SENTINELMARK-MOE: SECRET CHALLENGES FOR AUDITING MOE EXECUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-experts (MoE) models route each token to only a small set of experts. This saves computation, but it separates the route chosen by a trusted router from the work carried out by a distributed backend. Recording the intended route does not reveal whether the backend rerouted, omitted, or replaced expert work. Existing model watermarks mainly test ownership or generated text, not the route that a serving system actually executes.

We introduce SENTINELMARK-MoE, an online audit for MoE route execution. For each decision, it forms two routes that satisfy the routing constraints and uses secret scores under a fresh tag to choose one. After execution, a trusted observer records a small summary of the route that ran. The detector checks this summary for the expected secret pattern. It can raise an alarm at any time while controlling false alarms. It skips pairs with identical summaries and uses the score distribution to better detect infrequent changes. With fresh random scores, selection preserves the chosen routing distribution exactly; a pseudorandom function provides a computational version of this guarantee.

Experiments across three MoE families and five language benchmarks show that SENTINELMARK-MoE detects changes in expert execution while maintaining model quality. It consistently outperforms the tested statistical detectors against rerouting, omitted expert work, and expert-identity substitution. The results highlight two practical findings: model-specific calibration keeps routing changes small, and the choice of trusted observation determines which attacks the auditor can detect. Matched-prompt evaluations further show the trade-offs between detection, model quality, and serving cost.

1 INTRODUCTION

Mixture-of-experts (MoE) models route each token to a small subset of experts, making large models practical to serve (Shazeer et al., 2017; Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024). The selected route can differ from the work that runs. Even when the trusted router selects the intended experts, a distributed backend can reroute a token, omit an expert call, reuse earlier work, or substitute a cheaper replica. The API response does not reveal which path actually ran.

Existing watermarks address different questions. Text watermarks identify generated text (Kirchenbauer et al., 2023; Kuditipudi et al., 2024; Christ et al., 2024; Dathathri et al., 2024). WaterMoE changes routing to watermark generated text, while PathMark uses routing signatures to verify model ownership (Sun et al., 2026; Gao et al., 2026). These methods assume that the serving system follows the marked computation. Signing a route log still records only intent: the same backend can sign an honest route and execute another one. Full attestation or verifiable computation can check execution, but requires more system support (Ghodsi et al., 2017; Tramèr & Boneh, 2019; Liu et al., 2021).

We study a directly testable question: *did the backend execute the authenticated route summary selected for this request?* MoE routing often allows several token-to-expert assignments under the same top-k, capacity, and expert-call constraints. We use a trusted selector with secret scores under a fresh tag to choose among them. The chosen route carries both real work and a hidden challenge. Changing the observed route changes the input to the secret score function, which the backend cannot evaluate.

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.

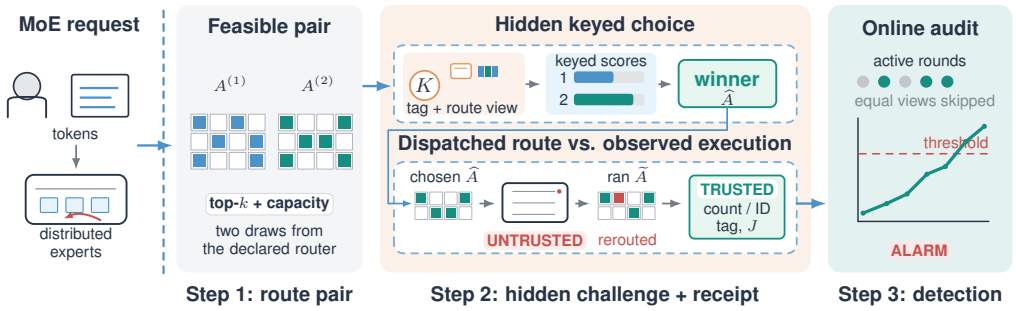


Figure 1: **SentinelMark-MoE turns routed work into a hidden execution challenge.** The selector ranks two feasible routes by a secret score over their observable views and dispatches the winner $\widehat{A}$. The untrusted backend may instead run $\widetilde{A}$; a trusted counter or identity receipt records its view. After the backend commits, the bit J identifies pairs with different views. The auditor skips equal-view pairs and accumulates evidence on the remaining rounds. When the auditor raises an alarm, it signals a changed view rather than an incorrect expert output. Grids and trace are schematic.

Figure 1 shows three stages. SENTINELMARK-MoE draws two valid routes from a declared routing distribution. It scores a summary, or *view*, of each route with a pseudorandom function and runs the higher-scoring route. After execution, a trusted observer supplies the observed view to an online detector. The detector controls false alarms even when the audit stops based on the evidence collected. Views may be per-expert counts or authenticated expert identities; they determine which changes are visible. The *active audit clock* counts only pairs with different views. To detect infrequent changes, we use the full score distribution rather than only its mean in a likelihood-ratio update.

We separately evaluate routing changes, detection, observable attacks, and running cost. Tests cover Qwen1.5-MoE, Mixtral-8×7B, and OLMoE-1B-7B on WikiText-2, MMLU, GSM8K, IFEval, and HumanEval, with a dense Llama-2-70B control. The held-out benchmark uses four fixed groups of route records per task, 8,000 attack audits per main model–task setting, and 20,000 audits with correct execution per false-alarm check. Against 10% native rerouting, every task setting exceeds the preregistered 0.80 detection-probability target after 2,048 active challenges. Against three 20% attacks, every one of 45 model–task–attack settings exceeds 0.90 after 1,024 active challenges. Audit keys quantify detector variation on these fixed workloads; they are not independent task examples.

Our four contributions are:

- We formulate how to audit the route a MoE backend actually executes, using trusted observations rather than the backend's own route logs.
- We design a secret-key rule for choosing between two valid routes and prove when it preserves the chosen routing distribution.
- We develop an online detector that controls false alarms even when stopped early. It skips uninformative pairs and uses the full score distribution to better detect infrequent attacks.
- We evaluate detection across three MoE families, five tasks, and three attack types. We also measure model quality and serving cost, and show how the trusted observation determines which changes can be detected.

2 PROBLEM, SCOPE, AND RELATED WORK

Feasible routing. For one layer group with tokens $t \in [T]$ and experts $e \in [E]$, let $A \in \{0,1\}^{T\times E}$ record which experts each token uses. It is feasible when each token uses k experts, expert e receives at most capacity C_e, and policy masks are respected. We write the feasible set as $\mathcal{A}$ and let the declared native sampler $\mathcal{D}$ be any distribution supported on $\mathcal{A}$. It may arise from balanced transport, expert choice, or a softened near-tie adapter (Lewis et al., 2021; Zhou et al., 2022; Clark et al., 2022; Cuturi, 2013). Preserving this sampler differs from preserving a deployed deterministic router. The keyed selection can preserve $\mathcal{D}$ exactly even when the adapter that defines $\mathcal{D}$ changes the original routing behavior.

Observation and attacker. We summarize each assignment through an observation map $\phi : \mathcal{A} \to \mathcal{O}$, using evidence from an independent trusted channel. We evaluate per-expert counts and authenticated logical expert identities. The backend sees the selected assignment but not the current key, losing candidate, score, or bit indicating whether the candidates have different views before it commits to its execution. It may reroute, omit, replay, or substitute expert work, and may adapt from public history. The auditor can detect a change only when the executed assignment $\widetilde{A}$ changes the authenticated view $\phi(\widetilde{A})$. Attacks that preserve this view are invisible to the auditor, regardless of the detector used.

This model checks whether the observed route matches the selected route. Checking that experts used the intended weights or returned correct outputs requires additional evidence, such as output commitments, attestation, or verifiable computation. The formal route-only impossibility statement appears in Theorem 3.3.

Choosing comparable baselines. Output watermark detectors observe generated tokens; routing-based watermarks support text identification or model ownership (Kirchenbauer et al., 2023; Sun et al., 2026; Gao et al., 2026). Their detectors therefore use different evidence from an execution auditor. With the same auditor inputs, we compare the original detector with four betting fractions, an active-clock online Newton-step (ONS) bettor, and our conditional likelihood-ratio process. These are detector variants, not independent systems. For direct comparison, we pair an authenticated digest of the expected view with the same trusted observer. This detects the first changed view, subject to hash security. It transmits more selector metadata than the informative-pair bit, but needs neither keyed selection nor evidence accumulated over many rounds. Its trade-off with statistical auditing is assessed separately from detection performance.

Anytime-valid evidence. Let a pseudorandom function return an integer $R \in \{0, \dots, L-1\}$. We use the centered score

$$s(R) = \frac{R + 1/2}{L}, \qquad x(R) = 2s(R) - 1. \tag{1}$$

We call a nonnegative process an e-process if its expectation under the null is at most one at every bounded stopping time; a nonnegative test martingale starting at one is sufficient. Ville's inequality then controls the probability that it ever crosses $1/\alpha$, even with adaptive stopping (Shafer & Vovk, 2019; Ramdas et al., 2023; Waudby-Smith & Ramdas, 2024). This matches online serving, where audit lengths and stopping times vary.

3 Method and Guarantees

For each routing group, the trusted selector draws $A^{(1)}, A^{(2)} \overset{\text{i.i.d.}}{\sim} \mathcal{D}$, forms a fresh tag from a unique request identifier, model version, layer, and context, and computes

$$R_i = \mathsf{F}_{K_\tau}(\text{tag} \,\|\, \phi(A^{(i)})).$$

It executes the candidate with the larger score, breaking score ties uniformly. The one-bit indicator $J = \mathbf{1}\{\phi(A^{(1)}) \neq \phi(A^{(2)})\}$ is released to the auditor only after the backend commits to its execution. We derive tenant keys using separate input labels (Krawczyk, 2010); expert identities include the model version, layer, slot, and replica class.

Algorithm 1 Keyed route selection

Require: sampler $\mathcal{D}$, view map ϕ, tenant key K_τ, unique request ID u
Require: metadata $m = (\text{version}, \text{layer}, \text{context})$
$t \leftarrow \text{Encode}(u, m)$
for $i \in \{1, 2\}$ **do**
  $A^{(i)} \sim \mathcal{D}$ ▷ independent draws
  $v_i \leftarrow \phi(A^{(i)})$
  $R_i \leftarrow \mathsf{F}_{K_\tau}(t \| \text{Encode}(v_i))$
end for
$J \leftarrow \mathbf{1}\{v_1 \neq v_2\}$
$S \leftarrow \arg\max_{i \in \{1,2\}} R_i$
$I \sim \text{Uniform}(S)$ ▷ uniform tie break
Dispatch only $A^{(I)}$ to the backend
Wait for execution commitment bound to t
Authenticate and release (t, J) to auditor

If the router supplies a fractional routing plan $X^\star$ that respects capacity, we can express it as a mixture of valid assignments: $X^\star = \sum_j \lambda_j A_j$ with $A_j \in \mathcal{A}$. Sampling index j with probability λ_j gives $\mathbb{E}[A] = X^\star$ and never violates a routing constraint. Production top-k routers often supply only a fixed assignment. Our adapter allows random choices among experts whose gate scores fall within tolerance η of the top-k boundary. The

preservation theorem compares keyed selection with this sampler. Experiments separately measure how much the sampler changes the original router.

All experiments use two candidates, whose selection preserves the sampler as stated in Theorem 3.1.

From the trusted executed view, the auditor recomputes $X_n = x(R_n)$. When a fresh view is chosen without the key, its score has conditional mean zero. This gives the presence e-process

$$\mathcal{W}_n^+ = \prod_{i=1}^{n}(1 + \lambda_i X_i), \qquad \lambda_i \in [0, 1]. \tag{2}$$

Under conforming pairwise selection, the exact finite-range mean is $d_i = \beta_L J_i$, where $\beta_L = (1 - L^{-2})/3$. This gives the manipulation process

$$\mathcal{W}_n^- = \prod_{i=1}^{n}\big(1 + \lambda_i(d_i - X_i)\big), \qquad 0 \le \lambda_i < 1. \tag{3}$$

In both tests, bets are chosen before seeing the current score, and the audit may stop based on the evidence observed so far.

The linear update in Equation (3) uses only the mean score. On an active round ($J_n = 1$), the score of a correctly executed winner has probability $q_{\max}(r) = (2r + 1)/L^2$. By contrast, choosing a fresh view without the key gives mass $1/L$, while the losing candidate has mass $q_{\text{lose}}(r) = (2(L - r) - 1)/L^2$. Let ρ be the modeled attack rate and π the fraction of replacements with fresh views. We use

$$e_{\rho,\pi}(r) = (1 - \rho) + \rho\left[\pi\frac{1/L}{q_{\max}(r)} + (1 - \pi)\frac{q_{\text{lose}}(r)}{q_{\max}(r)}\right]. \tag{4}$$

The factor is one when $J_n = 0$. We combine the cumulative products for 18 fixed (ρ, π) pairs with nonnegative weights that sum to one. Under correct execution, this gives an anytime-valid test martingale. Skipping identical-view pairs avoids updates that carry no selection signal, while the exact score distribution provides more information than its mean alone.

Identical candidate views, called collisions, are especially frequent during Mixtral token generation. We report both informative (active) rounds and the total (raw) routing groups needed to obtain them. The selector releases the authenticated J_n after the backend commits, so the backend cannot choose its action after learning whether the round is informative.

For any fixed pair with different summaries, fresh random scores and fair tie breaking give each candidate the same chance to win. If the summaries match, the scores match too, and a fair tie break gives each candidate half the probability. Averaging over pairs therefore recovers the original routing distribution. The audit uses a different property: among distinct summaries, correct selection favors larger scores. Replacing the winner with the losing candidate or a fresh, key-free view changes this score distribution. Identical-summary pairs have no such selection preference, so we skip them. The likelihood test then uses the probability of each observed score rather than just its average. Selection can therefore preserve routing while creating useful detection evidence.

3.1 Main theoretical results

Assume an honest sampler and selector, fresh uniquely encoded tags, and trusted reports of the executed view and J_n. The backend commits before learning J_n, either score, or the losing candidate. Conditional on past rounds, candidate draws and backend randomness are independent of the current score table. These are the protocol conditions in Section C.1. The exact results below use independent random scores with range $\{0, \dots, L - 1\}$, $L \ge 2$, and fair tie breaking.

Theorem 3.1 (Routing distribution is preserved)**.** *For two independent candidates from $\mathcal{D}$ on $\mathcal{A}$ and any deterministic view map ϕ, Algorithm 1 returns $\mathcal{L}(\hat{A}) = \mathcal{D}$. Thus routing constraints and the number of expert calls are preserved. For an efficient sampler and view map that can be simulated without the key, replacing random scores and ties by the PRF gives distinguishing advantage at most $\varepsilon_{\mathsf{F}}(3)$.*

Theorem 3.2 (False-alarm control and detection delay)**.** *Under correct execution, both the predictable linear test in Equation* (3) *and the fixed likelihood mixture in Equation* (4) *satisfy*

$\mathbb{P}[\sup_n W_n \geq 1/\alpha] \leq \alpha$ *for* $0 < \alpha < 1$. *For an efficient implementation with a concrete PRF, the bound through any declared polynomial horizon* N *is* $\alpha + \varepsilon_{\mathsf{F}}(4N)$. *For likelihood component* j *with weight* $w_j > 0$ *and* $0 < \rho_j < 1$, *let* $K_t^j = e_{\rho_j,\pi_j}(R_t)$ *on active round* t. *If, before the alarm,* $\mathbb{E}[\log K_t^j \mid \mathcal{H}_{t-1}] \geq \kappa_j > 0$ *for every active history, then in the random-score model*

$$\mathbb{E}[\tau_{\mathrm{LR}}] \leq \frac{\log(1/(\alpha w_j)) + b_j}{\kappa_j}, \qquad b_j = \max_r \log e_{\rho_j,\pi_j}(r).$$

Here τ_{LR} *counts active rounds. The raw-round bound replaces* κ_j *by* $p\kappa_j$ *when every pre-alarm raw history has active-round probability at least* $p > 0$ *and conditional active log growth at least* κ_j.

Theorem 3.3 (Observable changes reduce the expected score)**.** *Let* $I_n = \{\tilde{o}_n \neq \hat{o}_n,\ J_n = 1\}$, *where* $\hat{o}_n$ *and* $\tilde{o}_n$ *are the selected and executed views. Define* $q_{\mathrm{inf},N} = N^{-1}\sum_{n=1}^{N} \mathbb{P}[I_n]$ *and* $D_N = N^{-1}\sum_{n=1}^{N} \mathbb{E}[\beta_L J_n - X_n]$. *Then* $D_N \geq \beta_L q_{\mathrm{inf},N}$, *where* $\beta_L = (1 - L^{-2})/3$. *For a concrete PRF and efficient backend, subtract at most* $(2 + \beta_L)\varepsilon_{\mathsf{F}}(4N)$ *at a declared polynomial horizon. Two executions with identical route transcripts have identical alarm distributions for any route-only auditor, even if their output correctness differs.*

Section C gives the expanded statements, including the fixed-linear-test delay bound and collision scaling. All proofs are in Section D.

Only the winning candidate is dispatched, so the method adds no expert execution and no extra all-to-all expert traffic. It does add a second feasible sample, two keyed scores, and a trusted observation channel. Our prototype implements these steps in Python, and the serving results measure that prototype.

The sampler/selector, key derivation, and independent observation channel must be trusted. The distributed expert backend remains untrusted. Signing the selector's log provides weaker evidence because the log is produced before execution; a trusted counter or authenticated identity receipt is measured after dispatch. Keys are separated by tenant and epoch, while request, version, layer, and context fields separate PRF inputs. The 128-bit request identifier also includes the logical worker shard and boot identity so restarts cannot silently reuse a tag.

4 Evaluation

4.1 Design

Goals and models. We ask four questions: whether the selector preserves the chosen routing distribution; whether the detector finds infrequent route changes on real MoE traces; which attacks the recorded summaries reveal; and what the current implementation costs. We use Qwen1.5-MoE-A2.7B, Mixtral-8×7B-Instruct, and OLMoE-1B-7B-Instruct with vLLM on H100 GPUs. We also run dense Llama-2-70B to check that the quality and serving measurements are unchanged when there is no MoE selector.

Datasets. We evaluate WikiText-2, MMLU, GSM8K, IFEval, and HumanEval. To measure model quality, we run the original model and SentinelMark on the same examples. WikiText-2 uses complete 1,024-token sliding windows; the four generation tasks use 100 fixed test records per model. Qwen and Mixtral use near-tie tolerance $\eta = .05$. Our initial OLMoE run exceeded the 1% perplexity limit. We selected $\eta = .01$ on separate validation data before examining the OLMoE test results.

Security benchmark. We use four fixed groups of held-out records per model and task. The main setting applies 10% native rerouting, stops after 2,048 active challenges, and uses 2,000 audit keys per block (8,000 per model–task setting). Additional settings apply rerouting, omission, or authenticated identity substitution to 20% of decisions and stop at 1,024 active challenges. To check false alarms, we run 20,000 audits for each model, observation, and $\alpha \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. All main attack results use $\alpha = 10^{-3}$. The analysis rejects incomplete result files, overlapping splits, hash mismatches, or pooling that hides a model's failure. Further protocol and source details are in Section E.

Baselines. The detectors receive the same executed ranks, collision indicators, keys, and stopping budget. The *original* baseline mixes four fixed linear bets. Its *collision-gated* variant skips unin-

Table 1: **Main detection comparison across models and tasks.** Values are percentages for 10% native rerouting at 2,048 active challenges, with four frozen blocks and 8,000 audits per class in each model–task cell. Power uses the fixed threshold $1/\alpha = 1{,}000$.

Model	Detector	WikiText-2			MMLU			GSM8K			IFEval			HumanEval		
		Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑
Qwen	Original	0.70	68.62	3.65	0.78	70.80	3.62	0.99	71.30	3.73	0.78	68.90	4.38	1.71	74.43	6.56
	Collision-gated	21.29	94.24	43.50	21.19	94.18	44.80	20.75	94.06	47.32	21.69	93.89	44.60	21.76	94.53	46.94
	Active ONS	20.84	91.27	36.52	21.10	91.25	38.46	20.91	91.07	39.30	21.50	91.14	38.91	21.50	91.90	39.97
	Conditional LR	**86.33**	**99.69**	**93.32**	**86.65**	**99.78**	**95.15**	**86.30**	**99.69**	**93.59**	**85.78**	**99.72**	**93.23**	**86.48**	**99.74**	**93.79**
Mixtral	Original	0.45	58.86	1.19	0.34	59.07	1.08	0.30	58.84	1.26	0.24	59.01	1.32	0.21	58.56	1.23
	Collision-gated	21.14	93.98	44.32	21.61	94.02	42.07	21.26	94.14	46.22	21.32	94.21	42.76	21.22	94.27	46.78
	Active ONS	21.19	91.28	38.65	21.38	91.10	37.14	21.44	91.23	39.16	21.24	91.48	38.41	21.18	91.60	42.00
	Conditional LR	**86.22**	**99.67**	**93.67**	**86.08**	**99.69**	**93.68**	**86.49**	**99.75**	**94.35**	**86.55**	**99.77**	**94.43**	**86.44**	**99.73**	**94.34**
OLMoE	Original	0.29	61.79	2.22	0.55	62.71	2.38	0.43	62.23	2.13	0.39	60.76	1.81	0.44	62.01	2.19
	Collision-gated	20.76	94.41	43.96	21.44	94.33	44.28	21.50	94.17	42.82	21.79	94.12	42.55	21.26	94.34	41.37
	Active ONS	20.61	91.54	38.83	21.49	91.51	38.10	21.59	91.54	38.79	21.74	91.28	35.22	21.38	91.51	35.57
	Conditional LR	**86.40**	**99.66**	**93.93**	**86.92**	**99.74**	**93.63**	**86.48**	**99.73**	**93.58**	**86.71**	**99.71**	**93.14**	**86.95**	**99.69**	**93.07**
Mean	Original	0.48	63.09	2.35	0.55	64.19	2.36	0.57	64.12	2.37	0.47	62.89	2.50	0.79	65.00	3.32
	Collision-gated	21.06	94.21	43.93	21.41	94.18	43.72	21.17	94.12	45.45	21.60	94.07	43.31	21.42	94.38	45.03
	Active ONS	20.88	91.36	38.00	21.32	91.29	37.90	21.31	91.28	39.09	21.49	91.30	37.52	21.35	91.67	39.18
	Conditional LR	**86.32**	**99.67**	**93.64**	**86.55**	**99.74**	**94.15**	**86.42**	**99.72**	**93.84**	**86.35**	**99.73**	**93.60**	**86.62**	**99.72**	**93.73**

AUC: AUROC; pAUC: area below 1% FPR divided by 0.01. Detectors share observations, keys, attacks, and budgets; Mean rows equally average models. Bold/underline: best/second-best per model–task–metric; displayed ties share ranks. Metrics use the corrected score replay; original decision-audit intervals are in Table 5.

formative rounds but keeps the same bets. *Active-clock ONS* also skips collisions and applies an online Newton update. *Conditional LR* is our fixed mixture of 18 likelihood tests from Equation (4). The exact expected-view commitment is reported as a direct-verification reference with additional selector metadata.

Metrics. Detection probability (power) and alarm delay at $\alpha = 10^{-3}$ are the main measures. In a separate replay of the fixed traces, we record the maximum log e-value for each audit and compute AUROC and normalized partial AUROC below 1% false-positive rate as secondary ranking measures.

Uncertainty and serving tests. Intervals use 10,000 repeats that resample record blocks and then audit keys inside each block. Model-quality intervals resample paired records. For serving, 64 fixed prompts are run at loads 1, 8, and 32 with fixed 64-token outputs in native–SENTINELMARK–SENTINELMARK–native (ABBA) order. Prompt identifiers and token counts must match before a pair is accepted. We assess detection, model quality, and serving cost separately.

4.2 USING THE FULL SCORE DISTRIBUTION IMPROVES DETECTION

To test whether the full score distribution improves detection, we compare four detectors using the same inputs across three models and five tasks. As shown in Table 1, conditional LR achieves mean power of 0.865, compared with 0.006 for the original bettor and 0.213 for both collision-gated betting and active-clock ONS. Development tests help explain this gain: Qwen and Mixtral often produce routes with identical summaries, but the original detector still updates on these rounds and uses only the mean score. Skipping these rounds helps, and using the full distribution improves detection further.

To check false-alarm control in the implementation, we run 20,000 audits with correct execution per setting. At $\alpha = 10^{-3}$, we observe 12 and 16 alarms for Qwen Counter and identity observations, 8 and 9 for Mixtral, and 6 and 10 for OLMoE. Every count is below the preset limit of 40, and the checks at 10^{-2} and 10^{-4} also pass. These results support the implementation checks; the theorem provides false-alarm control when the auditor chooses when to stop.

We observe the same detection pattern in the original decision audits (Table 5). All 15 model–task settings exceed the 0.80 detection target, while model means remain below the separate 0.90 target.

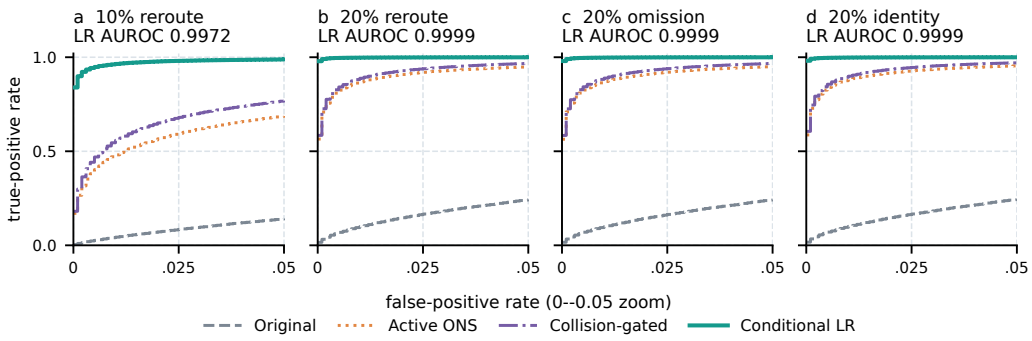


Figure 2: **Conditional LR ranks attacked audits above conforming audits even where false alarms are rare.** Each curve averages 60 model–task–record- block ROC curves, each with 2,000 attacked and 2,000 conforming audits. (a) 10% native rerouting at 2,048 active challenges; (b) 20% rerouting, (c) 20% omission, and (d) 20% identity substitution at 1,024 active challenges. The horizontal axis shows false-positive rates from 0 to 0.05; AUROC values in panel titles use the full 0–1 range. All four detectors see the same records, audit keys, and route observations.

When attacks affect 20% of decisions, detection reaches 0.983–0.992 across all 45 settings. All 18 false-alarm checks pass.

As presented in Figure 2 and Table 6, conditional LR also separates attacked audits from correct execution more effectively. At 10% rerouting, it achieves macro AUROC 0.9972, compared with 0.9419 for the strongest equal-access baseline, collision-gated betting. The gain is larger below a 1% false-positive rate, where normalized partial AUROC is 0.9379 versus 0.4429. For 20% rerouting, omission, and identity substitution, conditional LR reaches at least 0.9999 AUROC and 0.9949 partial AUROC. At 5% rerouting, however, AUROC is 0.9027 while partial AUROC is 0.3891. This shows why good overall ranking must be assessed alongside performance at strict false-alarm thresholds.

To examine sparse attacks further, we vary audit length and attack frequency. As shown in Figure 3b–c, longer audits and more frequent attacks both reduce the miss rate. At 2,048 active challenges, mean power rises from 0.2108 at 5% rerouting to 0.8645 at 10% and 0.9751 at 12.5%. The 32 tested audit lengths show that collecting more evidence helps recover attacks that shorter audits miss.

We also count the total routing groups needed to collect these informative challenges. Qwen requires 2.7–4.1 groups per active challenge, OLMoE 7.3–8.3, and Mixtral 18.3–19.4. Although the median alarm takes about 1,100 informative decisions, Mixtral needs roughly 20,500–21,500 raw decisions. This comparison shows why the active audit length alone does not describe the traffic needed for detection.

4.3 Model quality, visible attacks, and serving cost

To evaluate model quality, we compare native and marked execution on matched examples. As shown in Table 2, the complete WikiText-2 perplexity ratios are 1.006401 for Qwen, 1.002243 for Mixtral, and 1.003436 for OLMoE. Across the four generation tasks, score changes range from -0.070 to $+0.060$ for Qwen and Mixtral and -0.030 to $+0.030$ for OLMoE; every descriptive paired interval includes zero. Notably, the original OLMoE setting at $\eta = .05$ gives a ratio of 1.018601, above the 1% limit. Together, these results show the value of calibrating the routing tolerance separately for each model family.

We next separate the effect of keyed selection from the effect of the router adapter. Over 50 million decisions per condition, total variation (TV) distance between selection and its base sampler is 8.8×10^{-5} for Counter views and 6.7×10^{-5} for identity views. With the hard-router adapter, the distance to its softened sampler is also small (TV 1.55×10^{-4}), but the sampler itself differs from the native router by 0.30 TV. The selection differences are consistent with sampling noise. This comparison

Table 2: **Model quality, serving cost, and live detection.** (a) Four models at loads 1, 8, and 32; native/SM pairs share records and prompts. (b) A live Qwen pilot with eight development requests per condition.

(a) Task quality and serving cost

Model	Mode	Task quality						Load	Serving: fixed 64-token outputs						
		WT-2	MMLU	GSM8K	IFEval		HumanEval		TTFT (ms)↓		ITL (ms)↓		E2E (s)↓		Tok./s↑
		PPL↓	Acc.↑	EM↑	Strict↑	Loose↑	Pass@1↑		p50	p95	p50	p95	p50	p95	$\times 10^3$
Qwen	Native	**7.6685**	**62.0**	**54.0**	**26.0**	**34.0**	**49.0**	1	**23.2**	**30.8**	**21.3**	**21.6**	**1.368**	**1.385**	**0.047**
								8	**42.5**	**48.1**	**21.6**	**21.6**	**1.402**	**1.407**	**0.365**
								32	**59.1**	**81.9**	**21.8**	**21.9**	**1.432**	**1.460**	**1.426**
	SM	7.7176	60.0	53.0	**26.0**	33.0	42.0	1	38.9	49.8	36.3	36.4	2.328	2.343	0.027
								8	73.9	82.0	36.8	37.0	2.392	2.413	0.214
								32	88.8	89.6	37.0	37.0	2.420	2.423	0.846
Mixtral	Native	**3.9352**	70.0	66.0	**46.0**	**50.0**	32.0	1	**25.3**	**29.4**	**21.3**	**21.5**	**1.372**	**1.385**	**0.047**
								8	**89.5**	**128.0**	**23.2**	**23.9**	**1.541**	**1.585**	**0.330**
								32	**264.4**	**300.4**	**25.2**	**25.3**	**1.850**	**1.890**	**1.098**
	SM	3.9440	**73.0**	**68.0**	39.0	45.0	**38.0**	1	43.5	45.5	41.4	41.5	2.654	2.658	0.024
								8	121.4	155.1	41.7	42.4	2.747	2.783	0.186
								32	296.2	312.5	42.2	42.6	2.966	2.977	0.691
OLMoE	Native	**9.0043**	**51.0**	49.0	**35.0**	**40.0**	26.0	1	**16.0**	**18.9**	**14.0**	**14.1**	**0.895**	**0.901**	**0.071**
								8	**31.2**	**37.2**	**14.8**	**15.1**	**0.966**	**0.990**	**0.528**
								32	**41.8**	**45.8**	**15.1**	**16.0**	**0.991**	**1.053**	**2.048**
	SM	9.0352	**51.0**	**50.0**	32.0	39.0	**29.0**	1	26.9	33.0	23.9	24.3	1.535	1.557	0.042
								8	52.5	59.6	24.9	24.9	1.621	1.628	0.316
								32	62.4	63.6	25.1	25.1	1.643	1.647	1.245
Llama-2 70B	Native	**4.3944**	**40.0**	**46.0**	**36.0**	**43.0**	**14.0**	1	**37.8**	45.7	**27.0**	**27.0**	**1.736**	1.748	**0.037**
								8	190.1	**253.4**	**27.5**	**29.6**	1.924	**1.985**	0.266
								32	629.8	667.2	**29.8**	29.9	2.508	2.544	0.812
	SM	**4.3944**	**40.0**	**46.0**	**36.0**	**43.0**	**14.0**	1	37.9	**42.0**	**27.0**	**27.0**	1.737	**1.741**	**0.037**
								8	**185.4**	255.7	**27.5**	**29.6**	**1.921**	1.989	**0.267**
								32	**624.6**	**661.8**	**29.8**	**29.8**	**2.500**	**2.536**	**0.814**

SM: SentinelMark; WT-2: WikiText-2. Scores are % except PPL. TTFT: time to first token; ITL: per-request mean inter-token latency; E2E: request latency; throughput: mean batch rate. Bold/underline: best/second within each model and serving load; displayed ties share ranks (descriptive). Quality uses 100 paired records/task or complete PPL windows. $\eta = .05$ for Qwen/Mixtral, .01 for OLMoE (validation-selected); Llama is an inert dense control.

(b) Live route-fault pilot: Qwen / GSM8K, eight requests per condition

Condition	Trials	Active range	LR	Predictive	Direct	LR delay↓	Direct delay↓
Negative controls: fewer alarms/mismatches ↓							
Conforming	8	406–540	0/8	0/8	0/8	–	–
Count-preserving	8	682–857	0/8	0/8	0/8	–	–
View-changing attacks: more alarms/mismatches ↑							
Reroute (20%)	8	396–540	**5/8**	1/8	8/8	1232	3.5
Identity shift (20%)	8	387–897	**7/8**	0/8	8/8	541	3.5
Decode burst	8	425–888	**8/8**	5/8	8/8	406	193

Each request generates 64 tokens; $\alpha = 10^{-3}$. Delay is the median raw-decision position among detected requests. Direct uses the expected view; bold compares statistical detectors only. Gray rows are negative controls. All 61,440 kernel observations are paired using an in-process software observer.

illustrates the theorem's scope: selection preserves the chosen sampler, while adapting the router can change that sampler.

To test how the observation affects detection, we cyclically move token assignments between experts in an earlier pilot. This changes at least 95.2% of eligible Qwen assignments and 86.3% of Mixtral assignments, but preserves the Counter view. Alarm rates remain close to those under correct execution. When we change only the observation to authenticated expert identities, power reaches at least 0.99 against the same attack. This result confirms that the trusted summary determines which route changes are visible. Checking weights and output values requires additional evidence.

Table 2 also compares serving cost using matched prompts. At fixed load 32, Qwen, Mixtral, and OLMoE retain 0.593, 0.629, and 0.608 of native throughput, while the dense Llama control retains

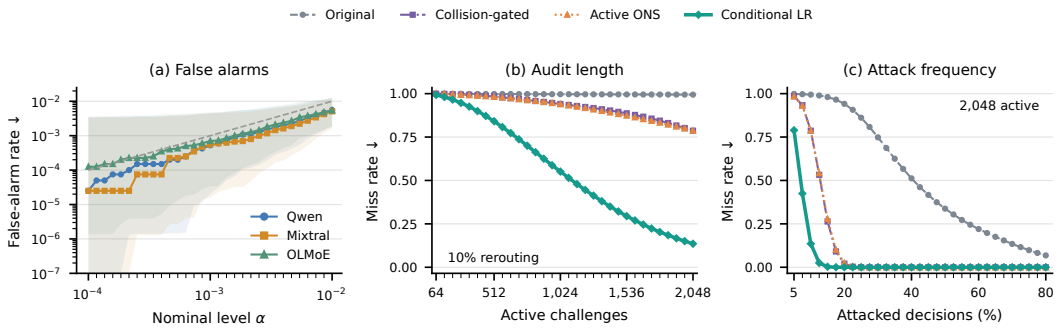


Figure 3: **False-alarm control and fixed-threshold detection.** (a) Conditional-LR false-alarm rates at 31 nominal levels, evaluated on the same 40,000 conforming audits per model at 2,048 active challenges. Shading gives conservative pointwise 95% intervals formed from exact task/block bounds; downward triangles mark upper bounds for zero observed alarms. The dashed line is the nominal rate. (b) Miss rates at 32 active horizons under 10% rerouting. (c) Miss rates at 31 attack frequencies, from 5% to 80%, at 2,048 active challenges. Both use fixed $\alpha = 10^{-3}$ and equal averages over three models and five tasks, with 95% block/key bootstrap bands. Every marker is a measured value; lines connect points without smoothing.

1.002. This points to the MoE selector as the source of the slowdown. Separately, the arithmetic kernel reaches 3.647 billion tournament decisions/s on H100. The kernel result shows that selection can run in parallel; the serving measurements capture the cost of the full prototype.

To check whether the replay findings also appear during generation, we run a live Qwen/GSM8K pilot. As reported in Table 2b, conditional LR detects 5/8 rerouting, 7/8 identity-shift, and 8/8 burst attacks, compared with 1/8, 0/8, and 5/8 for predictive betting. Neither negative control raises an alarm. This small pilot supports the detection pattern observed in replay. Direct comparison detects every view-changing attack using the expected view: the extra route information enables an immediate mismatch check, whereas statistical auditing accumulates evidence over several decisions.

5 Discussion

Our results suggest that MoE routing can support auditing as well as efficient computation. By choosing between valid routes, we build evidence about execution into ordinary inference while preserving the chosen routing distribution. This lets an auditor check whether routing decisions were followed.

What an auditor records determines what it can verify. Detailed records support direct checks, while statistical auditing builds evidence from compact records over time. Choosing between them depends on the trusted information available and the balance between detection delay and record cost. Model quality and serving cost also matter when applying either approach.

Our guarantees rely on trusted selection and observations, with audit scores hidden until execution is committed. They cover changes visible in the recorded summaries; output correctness requires additional evidence. The delay guarantee also requires continued informative evidence during an attack. Extending the theory to weaker trust assumptions and broader execution errors is a useful direction for future work.

6 Conclusion

We present SentinelMark-MoE, which uses ordinary routing decisions to audit expert execution with distribution-preservation and false-alarm guarantees under stated assumptions. Across the tested models and tasks, it improves detection over statistical baselines while keeping measured quality changes small after calibration. These findings support routing-based auditing as a way to strengthen trust in MoE execution, with reliable detection guided by what the auditor can observe.

Reproducibility statement

We provide the method and pseudocode in Section 3, together with the assumptions behind our guarantees; complete proofs appear in Section D. Section 4 describes the evaluation design, and Section E documents data processing, model settings, and experimental procedures. Additional results and hardware/software settings appear in Section A. The supplementary ZIP includes source code, experiment configurations, result summaries, and scripts for checking the reported values and recreating the result plots. Its README explains how to use these materials.

AI use statement

Generative AI tools assisted with language editing, LaTeX formatting, figure and table design, and code generation for manuscript checks. Reported measurements come from archived experiment outputs. The authors checked the displayed values, generated artifacts, claims, and compiled paper.

REFERENCES

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *Conference on Learning Theory (COLT)*, 2024.

Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George van den Driessche, Eliza Rutherford, Tom Hennigan, Matthew Johnson, Katie Millican, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models. In *International Conference on Machine Learning (ICML)*, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis, and Pushmeet Kohli. Scalable watermarking for identifying large language model outputs. *Nature*, 634: 818–823, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Yudong Gao, Qingyue Wang, Yuanyuan Yuan, Ruixuan Huang, Linghan Chen, Zimo Ji, and Shuai Wang. PathMark: Protecting intellectual property of mixture-of-expert LLMs via path watermarks. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2026.

Zahra Ghodsi, Tianyu Gu, and Siddharth Garg. SafetyNets: Verifiable execution of deep neural networks on an untrusted cloud. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

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 (ICML)*, 2023.

Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In *Advances in Cryptology – CRYPTO 2010*, 2010.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *ACM Symposium on Operating Systems Principles (SOSP)*, 2023.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning (ICML)*, 2021.

Tianyi Liu, Xiang Xie, and Yupeng Zhang. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2021.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023.

Glenn Shafer and Vladimir Vovk. *Game-Theoretic Foundations for Probability and Finance*. Wiley, 2019.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Z. Sun, Q. Jiang, S. Sheng, and L. Xiang. WaterMoE: Expert-routing-based watermarking for high fidelity and efficiency. *arXiv preprint arXiv:2607.13099*, 2026.

Florian Tramèr and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *International Conference on Learning Representations (ICLR)*, 2019.

Ian Waudby-Smith and Aaditya Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B*, 86(1):1–27, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Appendix contents

A Extended Experimental Results . . . 12
B Formal Routing Model . . . 16
C Expanded Theoretical Results . . . 17
D Complete Proofs . . . 20
E Implementation and Reproducibility Details . . . 26

A Extended Experimental Results

This appendix extends the two main result tables with comparisons across attack strength and audit budget. We also explain how validation guided the OLMoE setting and report hardware/software settings and mechanism checks.

A.1 Detector components and attack strength

To examine each detector component, we compare the original bettor, collision gating, active online Newton betting (ONS), and the combined conditional likelihood detector (LR) in Table 3. These are matched detector variants, not independently trained models. The 5% and 20% rerouting conditions complement the main table's 10% condition. The recorded primary budgets differ between the two conditions; the comparison of methods is matched within each condition. The sparse case shows why we report both ranking and detection probability: a detector can rank attacks well overall yet miss them at a strict alarm threshold.

Table 3: Detector components under sparse and stronger rerouting.

Model	Attack	Test	WikiText-2			MMLU			GSM8K			IFEval			HumanEval		
			Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑
Qwen	5%	Original	0.19	58.28	1.36	0.15	59.06	1.30	0.24	59.35	1.40	0.06	58.56	1.66	0.24	60.47	1.88
		Gated	1.64	73.16	6.95	1.40	72.98	6.93	1.39	73.15	8.40	1.44	73.29	7.08	1.38	73.39	7.27
		ONS	1.75	69.36	5.53	1.50	69.38	5.81	1.55	69.28	6.36	1.59	69.38	5.92	1.54	69.58	5.73
		LR	**20.94**	**89.85**	**36.74**	**20.69**	**90.33**	**42.42**	**20.82**	**90.55**	**38.31**	**21.52**	**90.25**	**39.18**	**21.25**	**90.12**	**37.77**
Qwen	20%	Original	3.39	79.07	10.24	4.05	81.58	12.01	5.15	81.91	12.70	3.43	79.38	11.69	8.64	86.64	22.79
		Gated	64.27	99.25	80.03	64.22	99.26	81.67	63.79	99.30	83.55	63.61	99.24	82.32	65.24	99.29	83.94
		ONS	63.84	98.83	77.70	63.67	98.85	79.65	63.44	98.85	80.44	63.84	98.76	80.20	64.95	98.81	81.72
		LR	**98.16**	**99.98**	**99.29**	**98.41**	**99.99**	**99.61**	**98.47**	**99.99**	**99.49**	**98.34**	**99.98**	**99.33**	**98.28**	**99.99**	**99.48**
Mixtral	5%	Original	0.24	54.67	0.77	0.16	54.62	0.77	0.21	54.44	0.88	0.20	54.44	0.83	0.11	54.37	0.86
		Gated	1.45	73.18	7.14	1.39	73.29	6.15	1.30	72.90	7.35	1.60	72.90	6.56	1.59	73.63	7.87
		ONS	1.54	69.28	6.20	1.64	69.42	5.50	1.34	68.93	6.00	1.68	68.85	6.08	1.70	69.86	6.80
		LR	**20.78**	**90.22**	**38.46**	**21.24**	**90.29**	**38.05**	**20.97**	**90.65**	**39.21**	**21.56**	**90.28**	**40.39**	**21.31**	**90.63**	**42.47**
Mixtral	20%	Original	0.61	67.37	2.13	0.51	67.60	2.06	0.50	67.27	2.54	0.40	67.63	2.42	0.44	67.20	2.23
		Gated	63.67	99.22	82.19	64.50	99.30	81.16	64.25	99.24	82.81	63.45	99.24	81.46	64.62	99.24	83.21
		ONS	63.40	98.83	80.91	64.26	98.88	79.57	63.88	98.82	80.70	63.60	98.76	80.44	64.77	98.84	82.50
		LR	**98.58**	**99.99**	**99.58**	**98.65**	**99.99**	**99.54**	**98.41**	**99.99**	**99.55**	**98.55**	**99.99**	**99.64**	**98.69**	**99.99**	**99.62**
OLMoE	5%	Original	0.11	55.30	0.99	0.14	55.85	1.03	0.10	55.57	0.99	0.19	54.91	0.95	0.15	55.45	1.03
		Gated	1.40	72.96	7.01	1.39	72.76	7.08	1.45	73.02	6.39	1.59	73.05	6.86	1.65	73.12	6.27
		ONS	1.56	69.09	6.19	1.68	68.91	5.99	1.55	69.68	5.85	1.84	69.34	5.12	1.62	69.22	5.21
		LR	**20.09**	**89.90**	**39.60**	**21.15**	**89.96**	**38.04**	**21.56**	**90.47**	**38.40**	**21.48**	**90.24**	**36.68**	**20.84**	**90.24**	**37.89**
OLMoE	20%	Original	0.60	70.19	3.56	1.03	72.13	3.93	0.53	71.33	3.47	0.59	69.61	3.16	0.66	70.65	3.52
		Gated	63.56	99.31	81.67	64.51	99.26	82.10	64.44	99.28	81.75	64.29	99.29	81.34	63.11	99.26	79.27
		ONS	63.58	98.93	80.78	64.11	98.82	80.06	64.16	98.88	81.32	64.19	98.84	78.37	62.96	98.81	76.48
		LR	**98.34**	**99.98**	**99.43**	**98.46**	**99.99**	**99.43**	**98.58**	**99.98**	**99.50**	**98.44**	**99.99**	**99.52**	**98.43**	**99.99**	**99.40**

Percentages; Pow.=power at $\alpha = .001$, AUC=AUROC, pAUC=normalized area at FPR$\leq .01$; all ↑. **Bold**/underlined: best/second distinct displayed value among four detectors within the same setting; ties share a rank. Pale teal rows identify LR; darker teal cells identify best values, including ties. Four blocks and 8,000 audits per class per model–task cell; no values are fitted. Recorded primary horizons: 5% uses 2,048 active challenges; 20% uses 1,024. Compare detectors within, not across, these two budgets.

A.2 JOINT AUDIT-BUDGET AND ATTACK-FREQUENCY STUDY

We vary audit length and attack frequency together to examine when each detector collects enough evidence. Table 4 reports five audit lengths and four attack frequencies. Each entry gives equal weight to the three model families, five tasks, and four record blocks. All methods use the same attacked and conforming audits within a cell. Reusing keys across horizons supports a matched budget comparison but does not create independent repeats. The short-budget settings show where detection becomes difficult, while longer audits show how each detector benefits from additional evidence. The dense replay in Figure 3 adds 32 horizons at 10% rerouting and 31 attack frequencies at 2,048 active challenges. All 1,560 new shards passed validation, and their score vectors exactly match the original replay at shared horizons. The reducer also verifies 420 reused shards. The 31 false-alarm thresholds reuse 40,000 conforming audits per model; these threshold evaluations complement the separate original calibration campaign described below.

Table 4: Audit budget and attack frequency jointly determine detector performance.

Attack	Test	$H = 128$			$H = 256$			$H = 512$			$H = 1024$			$H = 2048$		
		Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑	Pow.↑	AUC↑	pAUC↑
5%	Original	0.07	53.97	0.77	0.08	54.25	0.81	0.10	54.69	0.85	0.13	55.36	0.96	0.17	56.36	1.11
	Gated	0.03	57.09	1.39	0.14	59.19	1.74	0.29	62.13	2.42	0.64	66.64	3.57	1.47	73.12	7.02
	ONS	0.04	56.12	1.30	0.15	58.02	1.64	0.32	60.57	2.27	0.68	64.07	3.37	1.60	69.30	5.89
	LR	**0.45**	**61.77**	**3.54**	**0.98**	**66.29**	**5.41**	**2.37**	**72.57**	**9.44**	**6.80**	**80.89**	**18.45**	**21.08**	**90.27**	**38.91**
10%	Original	0.11	57.85	1.09	0.14	58.46	1.18	0.20	59.51	1.37	0.33	61.17	1.77	0.57	63.86	2.58
	Gated	0.11	64.45	3.16	0.53	69.17	4.74	1.67	75.75	8.71	5.95	84.82	17.69	21.33	94.19	44.29
	ONS	0.14	62.71	2.86	0.61	67.07	4.35	1.86	73.08	7.87	6.06	81.37	16.58	21.27	91.38	38.34
	LR	**1.91**	**72.35**	**9.45**	**5.34**	**80.17**	**17.46**	**15.90**	**89.32**	**34.83**	**44.88**	**96.76**	**65.31**	**86.45**	**99.72**	**93.79**
20%	Original	0.20	65.39	2.04	0.33	66.78	2.44	0.66	69.24	3.56	2.03	73.31	6.56	5.88	78.88	14.35
	Gated	1.01	78.61	11.37	5.49	86.84	22.00	22.29	94.82	47.43	64.10	99.27	81.90	97.92	99.99	99.35
	ONS	1.11	75.73	10.04	5.74	84.17	20.23	22.40	93.13	43.82	63.91	98.84	80.06	97.58	99.98	99.01
	LR	**11.36**	**87.83**	**30.84**	**33.26**	**95.52**	**57.28**	**74.33**	**99.35**	**87.95**	**98.45**	**99.99**	**99.49**	**100.00**	**100.00**	**100.00**
40%	Original	0.98	78.32	6.13	2.09	80.98	9.03	7.10	85.17	18.00	22.72	90.46	35.09	48.78	94.17	59.59
	Gated	16.50	95.65	52.06	59.80	99.34	83.61	97.03	99.99	99.16	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	ONS	16.64	94.09	47.75	60.01	99.00	81.45	96.68	99.98	98.88	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	LR	**58.33**	**98.63**	**79.96**	**93.71**	**99.93**	**97.92**	**99.97**	**100.00**	**99.99**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**

Percentages; Pow.=power at $\alpha = .001$, AUC=AUROC, pAUC=normalized area at FPR$\leq .01$; all ↑. **Bold**/underlined: best/second distinct displayed value among four detectors within the same setting; ties share a rank. Pale teal rows identify LR; darker teal cells identify best values, including ties. Four blocks and 8,000 audits per class per model–task cell; no values are fitted. Each entry equally averages 60 model–task–block cells. Budgets reuse audit keys; columns are not independent replications.

A.3 ORIGINAL DECISION-AUDIT ESTIMATES AND UNCERTAINTY

Table 1 combines power and score-ranking metrics from the corrected score replay, so each displayed detector uses the same set of audits for all three metrics. Table 5 retains the separate original decision-audit estimates and their hierarchical intervals. Both campaigns use the frozen route blocks; their different samples of audit keys give slightly different point estimates.

A.4 SCORE RANKING AND SPARSE-ATTACK AUDIT LENGTH

To test whether longer audits help distinguish sparse attacks from correct execution, we compare score ranking at different audit lengths in Figure 4. On the same held-out records, scores are maximum log e-values through the active horizon, including initial zero and all inactive-round updates for the original bettor. Each of 60 model–task–block cells uses 2,000 attacked and 2,000 conforming keys; ROC curves are averaged across cells. Uncertainty uses 10,000 block-bootstrap draws within model–task cells, conditional on audit keys. The corrected export includes previously omitted inactive-round updates and passes all 540 checks that scores and alarm decisions agree; the original decision archive is unchanged.

Table 5: **Frozen held-out detection results at $\alpha = 10^{-3}$.** The primary cells use 8,000 audit keys over four independent record blocks; brackets are seeded 95% hierarchical block-bootstrap intervals. Every task clears the preregistered 0.80 cell floor, but none of the three models reaches the 0.90 model-mean target. The 20% rows report the range over five tasks.

Model	Setting	GSM8K	HumanEval	IFEval	MMLU	WikiText-2	Mean/range
Qwen	10% reroute, 2,048 active	0.864 [0.851, 0.878]	0.867 [0.858, 0.877]	0.861 [0.849, 0.873]	0.863 [0.855, 0.870]	0.863 [0.851, 0.875]	0.863†
Mixtral	10% reroute, 2,048 active	0.862 [0.852, 0.871]	0.876 [0.868, 0.884]	0.868 [0.857, 0.881]	0.862 [0.854, 0.870]	0.864 [0.855, 0.873]	0.866†
OLMoE	10% reroute, 2,048 active	0.865 [0.856, 0.874]	0.862 [0.853, 0.871]	0.871 [0.863, 0.880]	0.862 [0.852, 0.871]	0.864 [0.855, 0.873]	0.865†
Secondary 20% attacks at 1,024 active challenges							
Qwen	Native reroute	all five task cells pass the 0.90 floor					0.983–0.985
Qwen	Omission	all five task cells pass the 0.90 floor					0.983–0.987
Qwen	ID substitution	all five task cells pass the 0.90 floor					0.983–0.986
Mixtral	Native reroute	all five task cells pass the 0.90 floor					0.985–0.987
Mixtral	Omission	all five task cells pass the 0.90 floor					0.984–0.987
Mixtral	ID substitution	all five task cells pass the 0.90 floor					0.989–0.992
OLMoE	Native reroute	all five task cells pass the 0.90 floor					0.984–0.987
OLMoE	Omission	all five task cells pass the 0.90 floor					0.984–0.986
OLMoE	ID substitution	all five task cells pass the 0.90 floor					0.983–0.987

†Preregistered aggregate target not met. Observed power is a binomial proportion over audit keys; the hierarchy resamples the frozen record block and then audit keys within block.

Table 6: **Score ranking across frozen attacks.** Macro AUROC and normalized partial AUROC (FPR ≤ 0.01), both higher is better; 60 curves with 2,000 audits per class per curve. Active budgets differ as shown.

Setting	Active	AUROC				Normalized partial AUROC, FPR ≤ 0.01			
		Original	ONS	Gated	Cond. LR	Original	ONS	Gated	Cond. LR
Reroute 5%	2,048	0.5636	0.6930	0.7312	0.9027	0.0111	0.0589	0.0702	0.3891
Reroute 10%	2,048	0.6386	0.9138	0.9419	0.9972	0.0258	0.3834	0.4429	0.9379
Reroute 20%	1,024	0.7331	0.9884	0.9927	0.9999	0.0656	0.8006	0.8190	0.9949
Reroute 40%	2,048	0.9417	1.0000	1.0000	1.0000	0.5959	1.0000	1.0000	1.0000
Reroute 60%	2,048	0.9868	1.0000	1.0000	1.0000	0.8328	1.0000	1.0000	1.0000
Reroute 80%	2,048	0.9984	1.0000	1.0000	1.0000	0.9574	1.0000	1.0000	1.0000
Omission 20%	1,024	0.7320	0.9884	0.9927	0.9999	0.0664	0.8015	0.8193	0.9951
Identity 20%	1,024	0.7338	0.9895	0.9934	0.9999	0.0672	0.8161	0.8340	0.9956

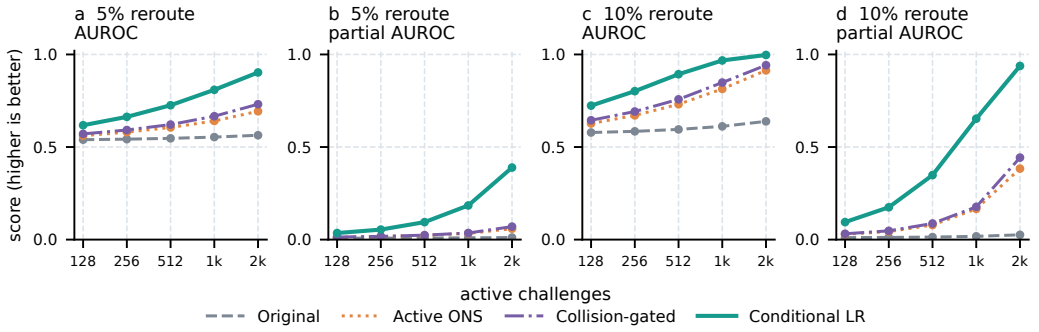


Figure 4: **Longer audits improve the ranking of sparse rerouting.** Panels (a)–(b) use 5% native rerouting and panels (c)–(d) use 10%, with the same held-out records and audit keys at each active horizon. AUROC uses the full false-positive range; normalized partial AUROC covers false-positive rates at most 0.01. Each point is the mean over 60 model–task–record-block curves. At 5%, full AUROC reaches 0.9027 after 2,048 active challenges, while low-false-positive partial AUROC remains 0.3891.

Table 7: **Validated preservation evidence.** TV distinguishes tournament-to-sampler preservation from the hard-router sampler-to-native gap. Qwen passes at the anchor tolerance; OLMoE passes at the tolerance selected on disjoint validation data.

Condition	Validated result
Fractional / Counter	50M decisions; tournament TV 0.000088
Fractional / Identity	50M decisions; tournament TV 0.000067
Hard + softening / Counter	50M decisions; tournament TV 0.000155; sampler–native TV 0.300000
Qwen / WikiText-2	299,077 tokens; PPL ratio 1.003592 [1.003019, 1.004177]
OLMoE / WikiText-2 ($\eta = 0.01$)	288,729 tokens; PPL 9.004301→9.035244; ratio 1.003436

We use 50 million decisions per setting to separate keyed selection from changes introduced by the router adapter. As shown in Table 7, selection with the hard-router adapter remains close to its softened sampler but that sampler has TV 0.30 from the deterministic native router. On Qwen, the complete WikiText-2 bootstrap interval lies below the frozen 1.01 non-inferiority margin. OLMoE at $\eta = .05$ instead gives a perplexity ratio of 1.018601; the separately locked $\eta = .01$ test gives 1.003436 and is the value used in the main table.

Table 8: **Compact receipt costs.** Authenticated selector bytes and CPU sender-plus-verifier time per decision; observer bytes are common to all protocols.

		Selector receipt bytes			Observer	CPU time (ns)		
Model	Batch	SM bit	Digest16	Digest32	bytes	SM bit	Digest16	Digest32
Qwen	1	65.458	76.458	92.458	597.934	3968	5573	5518
Qwen	16	4.215	4.840	5.840	541.316	282	1204	1198
Qwen	64	1.147	1.209	1.459	538.435	97	995	991
Mixtral	1	65.458	76.458	92.458	195.005	3905	5060	5052
Mixtral	16	4.215	4.840	5.840	138.387	270	778	773
Mixtral	64	1.147	1.209	1.459	135.506	84	548	547
OLMoE	1	65.458	76.458	92.458	629.989	4008	5631	5652
OLMoE	16	4.215	4.840	5.840	573.372	281	1244	1250
OLMoE	64	1.147	1.209	1.459	570.491	93	1030	1031

Digest16/32 use 16/32-byte digests; receipts include context and HMAC-SHA256. Times are medians of seven CPU repetitions over 2,048 authentic views. They exclude GPU selection, transfers, networking, storage, and SM score/detector updates; no total-cost ranking is implied. Gray rows mark batch size 64.

In the live pilot in Table 2b, the predictive baseline uses 16 rank bins and Dirichlet pseudocount 0.5. Paired trials under different execution conditions reuse tags but never pool evidence; all faults retain expert-call counts. Eight requests per condition are descriptive, not a false-alarm calibration. Direct comparison finds every tested view change. In Table 8, batched digests narrow the metadata gap; whether the metadata saving outweighs cost and delay remains unresolved.

Earlier arithmetic kernels reach 3.647 Gdec/s (H100 NVL) and 2.073 Gdec/s (L40S); the historical Qwen hook takes $2.821\times$ native time, including synchronization and logging. Separately, 16 epoch keys and 256 tenant/epoch keys yield 4,194,304 request identifiers across 64 shards and eight restarts without duplicates. Reusing a boot ID gives 4,096 duplicates in 8,192 allocations.

A.5 Validation-only adapter sensitivity

Figure 5 measures quality and route coverage on validation data; test-quality evidence comes from the separate held-out run.

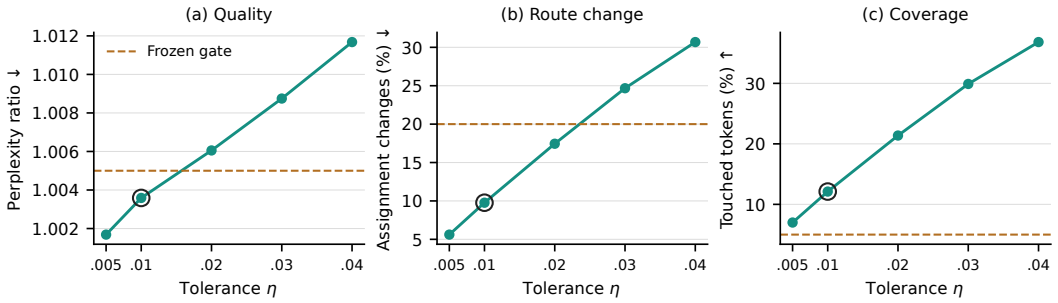


Figure 5: Five measured OLMoE tolerances on 252,480 tokens (493 windows), relative to native execution. Rings select the largest passing $\eta = .01$: PPL ratio ≤ 1.005, assignment changes $\leq 20\%$, touched tokens $\geq 5\%$. Lines join aggregate measurements, without independence assumptions or intervals.

Table 9 records 16 quality/serving configurations. We report requested CPU/RAM resources; CPU model and peak RAM use are unrecorded. Source hashes accompany the table; route-collection budgets appear below.

Table 9: Recorded hardware, software, and run settings.

Setting	Qwen				Mixtral				OLMoE				Llama-2 70B			
	W	M	G	S	W	M	G	S	W	M	G	S	W	M	G	S
GPU type	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
Visible GPUs	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
GiB / GPU	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1	93.1
CPUs requested	24	32	32	32	24	32	32	32	24	32	32	32	32	32	32	32
RAM req. (GB)	200	220	220	220	200	220	220	220	200	220	220	220	400	400	400	400
TP degree	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
EP degree	2	2	2	2	2	2	2	2	2	2	2	2	–	–	–	–
Experts	60	60	60	60	8	8	8	8	64	64	64	64	–	–	–	–
Top-k	4	4	4	4	2	2	2	2	8	8	8	8	–	–	–	–
vLLM	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V
PyTorch	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T
CUDA	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
Python	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P
Linux / libc	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
Precision	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16	BF16
Tolerance η	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.01	0.01	0.01	0.01	–	–	–	–
Context cap	1025	2048	2048	1024	1025	2048	2048	1024	1025	2048	2048	1024	1025	2048	2048	1024
Prompt cap	–	1024	1024	512	–	1024	1024	512	–	1024	1024	512	–	1024	1024	512
Output cap	–	1	512	64	–	1	512	64	–	1	512	64	–	1	512	64
Batch token cap	8192	4096	4096	16384	8192	4096	4096	16384	8192	4096	4096	16384	4096	2048	2048	16384
GPU mem. fraction	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.92	0.92	0.92	0.92
Windows cap	584	–	–	–	584	–	–	–	1000	–	–	–	666	–	–	–
Records / task	–	100	100	–	–	100	100	–	–	100	100	–	–	100	100	–
Repeat batches	–	–	–	4	–	–	–	4	–	–	–	4	–	–	–	4
Warmup batches	–	–	–	1	–	–	–	1	–	–	–	1	–	–	–	1

W=WikiText-2, M=MMLU, G=GSM8K, S=matched serving. H=H100 NVL (compute capability 9.0); V=vLLM 0.12.0; T=PyTorch 2.9.0+cu129; C=CUDA 12.9; P=Python 3.12.12; O=Linux 5.14.0-687.34.1.el9_8 / glibc 2.35. BF16=bfloat16. Blue rows begin hardware, software, and workload groups; gray cells are not applicable. CPUs/RAM are requested resources, not measured utilization. Settings have no best/worst ranking. Serving uses loads 1/8/32 and paired prompts. W uses window/stride 1,024/512. Llama-2 is a dense reference, not a fourth MoE detector.

B Formal Routing Model

For completeness, we state the polyhedral fact used by the feasible sampler before giving the three central guarantees.

B.1 Feasible sampler construction

Let $\mathcal{P}$ be the continuous relaxation of the feasible token–expert assignment constraints with integral token degrees, integral expert capacities, and binary policy masks. Every vertex of $\mathcal{P}$ is integral. Hence every fractional plan $X^\star \in \mathcal{P}$ is a convex combination of feasible integral assignments in $\mathcal{A}$.

The result follows from total unimodularity of the bipartite incidence matrix after adding slack variables for unused expert capacity. The full reduction is included in the proof appendix.

C Expanded Theoretical Results

We distinguish three kinds of guarantee. Exact distribution and false-alarm results use an ideal model in which a truly random function replaces the PRF. We call this the random-function hybrid. Results for the implemented PRF hold against efficient attackers over a stated polynomial number of rounds. Routing constraints hold in both models. This separation is needed: PRF security does not justify a statistical guarantee over an infinite audit.

For group n, write $o_n^{(1)}$ and $o_n^{(2)}$ for the candidate views. Let J_n indicate that they differ. Write $\hat{o}_n$ for the selected view and $\tilde{o}_n$ for the executed view observed by the auditor. Thus

$$J_n = \mathbf{1}\left\{o_n^{(1)} \neq o_n^{(2)}\right\}.$$

Let R_n be the PRF integer at $(\mathrm{tag}_n, \tilde{o}_n)$ and $X_n = x(R_n)$ be its centered score from (1). Define

$$\beta_L = \frac{1 - L^{-2}}{3}. \tag{5}$$

Let $\mathcal{F}_{n-1}$ contain the completed-round transcript, past queried scores, and past choices. It excludes unqueried current-tag values. Within a round we condition on the candidate pair before revealing its scores; this refinement is used only to prove conditional expectations. Backend decisions and betting fractions use only the information permitted below.

C.1 Protocol ordering and trusted inputs

In configuration C1: (i) the selector and sampler are honest; (ii) tags are distinct; (iii) the trusted channel reports $\tilde{o}_n$ and authenticates J_n; and (iv) the backend commits to $\tilde{o}_n$ before learning J_n, any raw score, or the losing candidate. It may see the selected assignment. It has no oracle that evaluates the current keyed score on views of its choice. Conditional on the past, candidates, tags, and backend private randomness are independent of the current score table. Tags and views use an injective encoding. This rules out score-dependent side channels, not only direct key access. Candidate draws are conditionally i.i.d. from the declared sampler; they may depend on past rounds but not current-tag scores. In the ideal model, ties use a fresh fair bit. The implementation derives that bit from a domain-separated keyed input. We count this query and define $\varepsilon_{\mathsf{F}}(q)$ for the joint score/tie primitive. All statements use $L \geq 2$.

These conditions specify what the backend may learn before choosing a route. If it learns numerical scores or can query the current key, it can search for high-scoring replacement views; the proof below does not cover that access.

C.2 T1: exact pairwise preservation

C.2.1 Finite-range score moments

Let $L \geq 2$ and let R be uniform on $\{0, \ldots, L-1\}$ and let R_1, R_2 be independent copies. Then

$$\mathbb{E}[x(R)] = 0, \qquad \mathbb{E}[x(\max\{R_1, R_2\})] = \beta_L.$$

Consequently, under conforming execution in the random-function hybrid,

$$\mathbb{E}[X_n \mid o_n^{(1)}, o_n^{(2)}] = \beta_L J_n. \tag{6}$$

The factor $1-L^{-2}$ is the exact finite-range correction. It is not a continuous-uniform approximation. For a 64-bit range it differs from one by less than 2^{-127}.

Theorem 3.1 states exact preservation and its concrete-PRF version in the main paper. Its proof uses two score queries and at most one independent tie-bit query.

The theorem averages over the fresh score table. Conditional on one realized table, the higher-ranked view is favored. Fresh tags are therefore essential to both the marginal and joint claims.

C.2.2 Routing constraints remain satisfied

The selected assignment lies in $\mathcal{A}$ surely, so its top-k width, capacity, and mask constraints hold deterministically, and it contains exactly kT expert calls. In the random-function hybrid, $\mathbb{E}[\hat{A}] = \mathbb{E}_{A\sim\mathcal{D}}[A] = X^\star$. More generally, every bounded statistic $\psi : \mathcal{A} \to [0,1]$ has the same expectation under $\hat{A}$ and $\mathcal{D}$. For a concrete PRF the same expectation differs by at most $\varepsilon_{\mathsf{F}}(3)$ whenever ψ is efficiently computable, using a randomized distinguisher that outputs one with probability $\psi(A)$.

This does not identify a softened sampler with a deterministic native router. It also fixes only the number of expert calls; it does not assert unchanged latency, device locality, or model quality for an adapter that changes $\mathcal{D}$.

C.2.3 Using more than two candidates

For m candidates, collisions can break sampler preservation: with view law $(0.6, 0.4)$ and $m = 3$, the first output probability becomes 0.576. The coupling bound and tie rule are given in Appendix D; the pairwise theorem does not extend unchanged to deeper tournaments.

C.2.4 Reusing identifiers changes dependence between decisions

For two uniform two-view groups, tag reuse changes the agreement probability from $1/2$ to $5/8 - 1/(8L)$ despite uniform marginals. Appendix D derives this finite-range expression.

C.3 T2: exact anytime-valid manipulation detection

Let $d_n = \beta_L J_n$. The auditor uses the e-process in (3), with each λ_n chosen from past information and J_n, but not from X_n. Define the view-change and informative-change events

$$V_n = \{\tilde{o}_n \neq \hat{o}_n\}, \qquad I_n = V_n \cap \{J_n = 1\}.$$

This distinction matters because, when $J_n = 0$, selection has no keyed preference in the declared view. Changing such a decision need not reduce the expected score.

C.3.1 Key-free changed-view score

Under the protocol conditions in Section C.1 in the random-function hybrid, if V_n occurs then the executed input is either the losing candidate input or a previously unqueried input. Conditional on the backend's allowed information, the former has expected centered score at most 0, and the latter has expected centered score 0. Hence

$$\mathbb{E}[d_n - X_n \mid \mathcal{F}_{n-1}] \geq \beta_L \mathbb{P}[I_n \mid \mathcal{F}_{n-1}]. \tag{7}$$

C.3.2 Expanded statement of Theorem 3.2

Let $0 < \alpha < 1$ and $\tau = \inf\{n : \mathcal{W}_n^- \geq 1/\alpha\}$.

1. *Validity.* In the random-function hybrid, under conforming execution ($V_n = \varnothing$ for all n),

   $$\mathbb{P}[\tau < \infty] \leq \alpha.$$

   This holds for adaptive stopping, variable response length, changing candidate distributions, and predictable betting fractions. With a concrete PRF, for every declared polynomial horizon N, $\mathbb{P}[\tau \leq N] \leq \alpha + \varepsilon_{\mathsf{F}}(4N)$. The same validity statement holds when $\mathcal{W}_n^-$ is replaced by the predeclared likelihood portfolio from (4): conditioned on $J_n = 1$, each

fresh/loser likelihood ratio has null expectation one, and when $J_n = 0$ its factor is one. Combining the cumulative products with fixed nonnegative weights that sum to one preserves this property. Here $4N$ allows two candidate scores, a tie bit, and an executed-view score per round; the implementation and stopping rule must be efficient.

2. *Fixed-linear-test delay.* In this part only, define τ using exactly the linear process with $\lambda_n = \lambda^\star$ below, not arbitrary bets or the likelihood portfolio. Suppose there is an $\eta > 0$ such that, before the alarm, every round satisfies $\mathbb{P}[I_n \mid \mathcal{F}_{n-1}] \geq \eta$. Set

$$a = \beta_L \eta, \quad B = 1 + \beta_L, \quad \lambda^\star = \frac{a}{2B^2}.$$

Then in the random-function hybrid $\mathbb{P}[\tau < \infty] = 1$ and

$$\mathbb{E}[\tau] \leq \frac{4B^2}{\beta_L^2 \eta^2} \left(\log \frac{1}{\alpha} + \log\left(1 + \frac{\beta_L \eta}{2B}\right)\right). \tag{8}$$

3. *Likelihood-portfolio delay.* Index active challenges by t and let τ_{LR} be the first crossing of $1/\alpha$ by $W_t = \sum_j w_j \prod_{i \leq t} K_i^j$. Fix a component with $w_j > 0$, $0 < \rho_j < 1$, and $K_t^j = p_j(R_t)/p_0(R_t)$. Suppose, before the portfolio alarm, $\mathbb{E}[\log K_t^j \mid \mathcal{H}_{t-1}] \geq \kappa_j > 0$, where $\mathcal{H}_{t-1}$ is the active-clock history. With $b_j = \max_r \log[p_j(r)/p_0(r)]$, in the random-function hybrid,

$$\mathbb{E}[\tau_{\mathrm{LR}}] \leq \frac{\log(1/(\alpha w_j)) + b_j}{\kappa_j}. \tag{9}$$

For raw-round delay the same bound holds with denominator $p\kappa_j$ if, conditionally on every pre-alarm raw history, an active round occurs with probability at least $p > 0$ and its conditional expected log factor is at least κ_j. These conditions must hold for each pre-alarm history, not only on average in the long run.

We require the attack condition to hold in each round, given the past. Averaging over the long run would allow an attacker to postpone all changes arbitrarily long, so it cannot ensure a finite expected alarm time.

C.3.3 Stationary collision scaling

Suppose the current pair is i.i.d. from a fixed view law μ, the backend commits independently of the current pair to a view-changing key-free replacement with conditional probability at least ρ, and $\gamma = \sum_o \mu(o)^2 < 1$. Then $\eta \geq \rho(1-\gamma)$, and (8) gives

$$\mathbb{E}[\tau] = O\left(\frac{\log(1/\alpha)}{\rho^2(1-\gamma)^2}\right).$$

The leading coefficient tends to 64 as $L \to \infty$. This is the stationary model used by RQ2. The theorem itself is stronger because it uses realized per-group calibration and allows the workload distribution to change.

C.4 T3: evidence loss and limits of observation

C.4.1 Expanded statement of Theorem 3.3

Use $q_{\mathrm{inf},N}$ and D_N as defined in Theorem 3.3. Under the protocol conditions in Section C.1, in the random-function hybrid,

$$D_N \geq \beta_L q_{\mathrm{inf},N}. \tag{10}$$

Therefore $D_N \leq \epsilon$ implies $q_{\mathrm{inf},N} \leq \epsilon/\beta_L$.

In the stationary, current-pair-oblivious setting of Section C.3.3, a view-changing manipulation rate q gives $q_{\mathrm{inf}} = q(1-\gamma)$ and hence $D \geq q\beta_L(1-\gamma)$. Manipulations beyond the bounded informative rate can evade this conclusion only by preserving ϕ, occurring on a colliding pair, or violating the protocol conditions in Section C.1. For a concrete PRF through horizon N and an efficient backend, subtract at most $(2 + \beta_L)\varepsilon_{\mathsf{F}}(4N)$ from the right-hand side of (10). Furthermore, every randomized route-only auditor has identical alarm distributions on two executions with identical

route transcripts. Consequently it cannot certify semantic correctness when such executions can have different expert-output correctness.

The guarantee concerns only changes visible in the chosen observation. Bounding all assignment changes by the same q would be false for the Counter channel: permutations can change many token-to-expert assignments while preserving the entire count vector.

C.5 Observation boundary and concrete examples

C.5.1 Separation from a signed route record

Consider omission or no-work replay. The backend forwards the selector's valid signed route record but performs no current expert calls. If the auditor sees only that record, it receives identical inputs under honest execution and attack, so its power equals its false-alarm probability. In contrast, if the trusted Counter channel reports current expert work and $kT > 0$, then $\tilde{o}_n \neq \hat{o}_n$ on every attacked round. Under a stationary view law, the informative rate is $1 - \gamma$, and Theorem 3.2 gives

$$\mathbb{E}[\tau] = O\left(\frac{\log(1/\alpha)}{(1-\gamma)^2}\right).$$

If the backend performs dummy work that reproduces the selected Counter view, the attack is view-preserving and this separation no longer applies.

C.5.2 Route-only evidence cannot certify semantic correctness

Theorem 3.3 uses two executions with the same route transcript but different final expert outputs. Appendix D constructs the pair and proves equality of alarm distributions, including randomized auditors. Output binding or attestation adds information absent from this transcript.

C.6 The fully malicious router configuration

Under configuration C2 the adversary chooses the plan as well as the execution. It can run the keyed selector honestly on a plan of its choice, so the C1 theorems certify conformance only to that chosen plan. To audit C2, we would need an external commitment to the feasible set and plan, followed by two checks: the keyed conformance test above and a goodness-of-fit test against the committed distribution. The latter has its own finite-sample assumptions and is not evaluated here; we make no empirical C2 claim.

D Complete Proofs

All probability statements in this appendix are first proved in the ideal random-function hybrid. For a fresh tag, different view inputs receive independent uniform values in $\{0, \dots, L-1\}$. Score ties are broken with independent uniform bits. The concrete-PRF reductions are stated separately.

D.1 Feasible sampler construction

Delete every masked token–expert edge. Construct a directed network with a source s, one node for each token t, one node for each expert e, and a sink z. Give arc (s,t) lower and upper capacity k, arc (t,e) capacity one for each allowed edge, and arc (e,z) capacity C_e.

For any $X \in \mathcal{P}$, put flow X_{te} on (t,e), flow k on (s,t), and flow $\sum_t X_{te}$ on (e,z). The row equalities give flow conservation at each token, the definition of the last flow gives conservation at each expert, and the column inequalities give its capacity bound. Thus X extends to a feasible flow. Conversely, restricting any feasible flow to the token–expert arcs gives a point of $\mathcal{P}$. These two operations are inverse on the token–expert coordinates.

To establish integrality, note that each directed incidence column contains one $+1$ and one -1. Every square submatrix has determinant 0 or ± 1: induct on its order, expanding along any column with at most one nonzero entry. If no such column exists, all column sums are zero, so the matrix is singular. Appending identity rows or their negatives preserves this property by the same expansion. At a

vertex, a full-rank square set of tight conservation and bound constraints therefore has determinant ± 1. Cramer's rule with an integral right-hand side gives integral coordinates. Delete redundant conservation rows first if necessary. The polytope is bounded by the finite arc capacities; when nonempty, it is the convex hull of its vertices. The given $X^\star$ supplies nonemptiness, so no feasibility claim is made for an arbitrary masked instance. This proves the needed total unimodularity and integrality facts rather than assuming an integral sampler. Adding lower and upper capacity bounds preserves the standard integral-flow polytope. Because every capacity and lower bound above is integral, every vertex of the flow polytope is integral. Hence that polytope is the convex hull of its integral flows. Projecting the convex combination onto the token–expert arcs shows that every $X \in \mathcal{P}$ is a convex combination of integral projected flows. Capacity one makes each projected coordinate zero or one; flow conservation gives row sum k; the expert arcs give the column bounds; and deleted arcs implement the masks. The projected integral flows are therefore exactly $\mathcal{A}$. This proves $\mathcal{P} = \mathrm{conv}(\mathcal{A})$ and the finite decomposition claim. □

D.2 Exact finite-range score laws

From (1),

$$x(r) = \frac{2r + 1 - L}{L}.$$

Using $\sum_{r=0}^{L-1} r = L(L-1)/2$,

$$\mathbb{E}[x(R)] = \frac{1}{L^2} \sum_{r=0}^{L-1} (2r + 1 - L) = \frac{L(L-1) + L(1-L)}{L^2} = 0.$$

Let $M = \max\{R_1, R_2\}$. For $r \in \{0, \dots, L-1\}$,

$$\mathbb{P}[M = r] = \mathbb{P}[M \le r] - \mathbb{P}[M \le r-1] = \frac{(r+1)^2 - r^2}{L^2} = \frac{2r+1}{L^2}.$$

Using $\sum r^2 = L(L-1)(2L-1)/6$ and the preceding formula for $\sum r$,

$$\begin{aligned}
\mathbb{E}[M] &= \frac{1}{L^2} \sum_{r=0}^{L-1} r(2r+1) \\
&= \frac{1}{L^2} \left(\frac{L(L-1)(2L-1)}{3} + \frac{L(L-1)}{2} \right) \\
&= \frac{(L-1)(4L+1)}{6L}.
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathbb{E}[x(M)] &= \frac{2\mathbb{E}[M] + 1 - L}{L} \\
&= \frac{1}{L} \left(\frac{(L-1)(4L+1)}{3L} - (L-1) \right) \\
&= \frac{(L-1)(L+1)}{3L^2} = \frac{1 - L^{-2}}{3} = \beta_L.
\end{aligned}$$

If the candidate views coincide, both candidates query the same random-function input and the selected score is uniform, so its centered mean is zero. If they differ, the two scores are independent and the selected score is their maximum; uniform tie-breaking does not change the maximum value. These are precisely the cases $J_n = 0$ and $J_n = 1$, proving (6). □

D.3 Proof of Theorem 3.1

Let $o_i = \phi(A^{(i)})$, let $\mu = \phi_\# \mathcal{D}$, and write $\hat{o} = \phi(\hat{A})$. Because $\mathcal{A}$ is finite, all conditional laws below are ordinary finite conditional probabilities.

First, either candidate wins a distinct-view comparison with probability one half. For distinct o, o', their random-function values R, R' are i.i.d. The events $R > R'$ and $R < R'$ have the same probability. On $R = R'$, the independent tie bit chooses either candidate with probability $1/2$. Thus

$$\mathbb{P}[o \text{ defeats } o'] = \mathbb{P}[R > R'] + \tfrac{1}{2}\mathbb{P}[R = R'] = \tfrac{1}{2}.$$

This identity is exact for finite L.

Next, we show that the winning view has law μ. Fix o. The event $o_1 = o_2 = o$ has probability $\mu(o)^2$ and forces $\hat{o} = o$. If exactly one candidate has view o and the other has view $o' \neq o$, the first step gives conditional winning probability $1/2$. Consequently

$$\begin{aligned}\mathbb{P}[\hat{o} = o] &= \mu(o)^2 + \sum_{o' \neq o} 2\mu(o)\mu(o') \cdot \tfrac{1}{2} \\ &= \mu(o)^2 + \mu(o)(1 - \mu(o)) = \mu(o).\end{aligned}$$

Conditioning on the winning view also preserves the distribution of assignments with that view. For each candidate, the conditional law given $o_i = o$ is $\mathcal{D}(\cdot \mid \phi = o)$. If $o_1 \neq o_2$, the selected index depends only on the two views, their random-function values, and the tie bit, not on which assignment with the selected view was sampled. If $o_1 = o_2 = o$, the tie bit chooses between two conditionally i.i.d. assignments from the same conditional law. Hence in both cases

$$\mathcal{L}(\hat{A} \mid \hat{o} = o) = \mathcal{D}(\cdot \mid \phi = o).$$

Finally, we average over the views. For every $S \subseteq \mathcal{A}$,

$$\begin{aligned}\mathbb{P}[\hat{A} \in S] &= \sum_o \mathbb{P}[\hat{o} = o]\mathbb{P}[\hat{A} \in S \mid \hat{o} = o] \\ &= \sum_o \mu(o)\mathcal{D}(S \mid \phi = o) = \mathcal{D}(S).\end{aligned}$$

Suppose a PPT algorithm distinguishes the real tournament output from $\mathcal{D}$ with advantage larger than $\varepsilon_{\mathsf{F}}(3)$. Give a PRF distinguisher oracle access to either the concrete PRF or a uniform random function. It samples the two candidates, makes two score queries and, on a tie, one separately encoded tie-bit query, runs the tournament, and forwards its output to the assumed distinguisher. With the concrete oracle this is the real experiment. With the random-function oracle, the argument above shows that its output is exactly $\mathcal{D}$. The resulting oracle advantage is the alleged advantage, contradicting the definition of $\varepsilon_{\mathsf{F}}(3)$. □

D.4 Feasibility and bounded statistics

Both candidates lie in $\mathcal{A}$, so selection between them preserves every deterministic feasibility constraint. Summing the row equalities gives exactly kT ones. Theorem 3.1 gives equality of laws in the hybrid, which gives the stated expectations and $\mathbb{E}[\hat{A}] = X^\star$.

For the concrete statement, suppose an efficient $\psi : \mathcal{A} \to [0, 1]$ had an expectation gap larger than $\varepsilon_{\mathsf{F}}(3)$. After receiving A, a randomized distinguisher outputs one with probability $\psi(A)$. Its acceptance probability gap equals the expectation gap, contradicting the concrete part of Theorem 3.1. □

D.5 Why deeper tournaments need not preserve the sampler

Let $o_1, \ldots, o_m$ be i.i.d. from μ, let S be the set of distinct sampled views, let μ be their common law, $\gamma = \sum_o \mu(o)^2$, and let $N_o = \#\{i : o_i = o\}$. By exchangeability of the random-function scores and uniform tie-breaking among winning views, conditional on the sampled views the winning view is uniform on S. Thus

$$\mathbb{P}[\hat{o}_m = o] = \mathbb{E}\left[\frac{\mathbf{1}\{o \in S\}}{|S|}\right], \qquad \mu(o) = \mathbb{E}\left[\frac{N_o}{m}\right].$$

Let D be the event that all m views are distinct. On D the two integrands agree for every o. On D^c they are two probability vectors, whose ℓ_1 distance is at most two. Hence

$$\begin{aligned} 2d_{\mathrm{TV}}(\mathcal{L}(\hat{o}_m), \mu) &\le \mathbb{E}\left[\mathbf{1}\{D^c\} \sum_o \left| \frac{\mathbf{1}\{o \in S\}}{|S|} - \frac{N_o}{m} \right| \right] \\ &\le 2\mathbb{P}[D^c]. \end{aligned}$$

As in Step 3 of the preservation proof, conditional on its output view, the selected assignment has law $\mathcal{D}(\cdot \mid \phi = o)$. These conditional laws have disjoint supports for different o, so the assignment-level TV distance equals the view-level TV distance. Finally, a union bound over the $\binom{m}{2}$ pairs gives

$$\mathbb{P}[D^c] \le \sum_{i<j} \mathbb{P}[o_i = o_j] = \binom{m}{2}\gamma.$$

For the example $m = 3$ and $\mu(a) = 0.6$, all three views are a with probability $0.6^3 = 0.216$ and all are b with probability $0.4^3 = 0.064$. Otherwise both views occur and symmetry makes the winning view uniform on $\{a, b\}$. Therefore

$$\mathbb{P}[\hat{o}_3 = a] = 0.216 + \tfrac{1}{2}(1 - 0.216 - 0.064) = 0.576 \neq 0.6.$$

□

D.6 Exact effect of tag reuse

Call the two views a, b. For their shared score table,

$$\begin{aligned} \mathbb{P}[R(a) > R(b)] &= \mathbb{P}[R(b) > R(a)] = \frac{L-1}{2L}, \\ \mathbb{P}[R(a) = R(b)] &= \frac{1}{L}. \end{aligned}$$

Conditional on $R(a) > R(b)$, one group emits a unless both candidates are b, so its a probability is $3/4$. Two groups use independent candidates, hence their conditional agreement probability is $(3/4)^2 + (1/4)^2 = 5/8$. The same value holds when $R(b) > R(a)$. Conditional on a score tie, each group is marginally uniform and their independent tie bits and candidates give agreement probability $1/2$. Averaging the three cases gives

$$\left(1 - \frac{1}{L}\right)\frac{5}{8} + \frac{1}{L}\frac{1}{2} = \frac{5}{8} - \frac{1}{8L}.$$

Symmetry gives a uniform one-group marginal. Independent tags use independent score tables, so the two uniform output views are independent and agree with probability $1/2$. □

D.7 Attack conditioning and the evidence drift

Fix the history, both candidate assignments, and the backend's private randomness. Let S be the selected candidate index, including the fair tie bit. For distinct views and any symmetric function g of the two ranks, exchangeability and equal splitting of ties give

$$\mathbb{E}[g(R_1, R_2)\mathbf{1}\{S = i\}] = \tfrac{1}{2}\mathbb{E}[g(R_1, R_2)], \qquad i = 1, 2.$$

Thus the maximum and minimum retain their unconditional laws given $S = i$. For coincident views, the common rank is uniform and independent of the tie bit. Conditional on the fixed information and S, the backend's executed view is fixed: it cannot depend further on numerical ranks. Classify this view as unchanged, the other candidate's distinct view, or outside the pair. This classification does not condition on hidden numerical rank values. Conditioning on the pair in the proof does not grant it to the backend.

First consider an informative pair $J_n = 1$. The selected candidate has the maximum of two centered scores. The unselected candidate has the minimum. Since $x(R_1) + x(R_2) = x(\max\{R_1, R_2\}) + x(\min\{R_1, R_2\})$ and each unsorted score has mean zero, the score-law calculation above gives

$$\mathbb{E}[x(\min\{R_1, R_2\})] = -\beta_L \le 0.$$

Every view different from both candidates is a previously unqueried random-function input. Its value remains uniform even after conditioning on the selected assignment, because distinct random-function inputs are independent; its centered mean is zero. The protocol conditions in Section C.1 prevent the backend from conditioning its choice on the hidden losing view, scores, or J_n. Therefore a changed executed view has conditional mean at most zero. More precisely, on an informative pair the expected deficit is 0, $2\beta_L$, or β_L for the unchanged, losing-view, and fresh-view cases. On a colliding pair it is zero in both possible cases. Even when the backend chooses a fresh view adaptively, its rank remains uniform because the choice is independent of all unqueried current-tag values. Averaging these cases handles dependence of the attack on the selected assignment.

If $J_n = 0$, any changed view is unqueried and has mean zero, while $d_n = 0$. If the view is not changed, the score-law calculation above gives mean d_n. Splitting on these cases and applying the tower property yields

$$\mathbb{E}[d_n - X_n \mid \mathcal{F}_{n-1}] \geq \beta_L \mathbb{P}[V_n \cap \{J_n = 1\} \mid \mathcal{F}_{n-1}],$$

which is (7).

D.8 Proof of Theorem 3.2

Refine the filtration within round n so that the candidate views and J_n are revealed before the fresh score values. We require λ_n to be measurable at this intermediate stage. Under conformance, Equation (6) gives

$$\mathbb{E}[1 + \lambda_n(d_n - X_n) \mid \text{past and } J_n] = 1.$$

Moreover $d_n \geq 0$, $X_n < 1$, and $0 \leq \lambda_n < 1$, so every factor is strictly positive. Thus $(\mathcal{W}_n^-)_{n\geq 0}$ is a nonnegative martingale starting at one. For clarity, stop at $\sigma = \min(\tau, m)$ for finite m. The stopped process has expectation one: expand its increments with the past-measurable indicators $\mathbf{1}\{\sigma \geq n\}$. On $\{\tau \leq m\}$ its value is at least $1/\alpha$, and elsewhere it is nonnegative. Hence $\mathbb{P}(\tau \leq m) \leq \alpha$. Taking the increasing union over m gives Ville's inequality without sampling at an unbounded stopping time: $\mathbb{P}[\sup_n \mathcal{W}_n^- \geq 1/\alpha] \leq \alpha$, which is the claimed anytime validity.

Under the distinct-view null, the selected rank has positive mass $p_0(r) = (2r+1)/L^2$. Fresh and losing ranks respectively have masses $p_f(r) = 1/L$ and $p_\ell(r) = (2(L-r)-1)/L^2$. All three sum to one. For predeclared $\rho, \pi \in [0,1]$, define

$$p_{\rho,\pi} = (1-\rho)p_0 + \rho\{\pi p_f + (1-\pi)p_\ell\}, \qquad K_n = p_{\rho,\pi}(R_n)/p_0(R_n).$$

Conditioning on the pre-score information with $J_n = 1$ gives $\mathbb{E}[K_n] = \sum_r p_{\rho,\pi}(r) = 1$. With $J_n = 0$, use $K_n = 1$. Every product $W_n^j = \prod_{i\leq n} K_i^j$ is therefore a nonnegative mean-one martingale. For fixed nonnegative weights summing to one, $W_n = \sum_j w_j W_n^j$ has the same property by linearity. The finite-stopping argument above applies to it. Validity does not require any alternative model to be correct. Selecting weights after seeing the audited ranks is not covered. Delay additionally requires positive conditional log growth, as shown below; validity alone does not imply power.

For a concrete PRF and fixed polynomial horizon N, the event $\{\tau \leq N\}$ is an efficient oracle test. Replacing its concrete oracle by a uniform random function changes its probability by at most $\varepsilon_{\mathsf{F}}(4N)$: simulate two candidate scores, an optional tie-bit query, and at most one new executed-view score per round, reusing values on repeated inputs. The whole adaptive history and stopping decision use this oracle. The hybrid probability is at most α. Therefore $\mathbb{P}[\tau \leq N] \leq \alpha + \varepsilon_{\mathsf{F}}(4N)$.

Let $Z_n = d_n - X_n$. Since $0 \leq d_n \leq \beta_L$ and $-1 < X_n < 1$, $|Z_n| \leq B = 1 + \beta_L$. The drift calculation above and the alternative condition give $\mathbb{E}[Z_n \mid \mathcal{F}_{n-1}] \geq a = \beta_L \eta$ before the alarm.

For $u \geq -1/2$,

$$\log(1+u) \geq u - u^2. \tag{11}$$

To verify it, set $f(u) = \log(1+u) - u + u^2$. Then $f(0) = 0$ and $f'(u) = u(1+2u)/(1+u)$. The derivative is nonpositive on $[-1/2, 0]$ and nonnegative on $[0, \infty)$, so $f(u) \geq f(0) = 0$ on both intervals.

Take $\lambda^\star = a/(2B^2)$. Because $a \leq \beta_L < B$, $|\lambda^\star Z_n| \leq a/(2B) < 1/2$, so (11) applies. It gives

$$\begin{aligned}\mathbb{E}[\log(1 + \lambda^\star Z_n) \mid \mathcal{F}_{n-1}] &\geq \lambda^\star a - (\lambda^\star)^2 B^2 \\ &= \frac{a^2}{4B^2} =: c > 0\end{aligned}$$

before the alarm.

Put $h = \log(1/\alpha)$, $b = \log(1 + \lambda^\star B)$, and $\tau_m = \min\{\tau, m\}$. The stopped sum of log factors has expected value at least $c\mathbb{E}[\tau_m]$. Explicitly, with $\Delta_n = \log(1 + \lambda^\star Z_n)$,

$$\mathbb{E}\left[\sum_{n=1}^{m} \mathbf{1}\{\tau \geq n\}\, \Delta_n\right] = \sum_{n=1}^{m} \mathbb{E}[\mathbf{1}\{\tau \geq n\}\, \mathbb{E}[\Delta_n \mid \mathcal{F}_{n-1}]] \geq c \sum_{n=1}^{m} \mathbb{P}(\tau \geq n) = c\mathbb{E}[\tau_m].$$

The indicator is determined by past information, and the finite sum is integrable since $|\lambda^\star Z_n| < 1/2$. No unbounded optional-stopping premise is used.

Pathwise, the same stopped log wealth is at most $h + b$. If no crossing occurs by m, it is below h. If a crossing occurs, the log wealth immediately before the crossing is below h and the last increment is at most b. Consequently

$$c\mathbb{E}[\tau_m] \leq \mathbb{E}[\log \mathcal{W}^-_{\tau_m}] \leq h + b.$$

Letting $m \to \infty$ and using monotone convergence gives

$$\mathbb{E}[\tau] \leq \frac{h+b}{c} = \frac{4B^2}{\beta_L^2 \eta^2}\left(\log\frac{1}{\alpha} + \log\left(1 + \frac{\beta_L \eta}{2B}\right)\right).$$

The finite expectation implies $\mathbb{P}[\tau < \infty] = 1$. □

Write $S_t^j = \sum_{i \leq t} \log K_i^j$, $h_j = \log(1/(\alpha w_j))$, and $T_m = \min(\tau_{\mathrm{LR}}, m)$. Since $p_0(r) > 0$ on the finite rank space and $p_j(r) \geq (1-\rho_j)p_0(r)$, each increment lies in $[\log(1-\rho_j), b_j]$ and is integrable. The pre-alarm indicator is measurable from the active past. Consequently,

$$\mathbb{E}[S^j_{T_m}] = \sum_{t=1}^{m} \mathbb{E}\Big[\mathbf{1}\{\tau_{\mathrm{LR}} \geq t\}\, \mathbb{E}[\log K_t^j \mid \mathcal{H}_{t-1}]\Big] \geq \kappa_j \mathbb{E}[T_m].$$

Before the portfolio crosses, $w_j \exp(S_t^j) \leq W_t < 1/\alpha$, so $S_t^j < h_j$. At a crossing its final increment is at most b_j. Thus $S^j_{T_m} \leq h_j + b_j$ whether or not a crossing occurs by m. Combining these inequalities and taking monotone convergence of T_m proves (9). The finite expectation also proves almost-sure crossing. This argument does not assume that component j triggers the alarm.

For the raw clock, use log increment zero on inactive rounds. Conditional on each pre-alarm raw history its expected increment is at least $p\kappa_j$ under the stated two conditions. The same stopped-sum and overshoot argument therefore proves the raw bound without assuming independence of waiting times. In a stationary active alternative with rank law q,

$$\kappa_j = \sum_r q(r) \log \frac{p_j(r)}{p_0(r)} = D_{\mathrm{KL}}(q\|p_0) - D_{\mathrm{KL}}(q\|p_j),$$

with zero-mass terms interpreted as zero. Hence a grid component gives this delay guarantee only when its mismatch from q is smaller than the null's. The fixed mixture contributes $\log(1/w_j)$ to the bound. Knowing the view-change rate alone does not establish positive growth for an arbitrary likelihood grid. □

D.9 Stationary collision scaling

Two i.i.d. views collide with probability γ, so $\mathbb{P}[J_n = 1] = 1-\gamma$. By assumption, the backend commits to a view-changing replacement independently of the current pair with conditional probability at least ρ. Therefore

$$\mathbb{P}[I_n \mid \mathcal{F}_{n-1}] \geq \rho(1-\gamma).$$

Apply Theorem 3.2 with $\eta = \rho(1-\gamma)$. All factors outside η^{-2} are independent of ρ and γ, giving the displayed order. Finally $\beta_L \to 1/3$ and $B \to 4/3$, so $4B^2/\beta_L^2 \to 64$. □

D.10 Proof of Theorem 3.3: evidence loss

Take expectations in (7) and use the tower property:

$$\mathbb{E}[d_n - X_n] \geq \beta_L \mathbb{P}[I_n].$$

Sum this inequality from 1 to N and divide by N to obtain $D_N \geq \beta_L q_{\mathrm{inf},N}$. If $D_N \leq \epsilon$, division by the positive number β_L gives $q_{\mathrm{inf},N} \leq \epsilon/\beta_L$.

Under the stationary independence assumptions,

$$\mathbb{P}[I_n] = q\mathbb{P}[J_n = 1] = q(1-\gamma).$$

This gives the stated specialization. The three listed escape cases are exhaustive relative to the proof: a manipulation is either view-preserving or view-changing; a view-changing manipulation occurs on either $J_n = 0$ or $J_n = 1$; and the last case is exactly the informative event controlled above, provided the protocol conditions in Section C.1 hold.

For the concrete statement, simulate all N rounds with an oracle and form the bounded random variable

$$Y = \frac{1}{N}\sum_{n=1}^{N}(d_n - X_n - \beta_L \mathbf{1}\{I_n\}).$$

Since $0 \leq d_n \leq \beta_L$, $|X_n| < 1$, and I_n implies $J_n = 1$, $-1 \leq Y \leq 1 + \beta_L$. In the ideal model the preceding calculation gives $\mathbb{E}[Y] \geq 0$. By accepting with probability $(Y+1)/(2+\beta_L)$, an oracle distinguisher transfers its expectation with error at most $(2+\beta_L)\varepsilon_{\mathsf{F}}(4N)$. Therefore $D_N - \beta_L q_{\mathrm{inf},N} \geq -(2+\beta_L)\varepsilon_{\mathsf{F}}(4N)$ in the concrete model. Both the drift and informative-change probability must be transferred together; transferring only X_n would not justify the same constant. The reduction uses at most four queries per round and requires an efficient finite-precision implementation. If probability sampling is approximated, its total variation error must be added to the bound. □

D.11 Separation from a selector-only signed record

The signed record is produced by the honest selector and is forwarded without change. Therefore a record-only auditor receives the same value under honest execution and attack. Conditioning on any internal randomness of the auditor shows that its alarm probability is identical in the two cases.

For the Counter channel, every selected feasible assignment has total count $kT > 0$, whereas omission or no-work replay creates total current count zero. Thus V_n occurs on every attacked round. In the stationary model, $\mathbb{P}[I_n] = \mathbb{P}[J_n = 1] = 1 - \gamma$. Apply Theorem 3.2. Dummy work that exactly matches the selected count vector makes V_n false and is therefore outside this conclusion. □

D.12 Proof of Theorem 3.3: observation boundary

Let U denote all private randomness used by auditor $\mathcal{T}$. For any fixed u, the auditor's decision is a deterministic function $T_u(\mathsf{Tr}(e))$. If $\mathsf{Tr}(e_0) = \mathsf{Tr}(e_1)$, then $T_u(\mathsf{Tr}(e_0)) = T_u(\mathsf{Tr}(e_1))$ for every u. Averaging this pointwise equality over U gives equal alarm probabilities. Since the correctness predicate differs on the two executions, no decision rule on the common transcript can accept every correct execution and reject every incorrect one.

For the claimed witness pair, choose any completed conforming execution as e_0. Construct e_1 by leaving all routing choices, tags, selector metadata, and measured route work unchanged, but replacing an expert's returned value in the final routed group after the last route observation. The route-only transcript is unchanged, while the predicate that every expert returned its prescribed value changes from true to false. □

E Implementation and Reproducibility Details

We evaluate on WikiText-2 (Merity et al., 2017), MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), IFEval (Zhou et al., 2023), and HumanEval (Chen et al., 2021). We serve the models with vLLM (Kwon et al., 2023). Table 9 lists the software versions and run settings; the following sections explain how we prepare and check the experiments.

E.1 Code organization

We separate algorithms from experiment scripts to reuse the same implementation. The core modules implement route models, pairwise selection, and the original betting-mixture and conditional

LR detectors. Separate modules manage keys and request identifiers, track result sources, and connect the selector and auditor to the vLLM/Qwen hook. Each research question (RQ) has a runner with a JSON configuration. Before submission, we validate its stage, entry point, device, resources, shard count, acceptance checks, and output policy against the schema.

E.2 Controlled route models

To check the selection mechanism in RQ1, we enumerate four feasible assignments of two tokens to two experts. We use base sampling probabilities

$$(0.4, 0.3, 0.2, 0.1).$$

The identity observation distinguishes all four outcomes. The Counter observation records tokens per expert, so different assignments can share a view.

To separate selection from router adaptation, we compare a deterministic native router $(1, 0, 0, 0)$ with a softened sampler $(0.7, 0.1, 0.1, 0.1)$. We measure total variation distance from tournament to sampler and from sampler to native router.

For RQ2–RQ3, we replay routing groups collected during Qwen, Mixtral, and OLMoE inference on the five public tasks. We first check that duplicate records from expert-parallel execution agree, then keep one record per logical event. For each group, we store the candidate and native Counter views and authenticated identity digests. We also record the inference phase, layer, and tag, together with whether candidate views coincide and how assignments change.

E.3 Detector implementation

In the diagnostic pilot, we use an equal mixture of betting fractions $(0.05, 0.15, 0.35, 0.65)$. We calibrate the centered null mean for each decision as $d_n = \beta_2 J_n$, using the bit that records whether the candidate views differ. Each pilot setting uses 200 independent audit keys. We identify the pilot's dense horizon figure as development evidence.

For held-out evaluation, we use the full distribution of the observed scores. The detector leaves wealth unchanged when $J_n = 0$. When $J_n = 1$, each component multiplies its wealth by the exact finite-candidate likelihood ratio for one of 18 alternatives: $\rho \in \{.025, .05, .10, .20, .40, .80\}$ crossed with $\pi \in \{0, .5, 1\}$. We combine these components with equal prior weights, compute in log space, and use an alarm threshold of $1/\alpha$. We fix the portfolio and active-clock ONS update rules before held-out replay; both use only information available before the current score. Attack schedules depend on public tags and attack seeds, independently of the audit key.

For each primary held-out setting, we use 2,000 keys in each of four fixed record blocks. To estimate uncertainty, we resample blocks and then keys in 10,000 seeded bootstrap replicates. We check false alarms under correct execution with 20,000 keys for each model, observation, and $\alpha \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. Before combining results, the analysis requires exactly 89 shards per model and checks each model against the preset criteria.

E.4 Qwen and Mixtral quality evaluation

For the preservation audit in Table 7, we use the Qwen1.5-MoE base model and compare native and SentinelMark negative log likelihoods on matched WikiText-2 windows. Each window contains 1,024 tokens, with stride 512. Model revisions and data hashes are recorded in the run configurations.

To estimate the 95% interval, we use 10,000 circular block-bootstrap replicates with blocks of eight contiguous windows. We fix the seed, these settings, and the 1.01 non-inferiority margin in the analysis configuration before the run.

For the main quality and serving comparisons, we use Qwen1.5-MoE-Chat and Mixtral-8x7B-Instruct. We analyze the Qwen chat-model results separately from the earlier base-model interval. Table 9 gives the shared execution settings; the run configurations record checkpoint revisions, corpus hashes, and seeds.

E.5 OLMoE CALIBRATION AND HELD-OUT EVALUATION

We evaluate OLMoE-Instruct using the settings in Table 9. To test whether routing tolerance transfers across families, we first apply the fixed $\eta = 0.05$ adapter to its top-8 router. The complete WikiText-2 run contains 563 model-tokenized windows and 288,729 scored targets. Window counts depend on the tokenizer, so the 584-window check used for another model does not apply here. The paired-token and expert-parallel consistency checks pass, but the perplexity ratio exceeds the preset 1.01 limit. This motivates calibration for the OLMoE family.

We then compare tolerances using validation data only. Each candidate scores 493 paired windows and 252,480 targets. The preset rule chooses the largest tolerance that passes the validation criteria, giving $\eta = 0.01$. At $\eta = 0.02$, the perplexity ratio is already above the stricter validation limit (1.006053 versus 1.005). Before reading any $\eta = 0.01$ test results, we fix the selected value and the WikiText-2, four-task quality, and fixed-load serving protocols. We also fix held-out route collection, the 89-shard replay, and the analyses of detection and active/raw delay. This keeps tolerance selection separate from test evaluation, with no further tuning on test results.

With $\eta = 0.01$, the WikiText-2 test ratio is 1.003436, and the paired intervals for all four public tasks include zero. For detection, the initial collection provided too few active groups only for MMLU. Before running any replay, we increased MMLU's forced decode length from 32 to 48 tokens and kept the other settings fixed. The replacement collection contains 500 pairs and 640,000 logical groups, including at least 2,468 active groups in every primary block. All 89 replay shards completed. Every 10% task setting exceeds the 0.80 power target, and all 20% attack and calibration settings pass their checks. The mean OLMoE power is 0.865, below the separate 0.90 aggregate target.

E.6 DENSE-MODEL REFERENCE

We use Llama-2-70B-Chat as a dense control, with settings listed in Table 9. We fix the single-user chat template and prompts and record their hashes. We check that each of the four tasks has a complete 100-record shard, uses the fixed official IFEval evaluator where applicable, and runs HumanEval in isolation. The WikiText-2 evaluation includes all 666 paired windows and 341,467 scored targets. As expected for a model without a FusedMoE layer, the paired control records no route groups. We report its quality and serving cost, with route-security cells left empty.

E.7 TIMING AND REQUEST IDENTIFIERS

We measure selection arithmetic on H100 using a vectorized test PRF, with ten warm-up iterations followed by 100 measured iterations per batch size. This isolates the arithmetic from BLAKE3 and model execution. Separately, we compare native and SentinelMark execution on 12 Qwen prompts using two H100s. This diagnostic records Python-level elapsed time; we report it separately from the serving benchmark.

We test request-identifier uniqueness across 64 logical shards and eight boot identifiers. Each 128-bit identifier encodes the deployment, logical shard, boot identifier, and local counter. Repeating a boot identifier provides a negative control to check that the test detects collisions.

E.8 ARTIFACT VALIDATION AND PAPER GENERATION

We save configurations, platforms, metrics, completion markers, and hashes with raw results. Before combining files, we check their schemas, run identifiers, expected ranges and file sets, and hashes. Tables and figures use checked results, with source hashes retained and sources listed in CSV files. We omit missing results rather than estimate them. L40S measurements are included after the pilot, full run, and validation are complete.