to q_i under check configuration $c \in \mathcal{C}_{\text{cons}}$
      if A_i *is valid and stable under* match_d **then**
        $a_i \leftarrow \text{Consensus}(A_i, \mathsf{match}_d)$
        $S_{d,t} \leftarrow S_{d,t} \cup \{(q_i, d, a_i, \mathsf{match}_d)\}$
    $H_d \leftarrow \text{Append}(H_d, |S_{d,t}|)$
    $P_{\text{ref}} \leftarrow P_{\text{ref}} \cup S_{d,t}$
    if $\text{StopDomainCriteria}(H_d)$ **then**
      break

STAGE II: SELF-CALIBRATION

$k_{\text{self}} \leftarrow 0$; $n_{\text{self}} \leftarrow 0$
foreach $(q_i, d_i, a_i, \mathsf{match}_{d_i}) \in P_{\text{ref}}$ **do**
  $\tilde{a}_i \leftarrow \mathcal{O}_{\text{ref}}(q_i, c_{\text{audit}})$
  if $\tilde{a}_i$ *is domain-valid* **then**
    $n_{\text{self}} \leftarrow n_{\text{self}} + 1$
    if $\mathsf{match}_{d_i}(a_i, \tilde{a}_i) = 0$ **then**
      $k_{\text{self}} \leftarrow k_{\text{self}} + 1$
$p_0 \leftarrow \text{CP}_\gamma(k_{\text{self}}, n_{\text{self}})$

return $(P_{\text{ref}}, k_{\text{self}}, p_0)$

rule, and stored output. The implementation also stores raw consensus responses and per-probe self-test outcomes for diagnostics and the matched-probe routing test in Section 4.3.

Null calibration and interpretation. Self-calibration estimates ordinary reference-side noise under the deployed interface, including nondeterministic generation, quantization, cache effects, provider-side wrappers, and other serving details outside the auditor's control. Under the binomial model, each scored probe is treated as an independent discrepancy trial with the same unknown reference error rate p. The CP_γ value is a γ-level one-sided upper confidence bound on this rate. For $0 \le k < n$, it is given by $\text{CP}_\gamma(k, n) = \text{Beta}^{-1}(\gamma;\, k+1,\, n-k)$, with $\text{CP}_\gamma(n, n) = 1$. If the true reference error rate were larger than this bound, observing k or fewer discrepancies would fall in the lower $1 - \gamma$ tail. A stable reference endpoint with many enrolled probes receives a tight null bound, while a noisier endpoint receives a wider bound.

Calibration and auditing use batch validation and retries to handle extensive parsing failures and detected answer misalignment. Responses that remain invalid are excluded from both discrepancy counts and denominators. The binomial model applies to these final scored responses, with independence as a modeling assumption. Shared request context or endpoint state can still affect several valid answers together, so the reported p-values should be interpreted under this assumption.

We report the resulting significance together with independent same-reference controls and robustness results, which provide empirical evidence under the tested deployment conditions.

Algorithm 2: KBF Endpoint Audit

Oracles: Suspect endpoint $\mathcal{O}_{\text{sus}}$
Inputs: Reference fingerprint $F_{\text{ref}} = (P_{\text{ref}}, k_{\text{self}}, p_0)$; audit configuration c_{audit}; significance level α
Output: SAME or DIFFERENT

$n \leftarrow 0$; $k \leftarrow 0$
foreach $(q_i, d_i, a_i, \mathsf{match}_{d_i}) \in P_{\text{ref}}$ **do**
  $\hat{a}_i \leftarrow \mathcal{O}_{\text{sus}}(q_i, c_{\text{audit}})$
  if $\hat{a}_i$ *is domain-valid* **then**
    $n \leftarrow n + 1$
    if $\mathsf{match}_{d_i}(a_i, \hat{a}_i) = 0$ **then**
      $k \leftarrow k + 1$
$p \leftarrow \Pr[X \geq k \mid X \sim \text{Binomial}(n, p_0)]$
if $p < \alpha$ **then**
  return DIFFERENT
else
  return SAME

A.4 STAGE III: SUSPECT ENDPOINT AUDIT

Batched queries. The auditor loads F_{ref} and sends the enrolled prompts to the suspect endpoint under c_{audit}. Oracle calls are issued in domain-homogeneous batches of ten probes, so a batch shares one valid range and one match rule. The recall prompt requests numbered answers, letting the parser map each value back to its probe when an endpoint skips a line, inserts commentary, or reorders answers. The same batching and parsing procedure is used for reference checks and self-calibration.

Response parsing. KBF uses one numerical parser for A_i, $\tilde{a}_i$, and $\hat{a}_i$. It normalizes each response, aligns the numbered answers with their probe slots, and extracts a final value, rejecting anything outside the valid range. A recoverable slot is kept when alignment is clear, and a formatting failure or ambiguous alignment triggers a retry. A probe succeeds only when the recovered value is domain-valid and matches the stored consensus. Missing, unparseable, and out-of-range answers are excluded from both the discrepancy count and the audit denominator n. Valid but nonmatching answers count as discrepancies.

Decision and diagnostics. Algorithm 2 counts these discrepancies and applies the one-sided binomial test with the stored null bound p_0 at $\alpha = 0.05$. It reports DIFFERENT when the observed count exceeds what calibrated reference self-noise can explain, and SAME otherwise. The decision is endpoint-level evidence under the audited interface. The implementation records McNemar's paired test (McNemar, 1947) against the stored self-test outcomes as an auxiliary diagnostic for quantization and provider-comparison analyses. The primary relay-audit decision uses the CP-calibrated binomial rule.

A.5 DETECTION WITH BOUNDARY PROBES

Experimental setup. E16 uses Claude Sonnet 4.6, GPT-5.4, DeepSeek-V3.2, and GPT-4.1-mini as references. For each reference, we generate candidates without early stopping across 15 domains and eight rounds per domain under each setting in Appendix H.1, then freeze 800 probes per setting. Of these 1,600 probes, 378–658 belong to K, passing reference-consistency checks while disagreeing with the construction contrast, Qwen3.5-9B. We evaluate this subset using separate calibration, same-reference, and cross-model response batches, with the other three references serving as targets. We exclude unparseable responses, estimate a CP_{99} null bound from calibration, and apply a one-sided binomial test at $\alpha = 0.01$.

Detection results. Using only K probes yields $\text{FP} = 0/4$ on the same-reference controls and $\text{TP} = 12/12$ across the directed cross-model pairs. Thus, boundary probes alone support calibrated detection in this four-model panel.

B PROBE FORMATS

Probe formats. We construct four variants of the original numerical cloze probes by changing the answer type, question wording, task wrapper, or request grouping. Table 8 compares their request features and gives an example of each format.

1. Entity cloze revises the candidate-generation rules to produce non-numerical probes whose answers are entity names. It retains the fill-in-the-blank format and compares answers by normalized string matching.
2. Numerical QA generates direct questions with numerical answers, replacing the fill-in-the-blank format while retaining the fixed task wrapper.
3. Natural batch QA uses GPT-4.1-mini to rewrite the numerical questions and verify semantic equivalence, then removes the fixed task wrapper and answer-format instructions. Questions remain batched, and GPT-4.1-mini extracts numerical answers from free-text replies.
4. Single-question QA sends the same rewritten questions one per request, removing the batch structure and brief-answer opener while retaining free-text answer extraction.

Table 8: KBF probe formats and detection results. Examples use GPT-5.4 reference probes. • = retained, ○ = removed. The features are numerical answers (Number), blanks (Cloze), a fixed task prompt and answer-format instruction (Wrapper), and ten-question requests (Batch). TP counts detected substitutions and FP counts same-reference false positives at $\alpha = 0.05$. Numerical cloze uses the full 16-model evaluation in Section 5.2, while the four variants use the five-model experiment below. Answers for the two wrapper-free variants are extracted by GPT-4.1-mini.

Format	Number	Cloze	Wrapper	Batch	Example query and stored answer	Detection
Numerical cloze*	•	•	•	•	The boiling point of hexamethyldisilazane at 1 atm is ___°C. Answer: 126.	TP 155/155 FP 0/16
Entity cloze	○	•	•	•	The author of "Fathers of Nations", a prescribed set text for the KCSE examination, is ___. Answer: Paul B. Vitta.	TP 10/10 FP 0/5
Numerical QA	•	○	•	•	At 1 atmosphere pressure, what is the boiling point in degrees Celsius of hexamethyldisilazane, $[(CH_3)_3Si]_2NH$? Answer: 126.	TP 10/10 FP 0/5
Natural batch QA	•	○	○	•	Request opens with "A few quick questions, brief answers are fine:" followed by ten rewritten questions. Item 1: What is the boiling point in degrees Celsius of hexamethyldisilazane, $[(CH_3)_3Si]_2NH$, at 1 atmosphere pressure? Answer: 126, extracted from the free-text reply.	TP 10/10 FP 0/5
Single-question QA	•	○	○	○	What is the boiling point in degrees Celsius of trimethylsilyl bromide, $(CH_3)_3SiBr$, at 1 atmosphere pressure? Answer: 79.5, extracted from the reply "about 79–80 °C".	TP 10/10 FP 0/5

*Original numerical cloze probes from the 16-model evaluation in Section 5.2.

Evaluation protocol. We evaluate all four variants on Claude Sonnet 4.6, GPT-5.4, GLM-4.7, GPT-4.1-mini, and Qwen3.5-27B. Each variant follows the same enrollment and audit procedure as the main experiment in Section 5.2, using its own probe format throughout.

Detection results. Table 8 shows that all four variants detect all ten tested downgrades with no false positives in five same-reference trials per variant.

Further variants and extensions. The core of KBF is the factual question and the stable reference answer it elicits near the knowledge boundary. This signal supports variation in how a question

is phrased, presented, and answered, as the four experiments demonstrate. The construction can extend to additional factual domains, other short-answer types, application-specific wording, mixed-domain requests, and probes embedded in ordinary workloads. Each extension requires a suitable answer-extraction and matching rule, reference-consistency checks in the intended format, and fresh self-calibration. Section 5.3 provides separate evidence for the tested prompt and decoding changes.

C Supplement to Detection Accuracy and Cost

Section 5.2 reports the 155 economically relevant directions of the 16×16 matrix. This appendix adds two results that the main text omits. The first is the complete matrix, including the 85 upgrade directions that Figure 4 leaves blank. The second is a pool of probe sets enrolled after the main evaluation was frozen, together with independent same-model retests. All reported audits use the decision rule of Section 4. The CP_{99} bound on the calibration self-test sets p_0, and a one-sided binomial test at $\alpha = 0.05$ decides each audit. We also provide per-model enrollment costs.

C.1 Upgrade Directions

Why upgrades are outside the threat model. An upgrade direction is a pair (M_r, M_t) in which the served target M_t is more expensive than the claimed reference M_r, for instance an endpoint that advertises GPT-4.1-nano but serves GPT-5.4. Such substitutions may lack an economic incentive and are excluded from the TPR in Section 5.2. Every pairwise audit was nevertheless run in both directions, and Figure 7 shows all 240 off-diagonal cells.

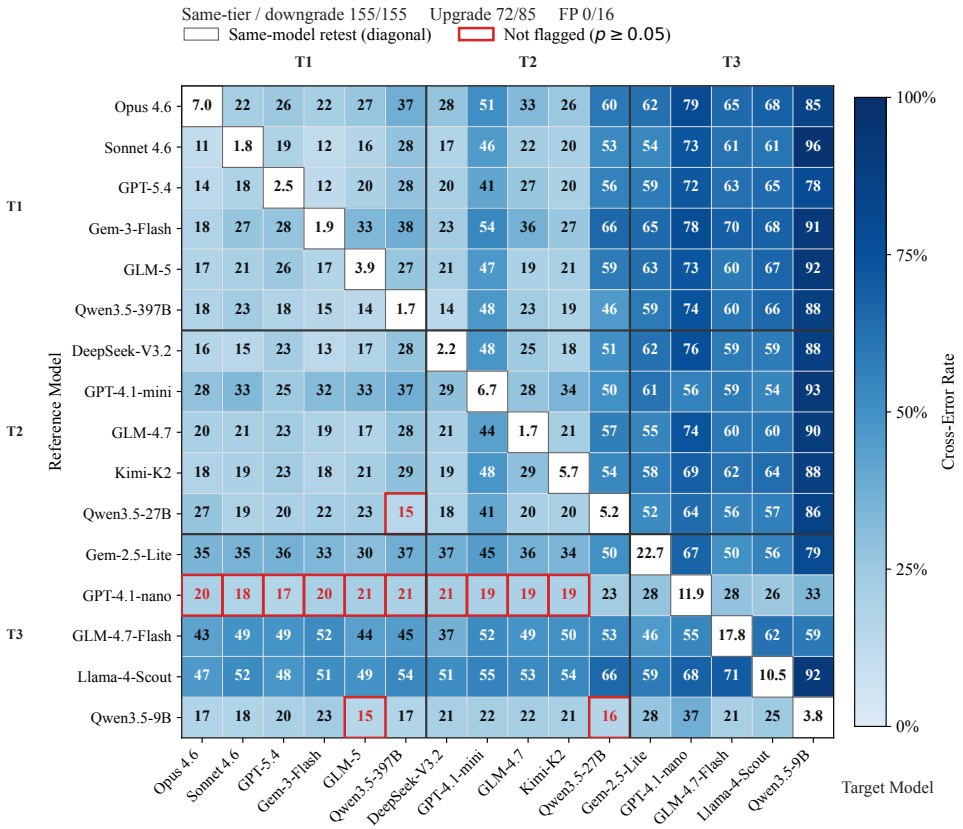


Figure 7: The complete 16×16 detection matrix, including the 85 upgrade directions that Figure 4 leaves blank. Each cell is the disagreement rate of the served target (column) on the reference probe set (row). Diagonal cells show the independent same-model retest of each reference, scored against the bound set by its calibration self-test. Red outlines mark the 13 pairs that the test does not flag at $\alpha = 0.05$; all of them are upgrade directions.

Results. KBF flags 72 of the 85 upgrade directions, so 227 of the 240 measured substitutions are detected overall. The 13 misses are outlined in red in Figure 7. They come from three references only:

GPT-4.1-nano accounts for 10, Qwen3.5-9B for 2, and Qwen3.5-27B for 1. In every case the reverse direction is flagged with a wide margin. The Qwen3.5-397B-A17B probe set flags Qwen3.5-27B at a 46% disagreement rate, while the Qwen3.5-27B probe set reads Qwen3.5-397B-A17B at 15% and does not reject. The GPT-4.1-nano probe set reads every T1 target between 17% and 21%, whereas every T1 probe set reads GPT-4.1-nano between 72% and 79%.

Directional asymmetry. Probes target the knowledge boundary of the reference model. A stronger target can reproduce many of its correct answers, whereas a weaker target fails on more of its boundary facts. All 13 unflagged upgrade pairs involve references with at most 115 retained probes and a null bound $p_0 \geq 10.7\%$, reaching 15.3% for GPT-4.1-nano.

C.2 Frontier Probe Sets

New probe sets. The endpoint pool of Table 2 was frozen in March 2026. Between 10 and 18 September 2026 we ran the released enrollment pipeline, unchanged, against 28 endpoints: 13 models released after the freeze and 15 of the original 16 (GLM-4.7-Flash was not re-enrolled). Table 9 lists them with the pinned provider, the OpenRouter list price recorded on 19 September 2026, the retained probe count, and the enrollment self-disagreement rate that sets p_0. T1, T2, and T3 denote flagship, mid-range, and budget models, respectively. Within T1, T1-1 denotes the latest flagship generation as of September 2026, and T1-2 denotes the generation that was latest in March 2026. Generation and product tier are separate attributes, so a newer mid-range model such as GPT-5.6 Luna remains in T2. Budget-tier references were enrolled without contrastive screening, because the default contrast model is itself a budget model. Probe counts range from 149 to 557, and the median calibration self-disagreement rate is 6.7%. None of the 28 retests is rejected, so this pool adds 0/28 false positives to the 0/16 of Section 5.2.

Table 9: The 28 probe sets enrolled in September 2026. • marks the 13 models released after the March 2026 freeze of Table 2. T1-1 and T1-2 denote the latest flagship generations as of September and March 2026, respectively. *Model Provider* is the OpenRouter inference provider pinned for that enrollment. *Price** reports input- and output-token costs in USD per million tokens. *Self-Err* is the enrollment self-disagreement rate, whose CP_{99} upper bound is the null bound p_0. *Prices were recorded from OpenRouter in September 2026.

Tier	Model	Model Family	Model Provider	Price* ($/M Tokens)		#Probes	Self-Err
				Input	Output		
T1-1	Claude Fable 5.1•	Anthropic	Anthropic	10.00	50.00	453	2.6%
T1-1	GPT-6 Astra•	OpenAI	OpenAI	10.00	50.00	538	1.5%
T1-1	Claude Opus 5•	Anthropic	Claude on AWS	5.00	25.00	414	1.7%
T1-1	GPT-5.6 Sol•	OpenAI	OpenAI	2.00	10.00	426	6.8%
T1-1	Kimi-K3•	Moonshot	Moonshot AI	1.70	8.50	424	8.5%
T1-1	GLM-5.3•	Z.AI	Z.AI	0.91	2.86	372	7.3%
T1-2	Claude Opus 4.6	Anthropic	Amazon Bedrock	5.00	25.00	445	5.2%
T1-2	Claude Sonnet 4.6	Anthropic	Google	3.00	15.00	436	6.4%
T1-2	GPT-5.4	OpenAI	OpenAI	2.50	15.00	531	7.7%
T1-2	GLM-5	Z.AI	Z.AI	0.60	1.92	332	6.6%
T1-2	Qwen3.5-397B-A17B	Alibaba	Alibaba	0.55	3.50	254	6.3%
T1-2	Gemini 3 Flash	Google	Google	0.50	3.00	409	5.1%
T2	Hunyuan 4 Preview•	Tencent	Tencent	0.83	2.50	538	5.8%
T2	GPT-5.4-mini•	OpenAI	OpenAI	0.75	4.50	194	13.4%
T2	Kimi-K2-0905	Moonshot	Novita	0.60	2.50	363	7.8%
T2	DeepSeek-V4-Pro•	DeepSeek	Novita	0.58	1.73	557	6.1%
T2	Seed 2.1 Turbo•	ByteDance	Seed	0.50	2.50	400	6.0%
T2	GLM-4.7	Z.AI	Z.AI	0.40	1.75	430	7.7%
T2	GPT-4.1-mini	OpenAI	OpenAI	0.40	1.60	265	6.4%
T2	DeepSeek-V3.2	DeepSeek	Google	0.27	0.40	380	6.1%
T2	GPT-5.6 Luna•	OpenAI	OpenAI	0.20	1.20	305	7.6%
T2	Qwen3.5-27B	Alibaba	Alibaba	0.20	1.56	149	6.7%
T3	MiniMax M3•	MiniMax	Minimax	0.30	1.20	311	8.1%
T3	DeepSeek-V4.1-Flash•	DeepSeek	Novita	0.15	0.60	363	8.0%
T3	Gemini 2.5 Flash Lite	Google	Google	0.10	0.40	319	10.7%
T3	GPT-4.1-nano	OpenAI	OpenAI	0.10	0.40	248	10.1%
T3	LLaMA-4-Scout	Meta	Novita	0.10	0.30	231	15.2%
T3	Qwen3.5-9B	Alibaba	Together	0.10	0.15	240	5.0%

Release. The probe sets and raw same-model retest files are part of the released data.

C.3 Reference Enrollment Cost

Table 10 reports a separate May 2026 cost run for the six T1 references. This run measures enrollment cost, so its probe counts differ from those in Table 2. It retained 3,696 probes for $7.07 using 1,936,012 tokens, with most of the expense coming from Claude Sonnet 4.6, Claude Opus 4.6, and GPT-5.4. The enrolled fingerprints can be reused across audits of endpoints claiming the same reference version.

Table 10: One-time T1 reference-enrollment cost.

Model	Probes	Cost (USD)	Tokens
Claude Sonnet 4.6	664	$2.17	346,316
Claude Opus 4.6	724	$2.06	280,718
GPT-5.4	531	$1.59	297,718
Gemini 3 Flash	716	$0.44	348,041
GLM-5	613	$0.50	345,303
Qwen3.5-397B-A17B	448	$0.31	317,916
Total	**3,696**	**$7.07**	**1,936,012**

Re-measured in May 2026, for cost assessment only.

D Additional Robustness Analysis

We assess KBF robustness under adversarial prompts, identity spoofing, quantization, backend-provider variation, temporal drift, threshold choice, agent interfaces, and deployment wrappers across the full model pool. Table 11 summarizes the results.

Key takeaway. Most tested settings preserve the audit decision. Two adversarial prompt settings cause false rejections, one that explicitly requests approximate answers and one small probe set with high baseline self-error. Temporal retests identify drift in a small subset of endpoints.

R1. Adversarial prompts. A dishonest relay may target the audit surface directly. We test two variants: adversarial prompt settings that make numerical extraction harder, and identity-spoofing prompts that ask a substitute model to present itself as the reference.

For prompt-level stress, we subject three models (DeepSeek-V3.2, Kimi-K2-0905, GPT-4.1-nano) to five settings: temperature 1.0, an "approximate answers" system prompt, a non-English system prompt with English probes, 2000-character padding, and contradictory few-shot examples. **13 of 15 trials pass**; both failures have clear root causes. Kimi-K2-0905 reaches a 20.0% mismatch rate under the "approximate answers" prompt, which directly instructs the model to abandon numerical precision. GPT-4.1-nano reaches 18.5% at temperature 1.0; this case combines a small probe set (109 probes) with elevated baseline self-error, leaving limited statistical separation. DeepSeek-V3.2 passes all five settings, with a maximum error shift of 3.3%.

R2. Identity spoofing. For identity spoofing, we prepend to each of the other 15 models a system prompt asserting "You are Claude Opus 4.6" at temperature 0.0, with the impersonated reference itself excluded. Under the same CP_{99} binomial test, **all 15 endpoints stay consistent with their own baseline** (0/15 FP). These results support robustness to the tested identity prompt when comparing each endpoint with its own reference.

R3. Quantization robustness. Providers may serve quantized variants (fp8, fp4) under the same model name. This test measures whether such backend choices change KBF's same-endpoint decision. We route GLM-4.7 queries through five OpenRouter providers spanning fp8 and fp4, anchored to the full-precision Z.AI reference, and DeepSeek-V3.2 queries through four providers anchored to the full-precision Google reference. **None of the nine backends is flagged** by the deployment rule (0/9). Every provider stays below the mismatch count at which the CP_{99} binomial test would reject, so serving a quantized variant does not by itself change the audit verdict. The error-rate gap introduced by quantization is small, so separating these backends requires a finer-grained comparison at the level of individual probes rather than aggregate error alone. For this we apply a more sensitive test,

Table 11: Additional robustness tests. Each row summarizes the test coverage, audit outcome, and practical conclusion.

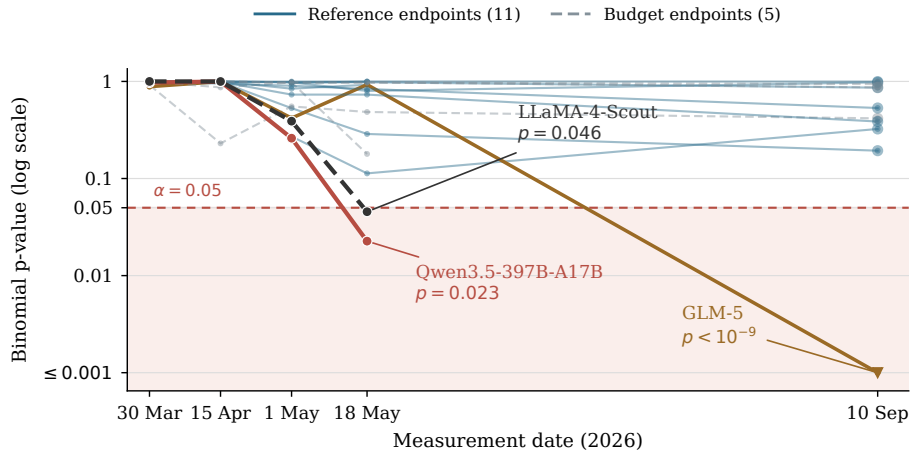
ID	Test	Coverage	Audit outcome	Conclusion
R1	Adversarial prompts	3 models, 5 settings	**13/15** pass	KBF tolerates most tested prompt changes, with failures under precision loss or a small, noisy probe set.
R2	Identity spoofing	15 substitutes	**0/15** false positives	Claimed identities do not override the measured fingerprint in these tests.
R3	Quantization	9 backends	**0/9** flagged	The tested quantized backends preserve the audit decision despite detectable differences in individual probe responses.
R4	Backend providers	5 routes, Sonnet 4.6	KBF **0/5** rejected, IRIS **1/5**	KBF tolerates the tested provider variation, while IRIS falsely rejects one route.
R5	Temporal drift	5 snapshots, then 11 endpoints retested on day 179	**16/16** stable at week 7, **10/11** at follow-up	Observed stability supports probe reuse over several weeks, with later drift motivating periodic refresh.
R6	Threshold choice	10 margins	**0/16** false positives at margins of 4%–10%	Detection tolerates a broad margin range, with 100% TPR at 1–6% and 99.4% at 7%.
R7	Agent interface	3 CLIs, 4 configurations	**0/4** false positives	KBF supports stable model auditing through the tested Claude Code, OpenClaw, and Codex interfaces.
R8	Extended KBF stress test	16 models, 5 configurations	**2/80** false positives	All T1 and T2 models remain stable. GLM-4.7-Flash accounts for both false positives.

McNemar's paired test, which compares per-probe agreement against the reference and separates three of the five GLM-4.7 providers (Venice fp4 $p = 6 \times 10^{-5}$, DeepInfra fp4 $p = 0.008$, AtlasCloud fp8 $p = 0.035$). On DeepSeek-V3.2 it isolates the Novita provider ($p = 0.005$) despite its nominal fp8 label, revealing backend inconsistency that the declared quantization level alone does not expose. Provider pinning in the main detection experiment (Section 5.1) removes this source of backend variation from the 155-pair matrix.

R4. Backend-provider robustness. We audit Claude Sonnet 4.6 through five provider routes using a fixed KBF probe set and layout. KBF accepts all five, with mismatch rates of 5.75%–6.44%. IRIS rejects the Amazon Bedrock route (10/64 tells, $p = 0.00124 < 0.01$), yielding one false rejection across the five routes. These results support KBF's robustness to ordinary backend variation in this setting and show that such variation can trigger a false alarm in IRIS.

R5. Temporal robustness. Probe sets are generated from a reference snapshot, while providers may silently update their backends. For all 16 models, we re-run each same-reference trial across five snapshots spanning 15 March to 18 May 2026 (64 days), applying the same CP_{99} binomial test as the main detection to each available snapshot against its earliest baseline (Figure 8). Probes stay valid through 7 weeks, where **all 16 models remain stable**. At the week-9 measurement, one of the eleven T1/T2 endpoints that can serve as a reference drifted, Qwen3.5-397B-A17B (self-error $1.7\% \rightarrow 7.8\%$, $p = 0.023$), a shift we reproduced on the official API. The rejection of LLaMA-4-Scout is borderline ($p = 0.046$), consistent with its limited stability as a T3 model. The observed increase in reference self-error motivates periodic retesting against the enrolled baseline and refreshing probes when reference behavior changes. On 10 September 2026 (day 179), we retested a sample of 11 endpoints once against their original baselines. Ten remained consistent. GLM-5 showed increased self-error from 4.2% to 16.5%, with $p < 10^{-9}$. Gemini 2.5 Flash Lite had 209 scored responses after excluding one unparseable answer under the archived scoring convention.

R6. Threshold robustness. The main evaluation uses the CP_{99} bound determined by each reference model's self-disagreement rate. As a diagnostic check, we compare this rule against additive margins from 1% to 10% across all 155 economically relevant pairs (Figure 9).



10 Sep: one follow-up test per endpoint; triangles mark values below the axis range.

Figure 8: R5. Temporal stability from 15 March to 10 September 2026 against each endpoint's earliest available baseline. Solid and dashed curves denote reference and budget endpoints, respectively. The September follow-up samples 11 endpoints, with one test per endpoint. Highlighted curves mark rejected endpoints. Triangles place $p < 0.001$ at the plotting floor. The dashed horizontal line marks $\alpha = 0.05$. Baselines are omitted and missing tests break the curves.

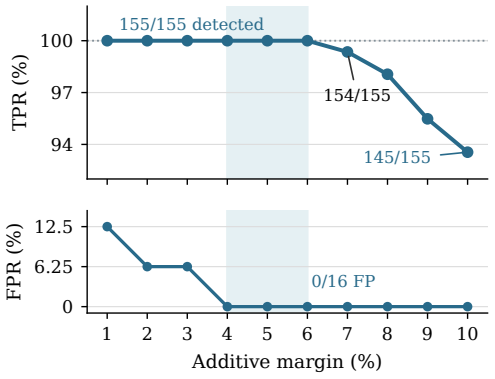


Figure 9: R6. Threshold sweep over 155 substitution pairs and 16 independent same-reference retests. Control rejections fall from 2/16 at 1% to 1/16 at 2%–3% and 0/16 at 4%–10%.

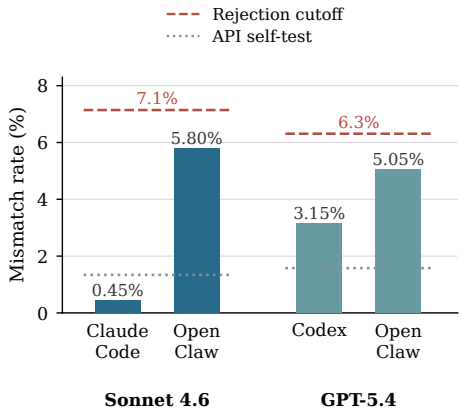


Figure 10: R7. Agent-interface audits. Bars show mismatch rates, dashed lines show rejection cutoffs, and dotted lines show API self-test rates. All four configurations pass.

At margins of 1%–6%, all 155 pairs are detected (100% TPR). At a 7% margin, detection remains at **99.4% TPR** (154/155 TP); the single missed pair (Claude Sonnet 4.6 vs. Claude Opus 4.6) is a within-family pair that sits at the borderline and is correctly detected under the CP_{99} binomial test with the full probe set. At 10%, detection falls to 93.5% TPR as the threshold absorbs the signal from moderately distinguishable pairs. False positives are evaluated on 16 independent same-reference retests, separate from the self-tests used to set the baseline error rates. There are 2/16 false positives at a 1% margin, 1/16 at 2%–3%, and 0/16 at 4%–10%. Thus, margins of 4%–6% retain all 155 detections with no observed false positives in these controls.

R7. Agent-interface probing. We finally test whether KBF stays stable under the system prompt of a real coding agent, so that a user can audit the API behind an agent simply by sending it the probes. We evaluate two references (Claude Sonnet 4.6 and GPT-5.4) across three widely used agent CLIs (Claude Code, OpenClaw, and Codex). On OpenClaw we connect both models for testing, while for the other two official CLIs we query the official APIs directly. In each, the agent reads the probe document and answers inline within its own session, under the agent's full system prompt

(the OpenClaw harness alone adds roughly 30 KB) and without temperature control; we then parse the replies and apply the standard test. **All four configurations pass**: under the CP_{99} binomial test, every agent-side result stays consistent with the model's own API baseline and produces no false positive (0/4 FP), as Figure 10 shows. The agent system prompt is the dominant variable: the Claude Code session with an explicit recall-only instruction lands at 0.45% mismatch, essentially the API baseline, while the two OpenClaw sessions sit near 5% yet remain well inside their rejection thresholds. A user can therefore audit the model behind an agent by sending it the probes directly, without API access.

R8. Extended KBF Stress Test. We extend the stress test in Section 5.3 to the full model pool. We sample five agent configurations from PromptConfFactory (Pasquini et al., 2025), including role prompts, chain-of-thought variants, and RAG injections, and apply each to all 16 models for 80 same-reference trials. All T1 and T2 models remain stable with 0 FP. GLM-4.7-Flash (T3), unstable because of its high baseline self-error, accounts for both false positives (2/80 trials, 2.50% FPR). This gives configuration-specific evidence that KBF remains conservative under ordinary endpoint wrappers.

E Partial-Routing Calibration and Extended Results

E.1 Experimental Setup

We evaluate the eight reference–substitute pairs in Figure 5 using the scoring procedure in Section 4.3. For each reference, all comparisons use a fixed common subset of enrolled probes selected by reference agreement and response availability. The replay preserves request groups of up to ten probes from one domain, reconstructed from archived domain labels and probe order under a consistent alignment assumption.

We reuse reference responses and substitute fingerprints from our earlier experiments. For each request in each online round, an independent Bernoulli draw with probability π selects the archived substitute response for the entire batch. Otherwise, the replay uses the honest-reference response recorded for that round. The reference anchor and candidate fingerprints remain fixed across replays. We run 10,000 replays per pair and round count at each routing fraction, using a 1% grid for detection and $\pi \in \{0, 0.05, \ldots, 1\}$ for estimation. All replays enter the estimation analysis, regardless of whether the audit detects substitution. The simulations condition on archived responses and involve no fresh API queries or reference recalibration.

E.2 Intervals and Candidate Ranges

Known substitute. Let $\widetilde{\pi}_S = \bar{d}/\bar{g}$ be the unclipped ratio estimate from Section 4.3. For $B > 1$ and $\bar{g} > 0$, the delta method gives the standard error

$$\widehat{\mathrm{SE}}_S = \sqrt{\frac{\sum_{b=1}^{B}(d_b - \widetilde{\pi}_S g_b)^2}{B(B-1)\bar{g}^2}},$$

where each request group is one variance unit. For one round, $d_b = Y_{b1} - A_b$. For two rounds, we average the mismatch fractions within each batch before computing d_b.

The nominal 90% delta-method interval is

$$\mathcal{J}_S = \left[\widetilde{\pi}_S - 1.645\,\widehat{\mathrm{SE}}_S,\ \widetilde{\pi}_S + 1.645\,\widehat{\mathrm{SE}}_S\right] \cap [0, 1],$$

with empty intersections retained. This approximation requires independent batches, stable reference means, and $\bar{g}$ sufficiently separated from zero. The calculation conditions on the stored substitute fingerprint.

Unknown substitute. The candidate pool is $\mathcal{C} = \{$GLM-5, Qwen3.5-397B-A17B, Kimi-K2-0905, DeepSeek-V3.2$\}$. We report the range of candidate point estimates defined in Section 4.3, provided all candidates have positive mean mismatch gaps and the pool contains the true substitute. Budget-substitute evaluation reports known-model estimates.

E.3 Extended Results

Partial-routing detection. For one round, we use $T_1 = \sum_{b=1}^{B} \mathrm{sgn}(Y_{b1} - A_b)$ and apply a one-sided paired sign test over non-tied batches. Two rounds use the joint test in Section 4.3. Figure 11 reports

one-round detection for the same pairs as the two-round results in Figure 5. We define MDR_{95} as the first grid fraction reaching 95% TPR under the paired test at $\alpha = 0.05$. For Opus → Kimi-K2, Sonnet → GLM-5, Gemini → Kimi-K2, and GPT-5.4 → Kimi-K2, one-round MDR_{95} values are 34%, 75%, 35%, and 42%, respectively. Two rounds reduce them to 23%, 55%, 22%, and 34%. The four budget substitutions reach 95% TPR at 22–50% routing with one round and 15–34% with two. A second round improves detection at the cost of additional audit queries.

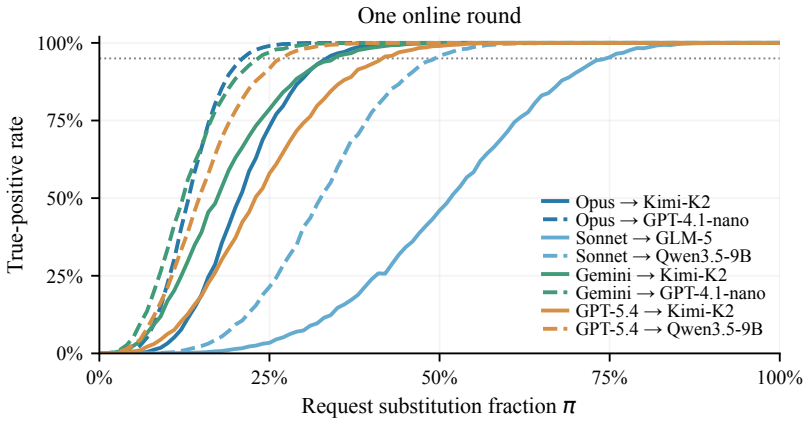


Figure 11: Paired batch detection with one online round. Solid curves denote same-class substitutes and dashed curves budget substitutes. Each point uses 10,000 routing replays at $\alpha = 0.05$, conditional on archived responses. The dotted line marks 95% TPR. Figure 5 gives the two-round results for the same eight pairs.

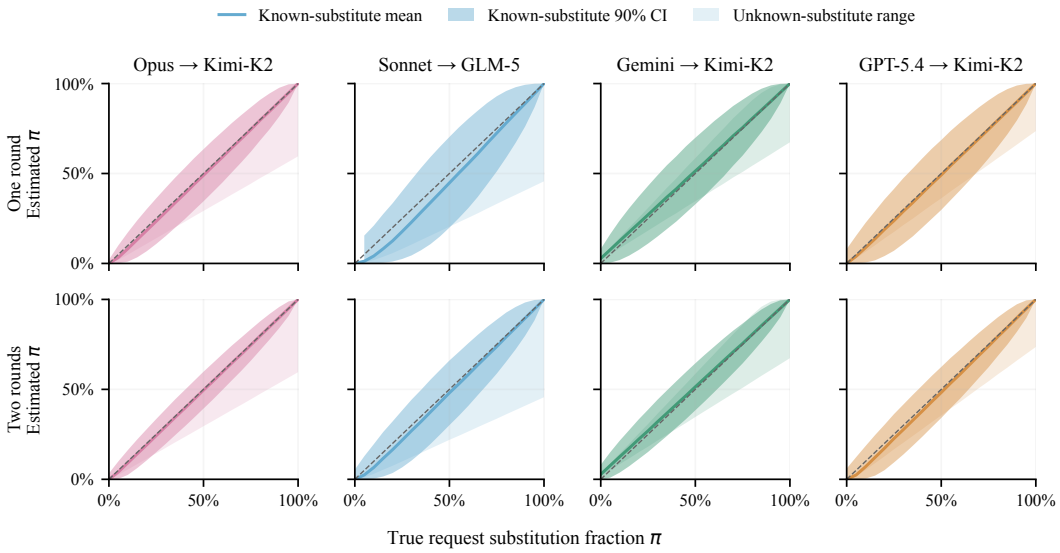


Figure 12: Routing-fraction estimation with unknown substitutes, with one round (top) and two rounds (bottom). Light bands connect mean minima and maxima of candidate point estimates. For comparison, solid lines show mean estimates for known substitutes, and dark bands connect mean endpoints of their nonempty nominal 90% delta-method intervals. Dashed lines show $y = x$.

Unknown-substitute estimation. Figure 12 compares one- and two-round routing-fraction estimates for four same-class pairs, using known-substitute estimates as a reference. The spread across candidate estimates reflects uncertainty about substitute identity. A second round improves substitution detection and narrows the nominal intervals for known substitutes.

F Real-World Relay Audits: Extended Results

This appendix expands the field-audit results of Section 5.6 to all eleven audited platforms.

Per-model overview across all eleven platforms. Figure 13 summarizes the per-model consistency split across all eleven audited platforms, where each bar covers every platform that exposes the model. The full survey shows the same concentration pattern as the representative subset. Claude Opus 4.6 is flagged on 7 of 10 platforms that serve it and Claude Sonnet 4.6 on 3 of 11, so the two most expensive proprietary models account for **10 of the 12** flagged endpoints across the eleven platforms. GLM-4.7 is flagged on 2 of 5 platforms, while GPT-5.4, DeepSeek-V3.2, and Gemini 3 Flash are consistent on every platform that exposes them. The aggregate flag rate across all eleven platforms is **12/42**, measuring statistical inconsistency with the probe-response distributions of the pinned references in this sample.

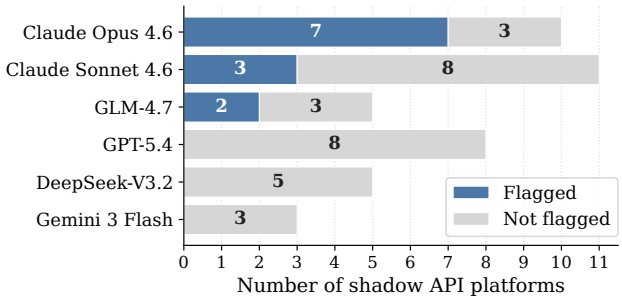


Figure 13: Per-model consistency split across the eleven-platform shadow API field audit. Each bar covers all platforms that expose the model.

To assess repeatability, the high-mismatch Claude endpoints on Platform 1 and Platform 5 are re-measured across three independent rounds, while the rest are single-round.

Platform Mapping. Table 12 maps the platform identifiers used in Section 5.6 to their salted SHA-256 digests.

Table 12: Platform identifiers and their salted SHA-256 digests.

P1 `SH-87b19c56`	P2 `SH-3c82a49c`	P3 `SH-4f754cc3`
P4 `SH-75f73c6e`	P5 `SH-0989276f`	P6 `SH-031d44cf`
P7 `SH-3607d87e`	P8 `SH-1a4df8c4`	P9 `SH-feecd406`
P10 `SH-7e45d454`	P11 `SH-bd05fa88`	

G Baseline Configuration and Related Work

G.1 Baseline Configuration

LLMmap, MET, ZeroPrint, and IRIS cover the 16-model pool, while GateScope covers a 12-model subset. We use the released implementations with the adaptations described below to decide whether a suspect endpoint is consistent with its claimed reference. Provider pins remain fixed within each collection. The later IRIS and GateScope runs use Novita for Kimi-K2 and LLaMA-4-Scout for these later collections.

LLMmap. LLMmap is an enrollment-and-retrieval fingerprint. We extend the pretrained open-set library with 16 new model templates, and at test time the unknown endpoint answers 8 queries in the bare-query setting at $\mathrm{temp}{=}0$. LLMmap converts the responses into a feature vector and ranks all enrolled templates by Euclidean distance.[1] An endpoint is consistent with its claimed reference when that reference is its Top-1 nearest template, and any other Top-1 match flags it as a different model.

MET. MET is a two-sample distributional test. We use 25 multilingual Wikipedia continuation prompts drawn from a 500-record candidate pool, and for each model collect two independent batches of 250 completions (10 per prompt). The test statistic is MMD with a Hamming kernel, and p-values are estimated from 1000 permutations. An endpoint is consistent with its claimed reference when the two-sample test does not reject the same-model null at $\alpha{=}0.05$, and a rejection flags it as a different model.

ZeroPrint. ZeroPrint uses structured perturbations to estimate local response sensitivity, building a Jacobian-style fingerprint compared by Pearson similarity. Following the original method, we

[1]We follow the public implementation and use Euclidean distance for the ranking computation.

issue each prompt and its perturbed variants 20 times, for 200 requests per model. Our adaptation is the relay-auditing decision. The predefined similarity threshold from the original evaluation is fixed on base-and-derivative model pairs and does not suit this setting, so we instead fingerprint each endpoint in two independent collections and build a reference library from the first. We then pool all query–reference similarity scores and set a single global decision threshold at the Youden-optimal operating point, matching the original protocol of choosing one threshold over the full score set. An endpoint is consistent with its claimed reference when its similarity to that reference's library fingerprint meets the global threshold, and a similarity below the threshold flags it as a different model.

IRIS. We use the publicly released IRIS code for reference enrollment and endpoint auditing. At $\mathrm{temp}{=}1$, we collect 100 enrollment responses per model for each of six random-generation prompts, 120 independent calibration responses, and 64 audit responses cycling through the six prompts. We train a random forest on content-only features pooled across the prompts and calibrate its threshold using independent honest responses. Under the released decision rule, a response is a tell when its negative log posterior for the claimed model exceeds the 95th percentile of the calibration scores. The audit rejects the same-model null when an upper-tail binomial test of the tell count against a nominal honest rate of 0.05 gives $p < 0.01$. Robustness tests keep enrollment and calibration fixed and collect 64 responses under each deployment configuration.

GateScope. GateScope uses 55 capability probes with a JSON system prompt at $\mathrm{temp}{=}0.7$ and one-vs-rest XGBoost classifiers. We retain vendor-default reasoning and use six repetitions per probe, split into three for training, one for probe selection, and two for calibration. The evaluated subset excludes Claude Opus 4.6, GPT-5.4, Qwen3.5-397B-A17B, and GLM-4.7-Flash. We implement selection of 12 probes per claimed model and add a decision rule to the original attribution scores. For each claimed model, 24 held-out responses estimate the honest attribution rate, and 60 independent audit responses are tested against it using a lower-tail binomial test at $\alpha{=}0.01$. Enrollment and audit collection for the 12 evaluated models cost $29.78.

G.2 Black-Box Auditors

Black-box auditors differ in their probe signals, decision rules, and reference requirements. Table 13 summarizes these properties. Appendix G.1 describes the baselines evaluated in this paper.

Prompt banks and classifiers. LLMmap sends eight fixed queries, encodes the answers with a trained trace encoder, and returns the nearest enrolled template (Pasquini et al., 2025). FDLLM trains a detector on generations from a closed model set (Fu et al., 2025), and GateScope classifies JSON answers to competition and factual questions with one XGBoost model per endpoint, then reports the share of gateway responses attributed to the advertised model (Lin et al., 2026). These methods identify enrolled models quickly, but the verdict is a label or a distance with no bound on how often an honest endpoint is flagged, and the fixed query set can be recognized and forwarded to the genuine backend. GhostPrint shows that a cheaper model can be fine-tuned to pass such signatures (Zhang et al., 2026b).

Distribution tests. MET samples continuations of Wikipedia prefixes from both endpoints and applies a two-sample MMD test (Gao et al., 2024). RUT sends each prompt once, scores the response by its log-rank under a locally hosted reference, and tests the ranks for uniformity, which needs open weights on the reference side (Zhu et al., 2026). AgentProv applies the same two-sample idea to which of several equivalent tools the model calls (Wang et al., 2026). All three use calibrated tests, but MET remains sensitive to deployment changes. Section 5.3 measures this for MET.

Random-generation bias. Asked for a random integer or bit string, a model produces a skewed distribution that identifies its backend. IRIS turns such strings into features, scores each response by how atypical it is for the claimed model, and adds a tell-count test for partial routing and a pilot that fixes the budget in advance (Zhang et al., 2026d). Bruckner compares single-word answer distributions by Jensen–Shannon distance (Bruckner, 2026), FLIPS classifies NIST randomness statistics of generated bit sequences (Richardeau et al., 2026), and Ventor-QTest compares single-token draws in frozen contexts with a coarsened KL statistic (Wu et al., 2026). These probes require no per-model mining. The signal, however, lives in the tail of the sampling distribution, so it needs a known nonzero temperature and shifts with quantization and kernels. IRIS separates fourteen of fifteen same-model provider pairs, Bruckner finds ten of thirty-four such pairs farther apart than different models, and Wimbauer et al. show that the inference engine and GPU leave readable traces in text (Wimbauer et al., 2026). A random-generation fingerprint therefore measures the serving

Table 13: Black-box auditors. **Probe**: univ = one fixed set for all models, per-ref = generated for the claimed reference, natural = ordinary user prompts. **Decision**: test = hypothesis test, classifier = learned classifier or nearest neighbor, threshold = similarity or distance cutoff. **FPR**: the rule bounds the false-positive rate. **Routing**: detects partial substitution.

Method	Signal	Probe	Decision	FPR	Temp	Reference	Routing
KBF (ours)	knowledge boundary	per-ref	test	Y	any	official API	Y
LLMmap (Pasquini et al., 2025)	prompt bank	univ	classifier	N	any	enrolled library	N
MET (Gao et al., 2024)	completion distribution	natural	test	Y	> 0	reference samples	N
ZeroPrint (Shao et al., 2026)	input perturbation	univ	threshold	N	any	reference fingerprint	N
FDLLM (Fu et al., 2025)	generated text	univ	classifier	N	any	labeled corpus	N
GateScope (Lin et al., 2026)	capability probes	univ	classifier	N	0.7	official API corpus	N
RUT (Zhu et al., 2026)	completion distribution	natural	test	Y	0.5	local open weights	Y
AgentProv (Wang et al., 2026)	tool choice	univ	test	Y	0.7	reference instance	Y
IRIS (Zhang et al., 2026d)	random generation	univ	test	Y	> 0	reference + candidates	Y
One Token (Bruckner, 2026)	random generation	univ	threshold	N	1	registered deployment	N
FLIPS (Richardeau et al., 2026)	random generation	univ	classifier	N	> 0	enrolled instances	N
Ventor-QTest (Wu et al., 2026)	random generation	univ	test	Y	1	official API + logprobs	N
IKP (Li, 2026)	factual recall	univ	threshold	N	0	open-weight calibration	N

configuration together with the weights. KBF compares greedy answers, which these changes flip far less often.

Perturbation and change detection. ZeroPrint fits a local Jacobian from paraphrased inputs to response embeddings and compares endpoints by correlation, which depends on the embedding model and on a threshold that must hold across configurations (Shao et al., 2026). B3IT, Log Probability Tracking, and the Stability Monitor compare an endpoint with its own past, through border inputs at zero temperature, first-token log probabilities, or embedding distances aggregated over time (Chauvin et al., 2026a;b; Leshin et al., 2026). These methods use historical behavior as the baseline for detecting endpoint changes. Frequent changes in relay backends and serving configurations make it difficult to collect a stable history of the claimed model at a relay endpoint. These methods are therefore poorly suited to black-box verification in our relay setting, and we exclude them from the empirical comparison.

Knowledge-based signals. Knowledge-boundary work asks which facts a model holds and how it should answer beyond them (Li et al., 2025; Deng et al., 2025; Wang & Lu, 2026). Concurrent work on Incompressible Knowledge Probes estimates model size from recall across rarity tiers and uses shared wrong answers as a lineage signal (Li, 2026). KBF independently develops a knowledge-boundary fingerprint for relay auditing, with per-reference probes, greedy matching, and a binomial test whose null is calibrated on the reference itself.

G.3 Additional Related Work

The methods below either need more than text from the audited endpoint or need another party to act first.

Adversarial suffixes and planted fingerprints. TRAP optimizes a suffix to induce a chosen completion (Gubri et al., 2024), RoFL optimizes prompts for model identification (Tsai et al., 2025), and LLMPrint injects instructions that force model-specific token preferences (Hu et al., 2025). The probes are built with access to the source model and look nothing like ordinary traffic, so a relay can refuse them or forward them to the genuine backend (Shao et al., 2026). Planted fingerprints

embed ownership signals through training or fine-tuning (Xu et al., 2024; Adi et al., 2018; Gloaguen et al., 2025; Xu et al., 2025; Russinovich & Salem, 2024). They settle ownership disputes over stolen weights, which a relay user never holds.

Internal and probability channels. Attention and gradient fingerprints need white-box access (Zhang et al., 2026a; Wu et al., 2025). Logit-space signatures work through an API that returns probabilities (Yang & Wu, 2024; Finlayson et al., 2025). Relays rarely pass probabilities through, and closed providers return at most a truncated top-k.

Watermarks and verifiable inference. Sampling watermarks introduce detectable statistical patterns at the token or semantic level (Kirchenbauer et al., 2023; Dathathri et al., 2024; Hou et al., 2024; Huo et al., 2025). They answer whether text came from a watermarked model, and they help a relay user only if the official provider watermarks every response and exposes the detector. Zero-knowledge proofs, trusted hardware, and selective verification prove that an inference ran on the claimed model (Sun et al., 2024; Maheri et al., 2025; Tramer & Boneh, 2018; Sabt et al., 2015; Guo et al., 2026). These give stronger guarantees than any behavioral test and start working only after the provider deploys them. KBF covers the intermediary that has deployed nothing.

H Ablation Study

Key takeaway. With the same number of generation rounds, specialist facts yield more retained probes and larger agreement gaps for most tested model pairs. Stability filtering lowers reference noise, while contrast filtering improves model discrimination, supporting detection at a fixed audit budget. Replacing Qwen3.5-9B with Phi-4 preserves detection performance across all tested model pairs.

H.1 Components and Evaluation

Specialist candidate generation. The reference proposes numerical facts from specialist topics within each domain. This component controls which facts enter the candidate pool. Its ablation replaces the specialist prompts and themes with common factual topics in the same domains, while retaining the question format and numerical comparison rules.

Stability filtering. The reference answers each candidate once at temperature 0 and twice at temperature 0.7. The filter retains candidates whose three responses satisfy the enrollment consensus rule. Its ablation removes this agreement requirement while retaining contrast filtering.

Contrast filtering. Qwen3.5-9B answers the same candidates. The filter retains a candidate when the contrast answer is parseable and differs from the temperature-0 reference answer under the domain comparison rule. Its ablation omits this test while retaining stability filtering.

The full method combines specialist candidate generation with both filters. Table 14 specifies the three ablations. The three component ablations hold difficulty at the initial level and register the temperature-0 reference answer, so a filter ablation changes probe selection without changing the answer used for scoring.

Table 14: Each ablation changes one component of the full method. Common-fact generation replaces specialist topic selection and retains both filters.

Configuration	Candidate topics	Stability filter	Contrast filter
Full method	Specialist	On	On
Common facts	Common	On	On
Without stability filtering	Specialist	Off	On
Without contrast filtering	Specialist	On	Off

Data and metrics. We evaluate Claude Sonnet 4.6, GPT-5.4, DeepSeek-V3.2, and GPT-4.1-mini as references, with the other three models serving as targets for each reference. Calibration, reference self-evaluation, and target evaluation use separate response batches. Agreement measures reproducibility of the registered answers rather than factual correctness, with invalid responses counted as mismatches. Model discrimination is the reference agreement rate minus the target agreement rate for each model pair.

Generation budget and evaluation coverage. The generation comparison uses the first four rounds in each of 15 domains, totaling 60 rounds per reference and arm. Round selection follows

generation order without conditioning on yield or evaluation outcomes. Independent calibration, reference self-evaluation, and all three target evaluations cover every retained common-fact probe in the 60-round pool, while specialist coverage remains partial for three references. Figure 14 reports yields and pairwise agreement gaps for this 60-round comparison.

Screening audit budget. Screening variants select probes from the same evaluated fixed-difficulty candidate set for each reference. We evaluate 200-probe audits across 100 selection seeds, using separate 200-probe calibration batches, a model-specific CP_{99} null bound, and the one-sided binomial test at $\alpha = 0.05$. Each directed reference–target pair therefore has 100 audit decisions. We report each directed pair separately and evaluate self-controls separately for each reference. The seeds select probe subsets from stored responses. The screening ablations report target rejection percentages for individual directed model pairs at 200 probes, restricted to the evaluated candidate set. The contrast-model comparison in Section H.5 uses its separately specified protocol.

H.2 Ablating Specialist Candidate Generation

Comparison. We replace specialist topic selection with common factual topics while retaining both filters. Both arms use the same 15 domains, numerical question templates, matching tolerances, and 60 generation rounds. For example, the astronomy control requests orbital and rotation periods of planets and major moons rather than measurements of obscure objects. Yield counts all retained probes, while agreement gaps use those with complete independent evaluations. Specialist evaluation covers 217/351, 338/338, 188/307, and 125/191 retained probes for Sonnet 4.6, GPT-5.4, DeepSeek-V3.2, and GPT-4.1-mini, respectively, while common-fact evaluation covers every retained probe.

Effect on probe yield. At the same 60 generation rounds, specialist generation meets our minimum requirement of 100 retained probes for every reference (Figure 14a). Common-fact generation reaches this count for three references, while DeepSeek-V3.2 yields 89 probes. For Sonnet 4.6, common facts yield 137 retained probes versus 351 with specialist topics. For GPT-5.4, the corresponding counts are 135 and 338. Across the four references, specialist generation supplies 1.71–3.45 times as many probes passing both filters. Across references, 69.33–74.26% of valid common-fact candidates pass stability filtering but share their registered answer with Qwen3.5-9B, so contrast filtering removes them. These counts compare generation rounds, not token or monetary cost.

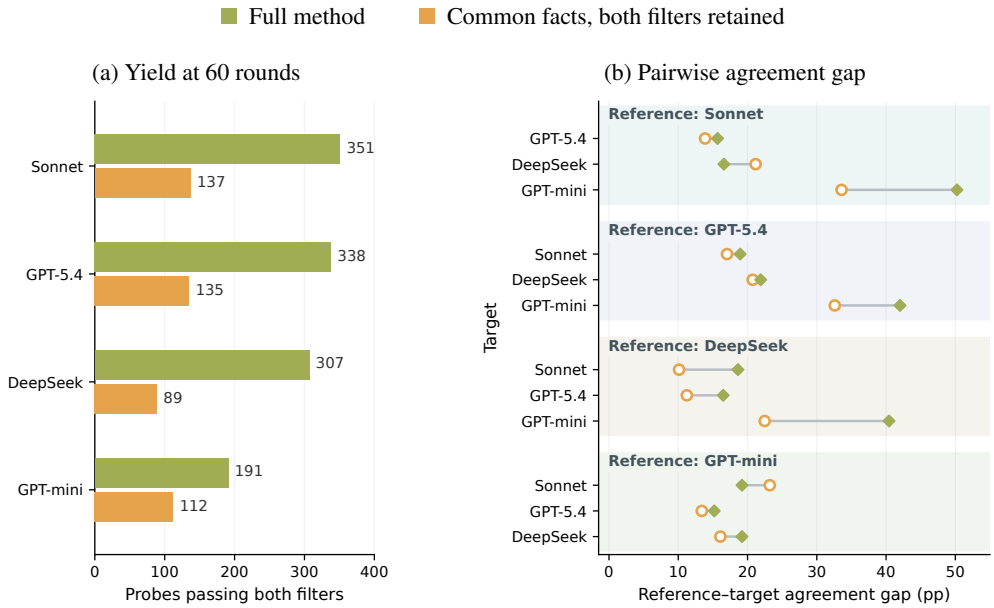


Figure 14: Specialist versus common-fact generation, with both filters retained. (a) Retained counts from the first four rounds per domain. (b) Reference self-agreement minus target agreement for each directed pair, in percentage points. Shaded groups identify references and rows identify targets. Green diamonds denote specialist probes and yellow circles common facts.

Effect on model discrimination. Common-fact probes also provide smaller agreement gaps for several model pairs (Figure 14b). With Sonnet 4.6 as the reference and GPT-4.1-mini as the target, the gap is 33.58 percentage points for common facts versus 50.23 for specialist topics. With GPT-5.4

as the reference and the same target, the gap is 32.59 versus 42.01 percentage points. Specialist probes have larger gaps in 10 of the 12 directed pairs, so the improvement depends on the target. The consistent benefit of specialist generation is its larger retained pool under the same number of generation rounds. Data were collected at different times, and specialist evaluation remains partial for three references, so the comparison does not isolate topic selection from temporal or serving variation.

H.3 Ablating Stability Filtering

Comparison. We remove only the requirement that the reference reproduce its answer across the three construction queries. Specialist candidate generation, contrast filtering, and the registered temperature-0 answer remain unchanged. Both configurations use the same evaluated candidate frame and the same 200-probe audit budget.

From reference stability to the audit decision. The audit rejects a target when its mismatch count exceeds what the calibrated reference noise can explain. For a fixed probe budget and significance level, a larger null bound p_0 requires more mismatches to reject the target. Stability filtering therefore affects the final decision through reference calibration, as well as through the reference–target agreement gap. Removing the filter raises the calibrated null bound for each reference at 200 probes (Figure 15a). The audit consequently tolerates more target mismatches before reporting DIFFERENT.

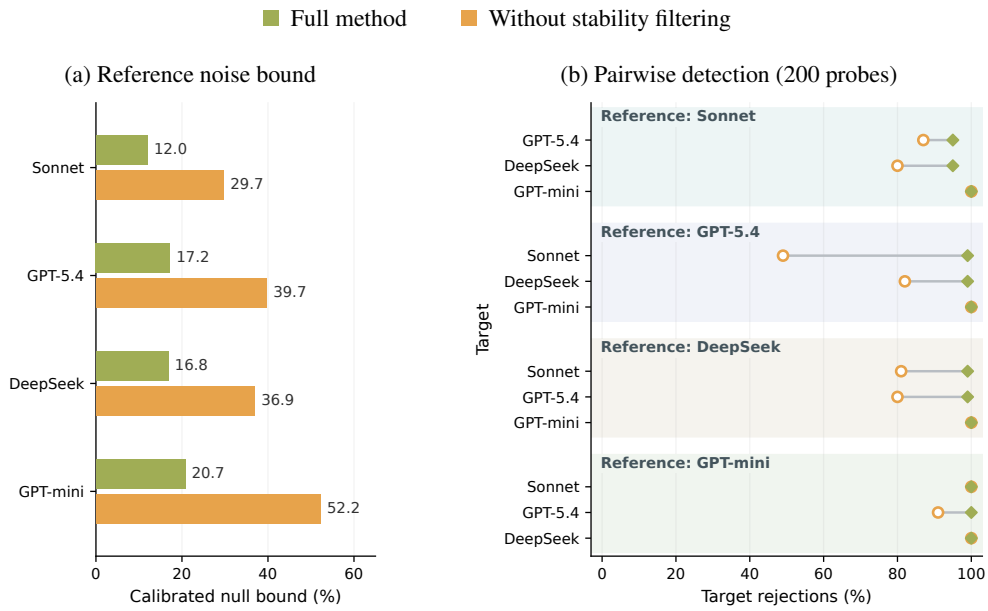


Figure 15: Effect of stability filtering at 200 probes. Green denotes the full method and yellow removal of the filter. (a) Mean calibrated CP_{99} reference mismatch bound across 100 selection seeds. (b) Target rejections per directed pair over those 100 seeds at $\alpha = 0.05$. Shaded groups identify references and rows identify targets. Seeds reuse stored responses, and each reference has 0/100 self-control rejections under either configuration.

Effect on discrimination and detection. The reference–target agreement gap alone does not determine the decision, which also depends on the calibrated null bound and the probe budget. With GPT-5.4 as the reference and Sonnet 4.6 as the target, removing stability filtering reduces rejections from 99/100 to 49/100 selection seeds at 200 probes (Figure 15b). The reverse direction, with Sonnet 4.6 as the reference, falls from 95/100 to 87/100. Each reference has 0/100 self-control rejections under either configuration. Stability filtering thus supports the final judgment by reducing reference noise and improving detection at a fixed audit budget.

H.4 Ablating Contrast Filtering

Comparison. We remove only the requirement that Qwen3.5-9B give a different answer. Specialist candidate generation, stability filtering, and the registered answer remain unchanged. The comparison again uses the same candidate frame and a 200-probe audit budget.

Effect on model discrimination. Without contrast filtering, the reference–target agreement gap decreases for every directed pair. The retained answers are more widely shared across models. Figure 16a summarizes target mismatch separately for each reference as a probe diagnostic. The detection results in panel (b) retain every target separately.

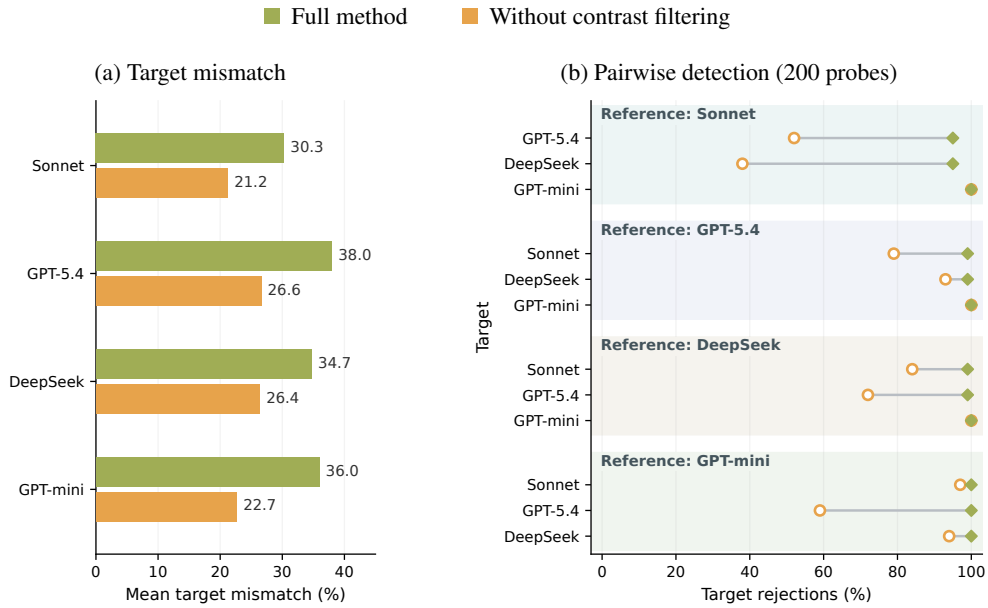


Figure 16: Effect of contrast filtering. Green denotes the full method and yellow removal of the filter. (a) Mean target mismatch over all retained probes in the evaluated frame. (b) Target rejections per directed pair over 100 selection seeds, using 200 audit probes, 200 independent calibration responses, and $\alpha = 0.05$. Shaded groups identify references and rows identify targets. Seeds reuse stored responses, and each reference has 0/100 self-control rejections under either configuration.

Effect on detection. For Sonnet 4.6 as the reference and DeepSeek-V3.2 as the target, rejections fall from 95/100 to 38/100 (Figure 16b). For Sonnet 4.6 as the reference and GPT-5.4 as the target, rejections fall from 95/100 to 52/100. Each reference has 0/100 self-control rejections under either configuration. The small improvement in reference stability does not compensate for the loss of model discrimination.

H.5 Choice of Contrast Model

Comparison and evaluation. We replace Qwen3.5-9B with Phi-4 or Llama-3.1-8B-Instruct when screening the stored candidates, while retaining specialist candidate generation and stability filtering. A fourth arm omits contrast filtering. This control uses the same four references and their 1,600-question evaluation frames, with progressive candidate generation, consensus answers, and up to 400 retained probes per arm over ten selection seeds. Each arm uses separate calibration responses to compute its own CP_{99} bound and applies the one-sided binomial test at $\alpha = 0.05$, excluding unparseable responses from each batch. These settings differ from the fixed-difficulty, 200-probe screening ablations above. For each reference, we evaluate the other three references and independent Qwen3.5-9B target answers. The selection seeds sample probe subsets from stored responses.

Effect on detection. Replacing Qwen3.5-9B with Phi-4 preserves detection for every tested directed pair, with both screening models yielding 10/10 rejections per pair. Llama screening yields 0/10 rejections for Sonnet 4.6 $\rightarrow$ GPT-5.4 and 10/10 for every other pair. Without contrast filtering, Sonnet 4.6 $\rightarrow$ GPT-5.4 and Sonnet 4.6 $\rightarrow$ DeepSeek-V3.2 yield 7/10 and 9/10 rejections, respectively, while all other pairs yield 10/10. Every arm detects Qwen3.5-9B in 10/10 seeds for each reference,

including when another model performs contrast screening. Each reference has 0/10 self-control rejections under every arm.

Effect on probe selection. Different contrast models select different probe sets even when their detection results agree (Figure 17). The Qwen and Phi-4 sets have Jaccard similarity of 0.49–0.59, measured as the size of their intersection divided by the size of their union. Llama retains 151–197 probes per reference, compared with 210–376 for Qwen and 263–373 for Phi-4, with Qwen–Llama similarity of 0.25–0.30. All three contrast-screened sets fit below the cap. The comparison therefore measures the combined effect of changes in probe membership, retained count, and calibration, rather than isolating contrast-model identity at a fixed probe count.

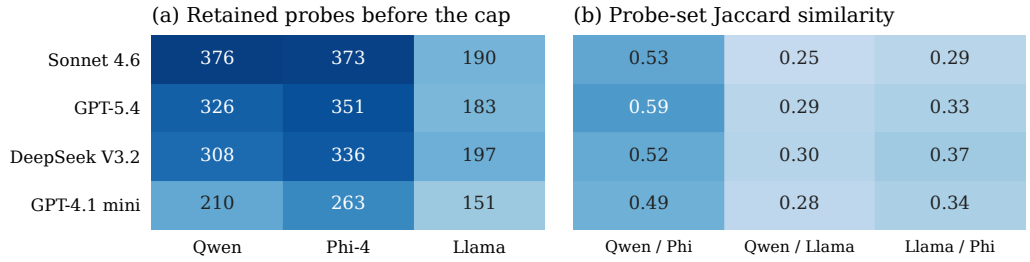


Figure 17: Effect of contrast-model choice on probe selection within the common evaluation frame. (a) Number of retained probes for each reference and contrast model. (b) Pairwise Jaccard similarity of the selected probe sets, where 1 denotes identical sets and 0 denotes disjoint sets. Rows identify the same references in both panels.

I Limitations and Future Work

Operational use. KBF should be used as a statistical triage tool, not as standalone proof of provider intent or backend identity. A rejected audit means that the suspect endpoint is statistically inconsistent with the reference endpoint on the enrolled probe set under the audited interface. Provider logs and independent route confirmation can support further investigation of the served backend and routing policy. Repeated audits under fresh private probes can assess whether an inconsistency persists. Auditors should keep the probe pool private and rotate probes across repeated audits.

Future work. KBF audits whether an endpoint serves the claimed model. Future work should extend this view in two directions. First, high-stakes operational use requires stronger statistical calibration through repeated same-endpoint audits across days, accounts, provider routes, regions, and fresh private probe sets. Second, auditors should measure beyond model identity, including routing transparency. A relay may fall back, mix providers, or apply user-specific routing policies while still advertising a single model. Message integrity is another direction. Recent work shows that malicious intermediaries can inject tool payloads and observe secrets in transit (Liu et al., 2026), and auditors need black-box tests for request rewriting, response mutation, and credential exposure. A complete model-serving audit should also cover billing transparency, including inflated token counts, hidden cache hits, batch-price arbitrage, and account or free-credit resale.