When Do Behavior Traces Validate MoE Interventions? Auditing Labels, Selection, and Decision Impact

Anonymous authors
Paper under double-blind review

Abstract

Behavior traces can validate an MoE cache policy when the policy preserves the ordered execution and replay matches its initial state and accounting. Interventions that change expert outputs require evidence about the execution that follows. We study these requirements in Qwen3-MoE, DeepSeek-V2-Lite, and GPT-OSS-20B, separating the accuracy of a candidate's value from the validity of selecting it. Matching execution order and initial state repairs the tested fixed-route labels. Replay and execution still select different windows in 9/40 Qwen records with FP32 routing: modified expert outputs alter the inputs to later routers. Cache history adds a delayed effect, allowing costs to differ even after routes agree. Yet label errors often leave decisions intact. In a separate two-family audit, route-local correction leaves 74/96 labels inexact, yet only 3/32 choices have positive regret, all within 1% of native loads. We construct two runtimes with the same checkpoint and observed behavior but opposite intervention rankings when action-path execution semantics are unobserved. The Trace-to-Action Audit Protocol (TAP) checks the selected action's value, its alternatives, and the loss from choosing it. These checks distinguish a wrong cost estimate from a suboptimal choice and from a loss that exceeds the evaluator's budget.

1 Introduction

MoE serving configurations trade model computation against the cost of loading experts. Candidate changes include substituting a resident expert, using lower-precision routers, and changing the cache policy. One way to compare candidates is to record the unmodified model's routes and expert accesses, then replay each candidate on that trace. For actions that preserve model outputs, matched replay can recover the loading cost exactly. An expert substitution can instead change the hidden states and routes that follow it, leaving the evaluator with the wrong trajectory.

Even checking the chosen candidate's cost can miss a selection error. Figure 1 shows one request from our controlled-FP32 Qwen audit. Replay chooses the *late* substitution window and predicts a saving of 44 logical expert loads. Executing that window saves exactly 44. The prediction is correct, but the discarded *early* window saves 47. Checking all candidates shows that the loss from choosing late is three loads, or 0.159% of the request's 1,881 native loads. The candidate is suboptimal, yet within a 1% regret budget. The opposite pattern also occurs: 30 of 37 inexact winner values in this cohort still lead to an optimal choice (Section 6.2).

These cases require three different checks. We use *label validity* for agreement between a replayed value and execution, *selection validity* for the chosen action's regret, and *decision impact* for the consequences at a specified budget and selection unit. The distinction determines what to measure: executing a winner checks its value, evaluating challengers tests the choice, and a loss budget establishes which errors matter. Table 1 collects the corresponding controls.

Our experiments separate two causes of replay error. A full-sequence logger and incremental deployment can route differently because they use different floating-point computations, even without an intervention. Changing expert outputs adds another cause: later routers receive altered hidden states. Both effects can leave a memory in the cache. Routes may agree again while the cache still

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.

A. Savings (logical loads)

Action	Replay	Executed
abstain	0	0
early	43	**47**
middle	40	44
late	**44**	44

B. Conclusions as evidence is revealed

Evidence	Winner value exact?	Optimal?	Within 1% budget?
Winner only	S	?	?
+ early challenger	S	R	?
Complete set	S	R	S

S: supported; R: refuted; ?: inconclusive.

Paired error 4 $>$ replay margin 1; native-relative regret = 3/1,881 = 0.159%.

Figure 1: **An exact winner value can conceal a better candidate.** The four candidates for controlled-FP32 Qwen pair 4, original view (the first positive-regret record in archived order). The right panel shows what each additional execution establishes. Bold marks column maxima. Early refutes late's optimality; evaluating the complete set establishes regret of three logical loads (0.159% of native loads), within the 1% budget.

contains different experts, so a route discrepancy need not incur its cost at the position where it occurs.

Contributions.

1. We show why checking a selected candidate's value can both miss selection errors and flag harmless label errors. Complete candidate matrices distinguish optimality, budget compliance, and aggregate choice (Sections 6.2–6.3).
2. We isolate execution mismatch, intervention feedback, and delayed cache costs with matched replay, controlled-FP32 routing, and a cache-state swap (Sections 6.1–6.2). These controls identify which part of the evaluation must be repaired.
3. We construct a same-checkpoint example in which the observed behavior, replay ranking, and attainable behavior opportunity remain fixed while deployment rankings reverse. Theorem 1 gives a positive regret lower bound when action-path execution semantics are unobserved (Section 4). TAP organizes the execution evidence needed to check these claims (Section 5).

2 SETTING: TRACES, ACTIONS, AND CLAIMS

Execution and cost model. An MoE checkpoint processes fixed token sequences; at each MoE layer a router selects top-k experts, and a serving-tier cache holds a bounded set of resident experts. A *logical expert load* is one fetch of an expert's parameters charged by the declared cache accounting on this fixed sequence, not a measured physical transfer. The *native loads* of a request are those incurred by the unmodified (no-action) deployment; regret budgets are relative to this quantity.

Traces, actions, and values. A *behavior trace* records the unmodified model's execution—ordered expert accesses, router scores, cache events—under a particular numerical schedule and initial state. An *intervention* (action) a is a declared change applied from a fixed continuation point; its *execution contract* specifies the target, write sites, initial state, and numerical transitions. The *replay value* $\widetilde{v}_a$ is computed from the behavior trace using a's selection and accounting rules; the *deployment value* v_a is measured by executing a; the *replay winner* is $\widehat{a} = \arg\max_a \widetilde{v}_a$, and other candidates are *challengers*. A *record* (request) is one prompt with a fixed continuation; corpus audits draw *source pairs* of requests, each contributing two evaluation *views* (one per member request as intervention target), so 32 pairs yield 64 views.

The three claims. Fix a regret budget $\tau \geq 0$ and a selection unit (per request or aggregate). **C1, label validity**: $\widetilde{v}_{\widehat{a}} = v_{\widehat{a}}$ for the executed action. **C2, selection validity**: $\max_b v_b - v_{\widehat{a}} \leq \tau$ among the declared candidates. **C3, decision impact**: whether the error crosses the declared budget or changes the configuration chosen for a workload. For request i, we record $R_i = \max_b v_{i,b} - v_{i,\widehat{a}_i}$ and the indicator $\mathbf{1}\{R_i > \tau_i\}$; native-relative thresholds use $\tau_i = \alpha C_i^0$, where C_i^0 is native cost. For a shared configuration, values are aggregated before choosing an action. A *native-relative promise* $v_{\widehat{a}} \geq s_{\min}$ is a weaker fourth target: checkable without challengers, and silent about C2.

We evaluate logical loads on fixed token sequences and use continuation NLL as an additional acceptance endpoint. Native-load regret budgets set the scale of the errors; they have no assumed conversion to latency or task quality. Section 4 specifies the information available to the theoretical evaluator, and Section 5.2 gives the experimental protocols.

3 Two failure mechanisms and their selection consequences

Execution mismatch. A behavior trace is bound to its collection schedule and initial state; replaying it under a different schedule, access order, or accounting state can change the value even when the action preserves model outputs. The control is matched ordered replay from the target execution, tested on previously unseen cache actions in Section 6.1.

Action-induced feedback. With common tokens, equal initial states, and matched deterministic pre-action computation, a post-router expert-output action leaves the first acted-on router's proposals unchanged: that router runs before the action, as does the same-layer KV write. Later routers receive hidden states that may have changed. We test this predicted onset in the controlled-FP32 audit (Section 6.2). Matching the collection schedule allows us to separate intervention feedback from execution mismatch.

Write computed scores as $\widehat{z}_\ell(h) = W_\ell h + b_\ell + \eta_\ell(h)$, with global top-$k$ set S_ℓ, behavior gap g_ℓ, state displacement $r_\ell = \|h_\ell^a - h_\ell^0\|_2$, and score displacement $d_\ell = \|\widehat{z}_\ell^a - \widehat{z}_\ell^0\|_\infty$. For $M_\ell = \max_j \|W_{\ell,j}\|_2$ and a valid numerical-error difference bound ϵ_ℓ, $d_\ell \le M_\ell r_\ell + \epsilon_\ell$. The standard margin condition $g_\ell > 2d_\ell$ preserves S_ℓ (Tsuzuku et al., 2018). Ordered dispatch and value require the remaining state and accounting conditions as well. We use measured displacements for ex-post mechanism checks; Appendix Theorem 3 states the assumptions needed for a full-state cascade envelope.

From label error to selection error. Orient values so larger is better and write executed values as $v = \widetilde{v} + \delta$, where $\widetilde{v}$ is replay. If replay selects $\widehat{a}$, define $g_b = \widetilde{v}_{\widehat{a}} - \widetilde{v}_b$. Then

$$\mathrm{Regret}(\widehat{a}) = \max_b \{(\delta_b - \delta_{\widehat{a}}) - g_b\}. \tag{1}$$

The self comparison makes the maximum nonnegative. In the opening example, $\delta_{\mathrm{early}} - \delta_{\mathrm{late}} = 4$ exceeds $g_{\mathrm{early}} = 1$, giving regret three. What matters is the difference between candidate errors: common errors cancel, and regret exceeds τ only when a paired error exceeds $g_b + \tau$. Knowing the winner's error alone leaves this comparison unresolved. For a single configuration chosen across requests, the same identity applies to aggregate candidate values. Averaging requestwise oracle regrets answers a different question. Sections 6.2–6.3 evaluate both with complete matrices.

4 What behavior evidence can identify

Can an evaluator infer an intervention's value from the checkpoint and unmodified traces alone? We construct two execution environments that agree on those observations but rank the interventions differently. The missing information is the numerical computation used on the action path.

4.1 Action-dependent partial identification

Let $P_{0,\theta}(dT \mid \mathcal{I})$ be the behavior-observation law under the contract declared below and $P_{a,\theta}(dT \mid \mathcal{I})$ the law induced by deploying action a in world θ. With C oriented as utility (negative cost when minimizing loads), its bounded value is $v_a(\theta) = \mathbb{E}_{P_{a,\theta}}[C(a, T, \mathcal{I})]$. Behavior observations restrict θ only to the fiber $\Theta_{\mathsf{O}}(P_0) = \{\theta : P_{0,\theta} = P_0\}$, so the sharp identified set is

$$\mathcal{V}_a(P_0) = \{v_a(\theta) : \theta \in \Theta_{\mathsf{O}}(P_0)\}. \tag{2}$$

The set represents uncertainty about the missing action-path information (Khan et al., 2024; Perdomo et al., 2020). Specifying the full runtime can resolve this uncertainty even if replay still disagrees with execution.

Observation contract. All identification statements are relative to $\mathsf{O}_{\mathrm{beh}}$: any number of no-action traces, the entire checkpoint, and quantities computed from the executed behavior path are observable, but the learner may neither query an action-induced transition nor observe which action-specific floating-point reduction schedule the deployment will use; $P_{0,\theta}$ denotes the law of the sigma-field this contract generates. "Behavior-only" restricts information, not analysis depth; observing the complete action law resolves the witness.

The standard identification criterion is constancy of v_a over this set of observationally equivalent worlds; Le Cam's argument gives a $1/2$ worst-world selection-error bound when two such worlds rank actions oppositely (Appendix A.2). Our construction realizes this separation inside one MoE checkpoint, holding both the replay ranking and behavior-oracle opportunity fixed.

Theorem 1 (Same-checkpoint opportunity–transport separation)**.** *Under $\mathsf{O}_{\mathrm{beh}}$, for every fixed-action replay margin $m > 0$ and behavior-oracle opportunity $g > 0$ over the best fixed action, there exist two worlds with the same law under $\mathsf{O}_{\mathrm{beh}}$, the same replay-optimal action and margin m, and the same oracle gap g, but with opposite strict deployment rankings. The worlds are realizable by two successive top-1 MoE blocks with two experts per block, a fixed downstream router, identical behavior-route margins, and identical checkpoint parameters. They differ only in the floating-point reduction tree used by an action-exposed expert computation. The current input is observable and a behavior policy attains the behavior oracle exactly. Every behavior-only randomized rule, including input-dependent policies, has worst-world deployment regret at least*

$$\frac{2g(m+g)}{m+2g} > 0. \tag{3}$$

If the complete joint observation, including input and added runtime information, has two witness laws at total variation at most ρ, the bound is multiplied by $1-\rho$. Thus approximate knowledge of the execution semantic has a quantitative value; checkpoint access alone has none in this witness.

Which evidence resolves the ambiguity? In the integer-load realization (Proposition 3), querying action 1 at x_+ distinguishes the worlds through costs 3 versus 1; querying it at x_- does not. Disclosing the action-path reduction tree also resolves this example; a checkpoint identifier alone does not. A stock-linear CPU realization and open rounding-cell extensions appear in Appendix A.2 and Corollary 1. The theorem concerns missing runtime information. Once the full action semantics are known, expensive evaluation is a computational issue, not this identification ambiguity. Our real-model tests ask whether replay is accurate *within a fixed runtime*.

Ordered fixed-route replay is the point-identified special case. For route $R = (r_1, \ldots, r_T)$ and state $s_{t+1} = F(s_t, r_t, a)$, common R, initial state, transition and accounting imply identical value by induction. An evaluator can check whether the action preserves the ordered experts, numerical outputs and KV transitions; residency-only bookkeeping qualifies. Changing experts, precision or reduction shape may instead alter the trace, so $\mathcal{V}_a(P_0)$ need not be a singleton. Such actions require an identified transition model or change of measure, an invariance argument, or an on-policy audit.

Valid simultaneous bounds on paired errors can also establish budget compliance (Appendix A.2). Our experiments obtain these errors by executing candidates; a cheaper bound would require additional information about the affected computation.

5 Auditing trace-derived claims

TAP is intended for offline evaluation of trace-derived configuration claims. Before reporting that a candidate is optimal or within a loss budget, the evaluator checks the execution evidence supporting that statement. Ordinary complete candidate evaluation is the reference.

Table 1: Evidence required for each evaluation target. The executable report returns supported, refuted or inconclusive for each claim separately.

Claim	Insufficient check	Required evidence / tested control
Fixed-route value (C1)	Same expert sets	Match ordered accesses, initial state and accounting (Section 6.1)
Intervention value (C1)	Matched no-action execution	Execute affected dependencies or establish their invariance (Section 6.2)
Inherited event cost (C1)	Current ordered routes agree	Propagate cache history; the state-swap control isolates a delayed extra load (Figure 2)
Candidate optimality (C2)	Winner's value passes	Check challengers or valid paired-error bounds (Equation (1); Table 13)
Decision impact (C3)	Many flips or strict errors	Evaluate loss at a declared budget and selection unit (Section 6.3)

The Trace-to-Action Audit Protocol (TAP) ties each claim to its required comparisons (Table 1). The procedure below applies to exact values on a declared request or aggregate workload; population inference additionally requires the sampling design described below.

> **TAP procedure.** Declare the candidate set, execution conditions, value to measure, selection unit, and loss budget. Execute the replay winner to check its value. Compare executed challengers with that winner: an advantage above the budget refutes the selection claim. Support it when no advantage exceeds budget and all candidates are covered, or valid bounds exclude any such unqueried challenger; otherwise return inconclusive. Report value agreement, selection regret or its bound, and the resulting budget verdict separately.

Figure 1 is a worked report: the winner's value passes, optimality fails, and the complete comparison passes the regret budget. The supplementary entry in Appendix J reconstructs these verdicts from the candidate matrix and separately checks inherited cache state for offline evaluation.

5.1 Witnesses and population inference

For a source pair, one executed challenger can refute the budget claim if its advantage exceeds the threshold. Our query protocol assigns no bounds to unqueried actions, so accepting the pair requires evaluating every candidate; stopping sooner leaves the result inconclusive. Valid simultaneous bounds offer another route through Equation (7).

Across pairs, the sampling scheme determines the test. The prospective audit uses a finite-population confidence sequence (Waudby-Smith & Ramdas, 2020), uniform sampling without replacement, and exact labels that are independent of query order (Appendix E). Failure to reject before the census leaves validity unresolved. Historical e-process crossings on sorted records lack the required conditional-null justification (Appendix A.4).

Ordinary prefix/KV reuse reduces the cost of these queries by skipping computation the action leaves unchanged. Proposition 6 gives sufficient conditions: fixed tokens, deterministic matched transitions, no upstream writes, and preserved attention-state and ordered-cache semantics. Under them, executing the affected suffix preserves both action values and the audit's subsequent queries and stopping decisions. Our timing comparisons include capture, copying and accounting (Appendices I.1–I.2).

5.2 Instantiation and experimental setup

Models and runtimes. The audit covers Qwen3-30B-A3B, GPT-OSS-20B and DeepSeek-V2-Lite (Yang et al., 2025; OpenAI et al., 2025; DeepSeek-AI et al., 2024) on HuggingFace/A800. Exact-state reuse is tested in Qwen/DeepSeek; native-wrapper parity requires NLL error below 10^{-4} and next-token argmax agreement of at least .999.

Corpora and cohorts. The fixed-route trace audit aligns 512 SICK inputs under full-sequence and token-at-a-time KV-cache execution, then freezes unseen cache actions before matched-runtime replay across three capacities and continuations. Route-changing selection audits use TaPaCo pairs (fixed-route tolerances), frozen equal-depth PAWS pairs (window choice, regret, NLL), and an

OASST cross-family audit; strong-reference cost cohorts use 32 HH-RLHF dialogue pairs (Bai et al., 2022) per architecture ("HH pairs"), two repeats, all six method orders, identical stopping/lazy policies, and separately paid complete matrices (Appendix Tables 46–47), with 64 further excluded HH pairs for a prospective DeepSeek audit.

Action classes. (i) Fixed-route cache actions whose write set is residency-only bookkeeping; (ii) cache rerouting, substituting a sufficiently probable unchosen resident expert under a declared relative tolerance (else the best remaining expert); and (iii) a simulated W4A4 router precision mask (quantize–dequantize selected routers; no packed integer kernels). Actions start only at the fixed continuation; replay updates its own LRU state but retains behavior logits, whereas execution recomputes the suffix. Appendix B.2 specifies slot ordering, candidate updates, model-specific weight normalization, and each experiment's token protocol and layer indices.

Statistical and reporting protocol. Pair bootstrap retains both views and describes sampled-source variation, not population coverage for curated cohorts; legacy binomial bounds require an i.i.d. interpretation (Appendix A.4). Confirmatory audits freeze multiplicity and materiality thresholds before their test cells; the same-cohort threshold sweep is retrospective. Cost comparisons charge capture, execution and overhead. Tables are generated or checked against hashed JSON. Appendix K indexes the evidence; Table 5 summarizes source reuse and confirmation status.

6 What the acceptance checks establish

We first match the logger to deployment, then test what changes when an action alters expert outputs. Complete candidate evaluations let us measure how the resulting errors affect choices and their costs.

6.1 Label validity: execution mismatch and matched-trace repair

Full and incremental execution differ on 6.65% of aligned DeepSeek expert sets and 12.07% of Qwen sets; low full-trace top-k margin predicts these flips with AUROC 0.896 and 0.859. Offline FP32 reevaluation removes 97.35% of Qwen layer-0 flips but only 5–10% at deeper Qwen layers and 5.66% overall in DeepSeek. These results point to shallow near-boundary rounding in Qwen and accumulated hidden-state differences at later routers. Raising the precision of a router alone leaves those input differences in place.

Only 34.65% of old full-trace cache action values reproduce exactly; preserving ordered incremental routes from the actual offload path yields exact simulated and real values for 108 prospectively unseen actions, then 197 across capacities and continuations: the matched trace repairs the tested supervision target. The decision effect is smaller— admission agrees in 98.43% of actions, and the deployment-consistency oracle adds only 0.356% of baseline demand loads at its best plausible point; no configuration reaches the frozen 0.5% materiality threshold.

6.2 Selection validity: action-induced feedback changes rankings

Feedback persists under controlled-FP32 routing. To separate router arithmetic from action-induced feedback, we rerun the Qwen cache intervention token by token with all 48 gates instrumented: gate weights, logits, softmax, and top-k execute in FP32 (291,840 audited calls), and an action-free repeat has identical mean NLL and argmax outputs. Replay still chooses a different depth window in 9/40 records (22.5%; pair-bootstrap 95% interval 12.5–32.5%; one equal-cost tie, eight positive-regret records), with mean deployment regret 1.775 loads (0.375–3.925). Across early/middle/late actions the first acted-on layer has zero mismatches in 400 positions while strictly downstream mismatch is 19.61/16.97/8.92%, each with a positive paired lower bound. Divergence begins after the modified expert outputs can affect the computation, even with FP32 routers. This is a mechanism test on the frozen Qwen sample.

The winner's value is exact in three of the 40 records. One is the opening example: the cost is correct but a challenger saves three more loads. All three fall within the 1% regret budget. Conversely, 30 of the 37 inexact winner values still select an optimal candidate (Table 4A). Checking value agreement would therefore both miss a selection error and flag many choices with zero regret. Appendix Tables 13 and 14 give the complete comparisons and ties.

Table 2: Execution and selection results from two separate cohorts. TaPaCo (20 pairs, full-depth tolerances) supplies fixed-route MAE and drift; PAWS (20 pairs, equal-width windows) supplies choice, regret and NLL columns. Selection maximizes replayed load saving; NLL harm is relative to native on the fixed continuation. Brackets: 95% source-pair bootstrap intervals.

Model	TaPaCo: tolerance sweep		PAWS: window selection		
	Fixed-route value MAE	Route drift range	Window choice mismatch	Deploy regret (loads/request)	Selected NLL harm rate
Qwen3-MoE	0.0	19.8%–31.9%	25.0% [12.5,40.0]	2.6 [0.625,4.88]	50.0% [35.0,65.0]
DeepSeek-V2-Lite	0.0	8.6%–19.7%	30.0% [15.0,45.0]	0.975 [0.25,2.08]	50.0% [32.5,67.5]

Window selection across models. The fixed-route negative control has zero replay/deployment route mismatch, zero action-value error, and zero NLL change in both families. In contrast, resident-aware substitution changes 19.8–31.9% of replayed Qwen routes and 8.6–19.7% of DeepSeek routes after re-execution, with mean absolute load error 9.1–13.2 and 3.7–6.0 despite small signed means. The first acted layer stays aligned. Disagreement starts downstream and decreases for later windows, which leave fewer layers for feedback to propagate.

Whether this changes the choice depends on the candidate set. Replay and execution select the same cost-optimal tolerance in all 40 records per family among the original three tolerances. On the frozen equal-depth PAWS windows, they disagree on 25% of Qwen and 30% of DeepSeek records. Both models have a strictly positive source-pair bootstrap lower bound on regret (Table 2). All three windows save loads on average, but replay sometimes picks a less effective one. Cost-only selection also increases continuation NLL on half the requests.

The OASST audit finds 20/29, 18/29 and 11/29 positive-regret pairs in Qwen/DeepSeek/GPT-OSS. At similar route mismatch, GPT-OSS has larger candidate margins and fewer changed choices, consistent with Equation (1) (Appendix Table 7; 756-view identity check: Appendix A).

A cost-neutral route change can leave a delayed cost. A route change can leave the immediate load count unchanged but different experts in the cache. Figure 2 shows the consequence one event later: identical ordered routes cost one load in replay and two under the intervention. Holding those routes fixed and exchanging only the pre-event cache states swaps the costs from 1/2 to 2/1. The difference comes from cache history. A frozen follow-up finds equal-route cost discrepancies in every model–corpus cell (Table 3).

Correcting costs only where ordered routes disagree misses these delayed effects, leaving 74/96 action labels inexact. Carrying the full cache state forward with ordinary ordered-route replay recovers all 96 labels without adding recorded errors (Appendix C). The remaining label errors change only 3/32 request-level choices, all within 1% regret; the aggregate winner stays unchanged. Cache history matters for exact accounting even when repairing it has little effect on the decision.

On eight natural-response source trees, 16/128-token continuations and two capacities per family give 180/192 inexact labels but only 9/64 positive-regret conditions, one above 1%, and no aggregate-choice changes (Appendix Table 17). The 64 conditions reuse eight sources across models, capacities, and continuation lengths. A single-substitution control holds action traces fixed across capacities on twelve excluded source trees. Discrepancy counts decrease in Qwen (12/12) and increase in DeepSeek (11/12), meeting the predeclared 10/12 directional rule (Appendix Table 18). Thus occupancy can mask or expose costs from the same route differences.

A second action class. On 20 further disjoint pairs per model, the simulated-W4A4 router precision-mask choice fails the recorded-router-slot diagnostic on 5/11/15 pairs in Qwen/DeepSeek/GPT-OSS (Appendix B.3). This experiment measures changes in recorded router slots. A prospective audit checks the score-margin condition using measured displacements (Appendix Table 11); obtaining useful bounds before execution remains open.

6.3 Budget and decision unit change the acceptance question

Table 4B evaluates loss thresholds and selection units on HH matrices, a separate cohort from the FP32 records in panel A. Mean per-view regret is 3.062/1.078 logical loads for Qwen/DeepSeek:

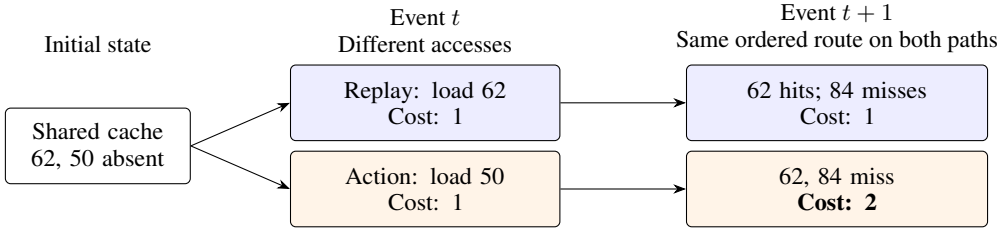


Figure 2: **A cost difference persists after routes agree.** Two consecutive events at downstream Qwen layer 22 have replay/action costs 1/1 then 1/2. The second ordered top-8 route is identical, but the earlier 62-versus-50 access determines whether expert 62 hits. The state-swap control isolates this history effect; propagating the full cache states recovers the action value.

Table 3: Cache memory after route agreement. Per cell, four source pairs (eight views) yield 24 non-null labels; after exact correction at every ordered-route disagreement, many labels remain wrong but few choices do (regret threshold 1% of native logical loads). State-aware checks and ordinary full ordered-route replay both recover all 96 labels, and all aggregate winners agree; inputs are disjoint from the mechanism pilot.

Model/corpus	Equal-route cost-error events	Remaining wrong labels	Positive regret views	Above 1% regret views
Qwen/PAWS	238	20/24	0/8	0/8
Qwen/OASST	208	21/24	1/8	0/8
DeepSeek/PAWS	135	16/24	0/8	0/8
DeepSeek/OASST	128	17/24	2/8	0/8

0.173/0.126% of native loads, but 6.298/5.761% of the best available saving including abstain. These are means of per-view quantities (Appendix Table 25). The same error can therefore be small relative to total cost yet consume a larger share of the saving an intervention offers.

Loss thresholds. In this retrospective comparison, errors above 1% of native loads are rare; lowering the threshold to 0.5% identifies 11/32 Qwen and 5/32 DeepSeek pairs. At 1%, ordinary query ordering already matches the within-protocol oracle (188/192 and 184/192 queries; Appendix F). The prospective 64-pair DeepSeek audit finds four positives at the frozen 1% threshold. It never rejects the hypothesis of at most six positive pairs ($K \leq 6$) and saves no queries (Appendix Table 21; the separate saving-promise test is in Appendix Table 22).

Choosing one configuration for the cohort. Replay and execution select the same aggregate configuration on HH. That agreement does not hold across all our tests: four of eight aggregate optima change in the twelve-source single-substitution control, with small logical-load regret (Appendix C.2; different actions and cohorts). Equation (1) explains the comparison by applying to aggregate values. A fixed choice survives when aggregate paired errors stay within its margins; requestwise oracle regret alone cannot establish whether this happens.

Cost of acquiring this evidence. Ordinary stopping, lazy capture and suffix reuse (TransformerLens Contributors, 2026) explain the audit savings; custom ordering and snapshot reuse add no supported increment (Appendix Tables 34, 46–47).

Adding an NLL budget. A secondary example adds an observed NLL constraint to cost selection. In five PAWS/OASST cohorts (254 views), checking alternatives retains more feasible saving than checking only the winner; cost-ranking repair accounts for 6.3–16.7% of that increment. The remainder comes from candidate coverage. All checks use the same 9–10-token artificial continuation for selection and scoring, without deducting evaluation cost. This is a finite-matrix constraint example, not an independent quality test (Appendix G).

Table 4: Different checks support different conclusions. **A:** complete controlled-FP32 Qwen matrix, 40 views from 20 source pairs; exact/inexact winner rows partition all views, and zero regret includes ties. **B:** separate retrospective HH cohorts, 32 source pairs (64 views) per family; a pair is positive if either view exceeds the threshold, and constant-action choice aggregates all 64 views. Percentages are strict thresholds relative to each view's native logical loads.

A. Winner-value accuracy versus candidate optimality

Winner check	Views	Regret > 0	Regret $= 0$	Regret $> 1\%$
All views	40	8	32	1
Winner value exact	3	1	2	0
Winner value inexact	37	7	30	1

B. Loss budget and selection unit on separate HH cohorts

Family	Pairs $> 0\%$	Pairs $> 0.5\%$	Pairs $> 1\%$	Constant choice changes?
Qwen	20/32	11/32	1/32	No
DeepSeek	13/32	5/32	2/32	No

7 Related work

Trace-driven evaluation. Trace-driven memory simulation has long depended on the fidelity and scope of the recorded execution (Uhlig & Mudge, 1997). We study its MoE-specific consequences: downstream routing and inherited cache state can invalidate candidate values, while paired errors and decision margins determine whether the selected action changes.

MoE evaluation and numerical mechanisms. Engmann et al. (2026) compare population expert statistics with token-level ablation effects. Concurrent studies examine fixed-route accounting and policy rankings (Zhang, 2026), and expert-reduction semantics and state reconstruction (Zhu, 2026). We follow the effects of an intervention through later routes and cache states to ask whether replay still selects the right candidate. R3 (Ma et al., 2025) uses recorded rollout masks to stabilize MoE RL training; its target computation deliberately fixes those routes.

Identification and reliable evaluation. Decision-dependent distributions, no-overlap identification, adaptive evaluation, and counterfactual estimation supply established tools (Perdomo et al., 2020; Khan et al., 2024; Cook et al., 2024; Park et al., 2025; Bottou et al., 2013; Swaminathan & Joachims, 2015; Wang et al., 2017); our construction instantiates the missing-information problem in one MoE checkpoint. Even a policy attaining the behavior oracle cannot resolve the unobserved action-path computation (contrast Shihab et al., 2026; Appendix L).

Directly evaluated interventions. Task-specific systems execute or supervise the modified model (Zhu et al., 2026; Suram, 2026; Yoon et al., 2026; Hu et al., 2026); GEMQ adapts routers to quantized experts (Deng et al., 2026) and Q-Strata evaluates assembled models in its outer search (Lee et al., 2026). These evaluate the intervention itself; we test candidate comparisons inferred from the unmodified model's trace.

Limitations. The theorem assumes unobserved action-path semantics; a known runtime can resolve its ambiguity. Experiments measure fixed-token logical loads, with exact-state reuse tested on Qwen/DeepSeek in HuggingFace/A800. The NLL endpoint uses a prescribed continuation and is uncalibrated to task quality. The audit requires executed candidates; a cheap pre-execution risk predictor remains open. Audit timings measure evaluation cost, not serving speed. Historical free-generation results are exploratory.

Conclusion. Ordered replay validates fixed-route costs when initial state, transitions and accounting match. Output-changing interventions require evidence about their affected execution. Checking the chosen action alone can miss a better candidate, while many inaccurate values preserve the choice. TAP checks alternatives and measures the resulting loss at the declared budget and selection unit, separating a failed estimate from a failed decision.

REPRODUCIBILITY STATEMENT

The supplementary archive provides executable CPU checks and selected archived evidence. Appendix J gives entry points for the same-checkpoint construction, replay selection witnesses, cache-state controls, and cost records. It distinguishes reconstruction from archived results and CPU tests from rerunning pretrained models on GPUs. Proofs and assumptions appear in Appendix A; the supplement provides protocols, mechanism controls and complete comparisons. Model weights are not distributed.

AI USE STATEMENT

Generative AI tools assisted with conceptual/theoretical development and stress-testing, including mathematical claims and proofs; hypotheses and experimental design; code implementation and inspection; data formatting and interpretation; drafting/editing text, tables and figures; and literature discovery and summarization. Reported numerical results come from logged execution, frozen protocols and hashed artifacts. The authors reviewed the AI-assisted material and take responsibility for the final content.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. URL https://arxiv.org/abs/2204.05862.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14: 3207–3260, 2013.

Thomas Cook, Alan Mishler, and Aaditya Ramdas. Semiparametric efficient inference in adaptive experiments. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, pp. 1033–1064, 2024.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable routing strategy for mixture of experts. *arXiv preprint arXiv:2204.08396*, 2022.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, et al. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

Jianing Deng, Song Wang, Dongwei Wang, Zijie Liu, Tianlong Chen, Huanrui Yang, and Jingtong Hu. GEMQ: Global expert-level mixed-precision quantization for MoE LLMs. arXiv preprint arXiv:2605.23078, 2026. URL https://arxiv.org/abs/2605.23078. ICML 2026; arXiv version first submitted May 21, 2026.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Leonard Engmann, Christian Medeiros Adriano, and Holger Giese. From observation to intervention: A causal audit of expert importance in mixture-of-experts models. *arXiv preprint arXiv:2606.10703*, 2026.

Zhiyuan Fang, Zicong Hong, Yuegui Huang, Yufeng Lyu, Wuhui Chen, Yue Yu, Fan Yu, and Zibin Zheng. Fate: Fast edge inference of mixture-of-experts models via cross-layer gate. *arXiv preprint arXiv:2502.12224*, 2025.

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.

Parvel Gu. Detecting a route flip is easier than knowing whether to fix it: Causal route-mediated damage in quantized mixture-of-experts. *arXiv preprint arXiv:2608.11212*, 2026.

Xin He, Shunkang Zhang, Kaijie Tang, Shaohuai Shi, Yuxin Wang, Zihao Zeng, Zhenheng Tang, Xiaowen Chu, Haiyan Yin, Ivor W. Tsang, and Yew Soon Ong. ExpertFlow: Efficient mixture-of-experts inference via predictive expert caching and token scheduling. *arXiv preprint arXiv:2410.17954*, 2024.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. *The Annals of Statistics*, 49(2):1055–1080, 2021. doi: 10.1214/20-AOS1991. URL https://arxiv.org/abs/1810.08240.

Wentao Hu, Yanbo Zhai, Xiaohui Hu, Mingkuan Zhao, Shanhong Yu, Xue Liu, Kaidong Yu, Shuangyong Song, and Xuelong Li. Awakening dormant experts: Counterfactual routing to mitigate MoE hallucinations. *arXiv preprint arXiv:2604.14246*, 2026. URL https://arxiv.org/abs/2604.14246.

Zhiyi Huang, Qinpei Lou, and Tao Xiao. Mixture-of-experts serving. *arXiv preprint arXiv:2607.17880*, 2026. URL https://arxiv.org/abs/2607.17880.

Samir Khan, Martin Saveski, and Johan Ugander. Off-policy evaluation beyond overlap: Sharp partial identification under smoothness. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 23734–23757, 2024.

Cyrille Kone, Emilie Kaufmann, and Laura Richert. Constrained pareto set identification with bandit feedback. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 31342–31378, 2025.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot moe compression. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ukGxWd2aDG.

Deokjae Lee, Sihun Chu, and Hyun Oh Song. Q-Strata: Hierarchical bit allocation for mixed-precision quantization of mixture-of-experts LLMs. arXiv preprint arXiv:2608.30564, 2026. URL https://arxiv.org/abs/2608.30564. EMNLP 2026 main-conference long paper; arXiv version first submitted August 31, 2026.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. R2-T2: Re-routing in test-time for multimodal mixture-of-experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 35292–35316, 2025.

Sirui Lin, Zijun Gao, Jose Blanchet, and Peter Glynn. Statistical inference in causal partial identification with smooth densities. *arXiv preprint arXiv:2602.20681*, 2026.

Wenhan Ma, Hailin Zhang, Liang Zhao, Yifan Song, Yudong Wang, Zhifang Sui, and Fuli Luo. Stabilizing MoE reinforcement learning by aligning training and inference routers. *arXiv preprint arXiv:2510.11370*, 2025. URL https://arxiv.org/abs/2510.11370v2.

OpenAI, Sandhini Agarwal, Lama Ahmad, Sam Altman, Andy Applebaum, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Hancheol Park, Geonho Lee, Tairen Piao, and Tae-Ho Kim. Value-and-structure alignment for routing-consistent quantization of mixture-of-experts models. *arXiv preprint arXiv:2606.05688*, 2026.

Sangwoo Park, Matteo Zecchin, and Osvaldo Simeone. Adaptive prediction-powered autoeval with reliability and efficiency guarantees. *arXiv preprint arXiv:2505.18659*, 2025.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-D"unner, and Moritz Hardt. Performative prediction. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 7599–7609, 2020.

PyTorch Contributors. Numerical accuracy. https://docs.pytorch.org/docs/2.6/notes/numerical_accuracy.html, 2026. PyTorch 2.6 documentation, accessed September 7, 2026.

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*, 2017.

Ibne Farabi Shihab, Abu Sa-Adat Mohamed Moon-Im Al Ahsan, and Md Najmus Swaqeeb. Opportunity is not realizability: Selection-valid diagnostics for multi-LLM routing. *arXiv preprint arXiv:2608.08265*, 2026.

Andrii Skliar, Ties van Rozendaal, Romain Lepert, Todor Boinovski, Mart van Baalen, Markus Nagel, Paul Whatmough, and Babak Ehteshami Bejnordi. Mixture of cache-conditional experts for efficient mobile device inference. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2412.00099.

Shriniwas Ramesh Suram. Cacheable by design? training mixture-of-experts routers for locality against the edge memory-bandwidth wall: A pre-registered negative result with a systems measurement study. *arXiv preprint arXiv:2608.18261*, 2026.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 814–823, 2015.

Peng Tang, Jiacheng Liu, Xiaofeng Hou, Yifei Pu, Jing Wang, Pheng-Ann Heng, Chao Li, and Minyi Guo. HOBBIT: A mixed precision expert offloading system for fast MoE inference. *arXiv preprint arXiv:2411.01433*, 2024.

TransformerLens Contributors. TransformerLens: Activation patching and intermediate-layer execution. https://github.com/TransformerLensOrg/TransformerLens, 2026. Official implementation and documentation, accessed September 7, 2026.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *arXiv preprint arXiv:1802.04034*, 2018.

Richard A. Uhlig and Trevor N. Mudge. Trace-driven memory simulation: A survey. *ACM Computing Surveys*, 29(2):128–170, 1997. doi: 10.1145/254180.254184.

Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudík. Optimal and adaptive off-policy evaluation in contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3589–3597, 2017.

Ian Waudby-Smith and Aaditya Ramdas. Confidence sequences for sampling without replacement. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e96c7de8f6390b1e6c71556e4e0a4959-Abstract.html.

Leyang Xue, Yao Fu, Zhan Lu, Luo Mai, and Mahesh Marina. MoE-Infinity: Offloading-efficient MoE model serving. *arXiv preprint arXiv:2401.14361*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuchen Yang, Yifan Zhao, Anisha Dasgupta, and Sasa Misailovic. PagedWeight: Efficient MoE LLM serving with dynamic quality-aware weight quantization. *arXiv preprint arXiv:2607.16184*, 2026. URL https://arxiv.org/abs/2607.16184.

Youngsik Yoon, Siwei Wang, Wei Chen, and Jungseul Ok. When are experts misrouted? counterfactual routing analysis in mixture-of-experts language models. *arXiv preprint arXiv:2605.07260*, 2026.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in LLM inference. *arXiv preprint arXiv:2506.09501*, 2025.

Jiamu Zhang, Liang Wu, Mayank Darbari, and Liangjie Hong. WiSP: A working-set view of mixture-of-experts serving on extremely low-resource hardware. *arXiv preprint arXiv:2606.21868*, 2026. URL https://arxiv.org/abs/2606.21868.

Yu Zhang. Reproducible evaluation of moe expert caching: Replay semantics, workload contamination, and operating regimes. *arXiv preprint arXiv:2608.07911*, 2026. URL https://arxiv.org/abs/2608.07911.

Shien Zhu, Samuel Bohl, Robin Oester, and Gustavo Alonso. Pre-attention expert prediction and prefetching for mixture-of-experts large language models. *arXiv preprint arXiv:2511.10676*, 2025.

Tianyang Zhu. From expert reduction to behavioral divergence: Tracing numerical state through sparse moe inference. *arXiv preprint arXiv:2607.28097*, 2026. URL https://arxiv.org/abs/2607.28097v1.

Xiongwei Zhu, Xiaojian Liao, Tianyang Jiang, Yusen Zhang, Liang Wang, and Limin Xiao. ReMoE: Boosting expert reuse through router fine-tuning in memory-constrained moe llm inference. *arXiv preprint arXiv:2605.27081*, 2026. URL https://arxiv.org/abs/2605.27081.

SUPPLEMENT GUIDE TO THE CORE EVIDENCE

The main contribution is a distinction between label validity, candidate selection and decision impact. The core supplement below supplies the conditions, controls and complete comparisons supporting that distinction. It includes the ordinary-reuse baselines and the loss-budget comparisons.

Question	Evidence path
What does the theory establish?	Section A: same-checkpoint construction, exactness conditions and sampling assumptions.
Which execution was evaluated?	Section B: action definitions, ordered accounting and controls.
Why can cost disagree after routes agree?	Section C: inherited cache state and fixed-intervention capacity tests, including their limits.
When does error matter for selection?	Sections D, E and F: complete candidates, decision units and loss budgets.
Does an accepted saving survive an NLL budget?	Section G: selection rules, complete per-cohort rates and budget sensitivity for the retrospective quality endpoint.
What does acquisition cost?	Section H: first-witness timing and full ordinary suffix/KV baselines, including negative incremental results.
How can the evidence be checked?	Section J: claim-indexed CPU and archive checks, distinct from pretrained GPU replication.

Section K indexes the claims, comparison units and evidence in this supplement. Section J states which checks use archived records and which require model execution.

TERMINOLOGY AND DECISION UNITS

Term	Meaning in this paper
Behavior trace	Ordered observations from the unmodified model, tied to its collection schedule and initial state.
Execution contract	Declared target, action write sites, initial state, numerical transitions and cost accounting.
Replay / deployed value	$\widetilde{v}_a$ uses the recorded behavior path; v_a follows the action's declared execution. Values are oriented so larger is better.
Logical load / native loads	An expert fetch charged by the logical cache; the no-action deployment count supplies the native denominator. Neither is measured latency.
Source pair / view	Two requests grouped by the cohort protocol; one view targets one member. The pair is the resampling cluster, without assuming semantic equivalence.
Candidate / challenger	One declared action, including abstention where specified; a challenger is an alternative to the replay winner.
Label validity (C1)	Equality of the target replay label and its executed value. It does not establish that no challenger is better.
Selection validity (C2)	$\max_b v_b - v_{\widehat{a}} \leq \tau$ for the declared candidates. $\tau = 0$ asks optimality, including ties.
Decision impact (C3)	Consequences assessed at an explicit loss threshold and selection unit; per-request oracle regret is not aggregate-configuration regret.
NLL budget	$\Delta\mathrm{NLL} \leq \varepsilon$ on prescribed continuation tokens. Distinct from the logical-load regret budget τ and from task-quality certification.
Support / refutation / inconclusive	Sufficient evidence within the declared scope / a checked counterexample / missing evidence. A finite pass is not a population guarantee.

COHORTS, REUSE AND CONFIRMATION STATUS

Table 5 maps the principal comparisons to their source units. Counts across rows cannot be added: several analyses reuse the same requests. "Frozen" describes choices fixed before the corresponding outcomes, not an independently sampled population. Token and action details are in Appendix B; each result retains its own denominator.

Table 5: Source units and evidence reuse in the main comparisons. Views are requests; model, action, capacity and horizon repetitions are evaluation conditions, not additional independent sources.

Comparison	Source units / reuse	Tokens and endpoint	Evidence status
Execution-mode mismatch	256 SICK pairs; 512 inputs, reused across execution modes	Fixed 8-token continuation; route agreement	Paired mechanism comparison. The separate 108/197 matched-action tests count actions, not new source pairs.
Tolerance / fixed-route controls	20 TaPaCo pairs per Qwen/DeepSeek	Common artificial continuation; routes, costs, NLL	Candidate-tolerance comparison and negative controls; distinct from the PAWS window sample.
Window choice and FP32 control	20 PAWS pairs per Qwen/DeepSeek; 40 views each. Qwen FP32 reuses its sample	Artificial continuation; candidate values and regret	Frozen window test. FP32 is a mechanism rerun, not an additional source cohort.
Cross-family window audit	29 OASST pairs per family; 58 views, reused across three families	Artificial continuation; candidate values and regret	Frozen audit sample; source reuse is retained in later NLL analysis.
Cache-state confirmation	8 pairs total: 4 PAWS and 4 OASST; reused in two models	Artificial continuation; 96 action labels over 32 model–views	Follow-up fixes pairs 4–7, excluding discovery pairs 0–3; current-route and cache-state controls.
HH strong reference / budget scan	32 pairs, 64 views per Qwen/DeepSeek; disjoint family samples	Artificial continuation; costs, regret, query counts	Timing repeats reuse sources. Full matrices support retrospective threshold and denominator scans.
Prospective materiality	64 new HH roots, DeepSeek only; excludes earlier data	Artificial continuation; either-view regret above 1% of native loads	Model, actions, threshold and acquisition order fixed before outcomes; finite-frame confirmation.
Natural-response boundary	8 OASST trees shared across models and capacities; 64 source–conditions	Recorded response, nested 16/128 tokens; cost and regret	Source frame and settings fixed before execution. Later denominator scan reuses this pilot.
Single-substitution control	Exploratory: same 8 trees. Confirmation: 12 excluded trees, shared across models	Nested 16/128 tokens; occupancy, cost and selection	One substitution per candidate; held-out confirmation uses the frozen 10/12 directional rule.
Observed NLL constraint	Reuses the 20-pair PAWS and 29-pair OASST matrices above; 254 model–views	Same 9–10-token artificial continuation for selection and scoring	Retrospective feasible-set analysis; no new quality-evaluation sample.

The separate aggregate transfer analysis uses 48-source OASST and HH archives (Appendix D), not the 32-pair HH reference above. It chooses a constant on the stored OASST labels and evaluates it on the stored HH labels. This is a retrospective split comparison. The common artificial continuation is specified in Appendix B.2; its token count depends on the tokenizer. Natural-response experiments hold the recorded response fixed rather than generating new tokens.

A FORMAL IDENTIFICATION AND AUDIT CONDITIONS

Proof roles and dependency map. The specific theoretical contribution is the same-checkpoint separation: even fully realized behavior opportunity leaves action-path semantics unidentified. Its integer-load realization removes the arbitrary utility. The surrounding fiber, testing, margin and support-function arguments are standard tools. They serve four separate questions: which information is missing (Section A.2); where state divergence can begin (Section A.5.1); whether candidate errors cross a decision budget (Section A.2); and which evidence supports a witness or population claim (Section A.4). None of the conditional propagation bounds supplies an empirically validated, cheap pre-action certificate. Exact reuse conditions appear separately in Proposition 6.

A.1 TRACE-CONDITIONAL EXACTNESS

Proposition 1 (Ordered trace sufficiency)**.** *Let the cache runtime and simulator start from the same state, consume the same ordered route sequence, apply the same deterministic transition and transfer accounting, and let the tested action leave routing invariant. Then simulated and deployed cache action values are identical.*

Proof. After applying the same action, the two executions have the same state. At each expert access they therefore make the same hit/miss decision, charge the same transfer, and enter the same next state. Induction over the ordered access sequence gives identical accumulated cost. □

The order qualification is necessary for LRU/MRU. Sorting a within-token top-k set preserves membership but changes recency updates and can alter future evictions. It is also insufficient to match only continuation routes: the initial state must incorporate the same prompt-prefix accesses.

A.2 ACTION-DEPENDENT PARTIAL IDENTIFICATION

Proposition 2 (Standard fiber characterization under $\mathsf{O}_{\mathrm{beh}}$)**.** *$v_a$ is identified from behavior traces iff it is constant on every behavior fiber. For a two-action decision set $\mathcal{A} = \{a, b\}$, if two worlds θ_+, θ_- have the same behavior law but opposite strict rankings, every behavior-only rule has worst-world misselection probability at least $1/2$. If their behavior laws differ, the lower bound becomes $\{1 - \|P_{0,\theta_+} - P_{0,\theta_-}\|_{\mathrm{TV}}\}/2$.*

Fix an observation map $\mathsf{O}_{\mathrm{beh}}$ before defining a fiber. In our lower bound it exposes arbitrarily many no-action traces, the entire checkpoint, and every quantity computed along the executed behavior path, but it excludes an action-transition query and the action-specific runtime reduction schedule. For the induced law P_0^{O}, define its observational fiber $\Theta_{\mathsf{O}}(P_0) = \{\theta : P_{0,\theta}^{\mathsf{O}} = P_0\}$ and the value functional $v_a(\theta) = \int C(a,t) P_{a,\theta}(dt)$. By definition, v_a is point identified exactly when $v_a(\theta) = v_a(\theta')$ for every pair in the same fiber; equivalently the set in Eq. (2) is a singleton. This equivalence separates a property of the observation experiment from consistency of any estimator.

For the rank statement, restrict the decision set to $\{a, b\}$ and first suppose $P_{0,\theta_+} = P_{0,\theta_-}$. If action a is uniquely optimal in θ_+ and b in θ_-, a randomized behavior-only rule selects a with the same probability q in both worlds. Its error probabilities are at least $1 - q$ and q, hence their maximum is at least $1/2$. For unequal behavior laws, any action rule induces a test between the two worlds. Le Cam's testing identity gives average error at least $(1 - \|P_{0,\theta_+} - P_{0,\theta_-}\|_{\mathrm{TV}})/2$, and maximum error is no smaller than average error. The same argument applies to product laws for multiple behavior samples.

The separation from opportunity ordering is constructive rather than merely terminological, and it holds with one fixed checkpoint. Consider any binary floating-point format with p significand bits, round-to-nearest, and enough exponent range, and set $M = 2^{p+1}$. All quantities below are representable. Non-associativity gives the exact two-tree identity

$$S_+ = \mathrm{fl}(\mathrm{fl}(M + (-M)) + \tfrac{1}{2}) = \tfrac{1}{2}, \tag{4}$$

$$S_- = \mathrm{fl}(M + \mathrm{fl}((-M) + \tfrac{1}{2})) = 0, \tag{5}$$

because $\frac{1}{2}$ is strictly below half an ulp at magnitude M. Consequently $D_+ = S_+ - \frac{1}{4} = +\frac{1}{4}$ and $D_- = S_- - \frac{1}{4} = -\frac{1}{4}$. This holds, for example, with $M = 2^{25}$ in binary32 and $M = 2^9$ in bfloat16.

Orient utility so larger is better and fix arbitrary $m, g > 0$. Let $Z \in \{-1, +1\}$ be uniform and consider two successive feed-forward MoE blocks in one checkpoint, each with two experts. In the first block a fixed behavior gate selects the identity expert E_0, so the downstream router input is $H_2 = Z$. Intervention 1 overrides that dispatch with the same constant expert E_1 in both worlds. Its weights on constant input coordinates expose the fixed accumulation terms $(M, -M, \frac{1}{2})$ followed by bias $-\frac{1}{4}$. Both experts can be affine maps from $(1, 1, 1, Z)$ to a scalar, with weights $(0, 0, 0, 1)$ for E_0 and $(M, -M, \frac{1}{2}, 0)$ for E_1. The action exposes an otherwise unobserved expert computation; no tensor-shape change is required. Runtime world θ_+ uses the left tree defining S_+ above; θ_- uses the right tree defining S_-. No model parameter changes between worlds, and both use the same computation for the behavior-selected E_0.

Both use the fixed second-block top-1 router with logits $(H_2, -H_2)$ and trace $T = \mathrm{sign}(H_2)$. Under behavior, $T = Z$ in both worlds, so the entire checkpoint, behavior law, routes, and route margins are identical. Under intervention, $H_2 = D_+$ in θ_+ and D_- in θ_-, hence $T = +1$ and $T = -1$, respectively. The witness propagates across depth and does not posit a feedback edge from a token's expert output to the same layer's next-token KV. A fixed bounded two-point value function assigns

$$u(0, T) = 0, \qquad u(1, T) = (m + 2g)T + m.$$

Behavior replay prefers action 1 by margin m, while a trace-aware behavior oracle has value

$$\mathbb{E}_0 \max\{0, (m + 2g)T + m\} = m + g,$$

so its opportunity over the best fixed behavior action is exactly g. Both are identical across runtime worlds. In deployment, however, the advantages of action 1 are $2m + 2g > 0$ and $-2g < 0$. Thus checkpoint inspection, perfect behavior estimation, replay ordering, margin, and oracle gap all remain fixed while an action-induced floating-point execution choice reverses the deployment ranking. The two-point utility is bounded for each fixed (m, g); it is an external functional, not a claim of measured cache latency or NLL.

An executable two-block check in the artifact instantiates the affine experts, both top-1 gates and the external utility. Across 16 format/parameter settings (FP16, BF16, FP32 and FP64) and 144 utility settings, checkpoint hashes and the complete declared behavior records agree, action routes have positive margins, and deployment rankings reverse. This finite synthetic verification illustrates the proof. Revealing the expert reduction-tree choice resolves the ambiguity, as required by the observation-relative claim.

Fixed-action opportunity versus conditional realizability. The input contains Z, so the behavior policy $\pi_{\mathrm{beh}}(Z) = \mathbf{1}\{Z = +1\}$ is already available before execution and attains $m + g$. Consequently

$$V^0_{\mathrm{fixed}} = m, \qquad V^0_{\mathrm{Bayes}} = V^0_{\mathrm{oracle}} = m + g.$$

The quantity g is the oracle opportunity over a fixed action, not an unrealizable information gap: $V^0_{\mathrm{oracle}} - V^0_{\mathrm{Bayes}} = 0$. Nonetheless, deploying this perfect behavior policy yields regret $m + g$ in θ_+ and g in θ_-. All fixed-experiment information-ordering quantities agree across the worlds even when behavior opportunity is fully realized. An additional executable check verifies all four deterministic input policies in 144 format/utility settings and 3,600 input-randomization grid cases. Here "fixed action" means the same action for every input.

The construction also yields an exact regret statement for arbitrary input-dependent behavior-only rules. Let $q = \mathbb{E}[\pi(1 \mid Z, \mathsf{O}_{\mathrm{beh}})]$; the common contracted observation/input law makes this average identical in the two worlds. Since action 1's deployment value is constant in Z within each world, the regrets are $(1 - q)(2m + 2g)$ and $q(2g)$. Their minimax value is

$$\inf_q \max\{(1 - q)(2m + 2g), q(2g)\} = \frac{2g(m + g)}{m + 2g} > 0,$$

which proves Theorem 1 and Eq. (3). Thus more behavior samples, perfect estimation of the replay margin, or exact knowledge of the behavior-oracle gap cannot repair the missing action-to-trace link. What is required is action invariance, overlap with an identified change of measure, an identified transition model, a valid transport set, or on-policy evidence. Differentiating the bound with respect to m gives $2g^2/(m + 2g)^2 > 0$; its limits at $m \downarrow 0$ and $m \to \infty$ are g and $2g$, respectively.

Table 6: Ordinary CPU linear kernels realize the integer-load separation. Checkpoint and every no-action call/trace agree. Only the action-exposed expert's internal batching rule differs; revealing that rule distinguishes the worlds. Costs are mean integer expert loads under the two-input law, after a common paid three-load warmup, not serving latency.

Action-path runtime	No action	Action	Preferred fixed action
One-row linear	1.8	2.4	No action
Padded action-expert linear	1.8	0.8	Action

Corollary 1 (Non-tie rounding-cell robustness)**.** *Let $U_{\to} = \mathrm{nextafter}(-M, +\infty) + M$ be the representable spacing toward zero. Replace $\frac{1}{2}$ and the subtractive bias $\frac{1}{4}$ in the construction by any representable e, b satisfying $0 < b < e < U_{\to}/2$, with normal, non-underflowing subtractions. The left and right reduction trees followed by bias $-b$ give $D_+ = \mathrm{fl}(e-b) > 0$ and $D_- = -b < 0$. Hence the behavior experiment and both opportunity quantities remain fixed while deployment rankings reverse with the same regret bound. Moreover, every such pair lies inside real-valued neighborhoods that round to the same (e, b), so the reversal does not require an exact router tie.*

Proof. Because $e < U_{\to}/2$, round-to-nearest gives $\mathrm{fl}(-M + e) = -M$, whereas $\mathrm{fl}(M - M) = 0$. Therefore $\mathrm{fl}(\mathrm{fl}(M - M) + e) = e$ and $\mathrm{fl}(M + \mathrm{fl}(-M + e)) = 0$. Subtracting b, with the assumed absence of underflow, gives the two strict signs. The remainder of the construction depends only on those signs, so its replay margin, oracle gap, deployment advantages, and minimax regret are unchanged. Rounding is locally constant away from cell boundaries; choosing interior preimages of representable e and b gives the claimed neighborhoods. □

The witness is robust to partial runtime metadata. Let Q_+ and Q_- be the laws of the *complete joint decision observation* $(Z, \mathsf{O}_{\mathrm{beh}}, S)$, including any added runtime information S, and put $\rho = \|Q_+ - Q_-\|_{\mathrm{TV}}$. A maximal coupling contains a common sub-probability measure of mass $1 - \rho$. If q is the conditional probability of choosing action 1 on that common part, the two regrets are at least $(1 - \rho)(1 - q)(2m + 2g)$ and $(1 - \rho)q(2g)$. Balancing them gives

$$\inf_{\pi} \sup_{\theta \in \{\theta_+, \theta_-\}} \mathrm{Regret}_{\theta}(\pi) \geq (1 - \rho)\frac{2g(m + g)}{m + 2g}. \tag{6}$$

The factor is sharp for a side channel that reveals the runtime world with probability ρ and otherwise emits one common symbol. This interpolation separates full execution disclosure ($\rho = 1$) from any side channel that leaves an observationally common component; it also gives an operational target for how informative an audit or kernel signature must be. Here the two world-specific action gaps are constant across observed inputs. The TV of S alone cannot replace joint TV: with uniform observed bit Z, let $S = Z$ in one world and $S = 1 - Z$ in the other. Both metadata marginals are uniform (TV zero), but (Z, S) reveals the world (joint TV one), allowing zero regret. Marginal TV is sufficient if the remaining observed variables have the same conditional law given S in both worlds, as when S is independent of all behavior/input observations within each world. The proof uses no such independence when TV is defined jointly. An exact finite-channel check compares primal randomized decisions and dual Bayes risks in 675 cases, validating the calculation.

The construction is a realizability statement for the contracted behavior-only statistical experiment, not an assertion that the runtime schedule can never be inspected or executed. Granting every checkpoint weight does not distinguish these worlds. If a method learns the action-specific reduction tree, simulates its rounded transition exactly, or executes the intervention, that information belongs in an enlarged map O'. Its fiber obeys $\Theta_{\mathsf{O}'}(P_0) \subseteq \Theta_{\mathsf{O}}(P_0)$, so its identified set and minimax regret cannot increase. In the witness, observing either the action kernel or its router-sign output separates the worlds and collapses the fiber. The theorem therefore lower-bounds rules trained from behavior-path evidence without an action-transition/runtime query; the on-policy audit below buys precisely that query.

Stock-linear CPU realization (developmental). The reduction-order construction also has an executable realization using ordinary `torch.nn.functional.linear`, without hand-parenthesized scalar adds. Shape-dependent numerical differences are already documented by PyTorch (PyTorch Contributors, 2026). We first tested 192 single-output configurations, with no

strict sign reversal, then explicitly extended development to 8/32/64 output channels. The latter grid has 576 enumerated configurations (528 unique), including 120 unique strict reversals. These are searched synthetic configurations, not independent observations or a failure-rate estimate. All outcomes, including the negative parent screen, remain in the artifact.

The first selected case has eight FP32 output channels, each with weight row $(2^{25}, -2^{25}, 1/8, 0)$ and bias $-1/16$. On input $(1, 1, 1, 1)$, ordinary one-row linear gives $+1/16$ in every channel; evaluating the same row in a zero-padded 32-row batch gives $-1/16$. Both use stock linear operators and identical parameter tensors. This does not identify the backend's exact internal reduction tree. The recorded environment is PyTorch 2.6/CPU with MKL; the local CUDA wheel cannot execute on the installed GPU, so no GPU result is claimed.

For the three-block model, the selected expert additionally reads the final input coordinate (weight 1); another expert exposes the observable input sign. The runtime rules differ *only when an action sends the positive input to the cancellation-sensitive expert*. Every no-action operation, including calls to both experts and a common warmup, uses the identical one-row path. Thus the complete recorded no-action tensors, bits, call shapes and checkpoint agree, not just expert IDs. Both experts are seen under behavior. At the acted positive input the final coordinate is zero, retaining strict router signs $\pm 1/16$. The two downstream top-1 blocks amplify these signs into different expert paths.

With positive-input probability $4/5$, the common warmed cache is $(0, 1, 1)$. Rows for the positive/negative inputs and columns for no action/action give replay and one-row costs $((2, 3), (1, 0))$, but padded-action costs $((2, 1), (1, 0))$. Hence fixed-action ordering reverses as in Table 6. The behavior oracle gap is $1/5$, attained by an input-dependent behavior policy in both worlds; the worst-world deployment regret remains at least $2/5$, by the earlier integer-load argument. Exact cache execution and 441 policy-grid checks verify these numbers. Five tests check complete no-action equality, the cost matrices and positive margins, paid warmup, and controls with a common runtime, one output channel, or a shared bias moving both routes away from the boundary. This is a selected, ill-conditioned synthetic realization of the stated observation contract. Exposing the action-path batching rule resolves the ambiguity.

The distinction can also be stated through statistical experiments. For $\mathcal{E}_0 = \{P_{0,\theta} : \theta \in \Theta\}$ and $\mathcal{E}_a = \{P_{a,\theta} : \theta \in \Theta\}$, define the directed deficiency

$$\delta(\mathcal{E}_0, \mathcal{E}_a) = \inf_K \sup_\theta \|K P_{0,\theta} - P_{a,\theta}\|_{\mathrm{TV}},$$

where K ranges over Markov kernels. Zero deficiency means a deployment observation can be simulated from behavior without knowing θ; the randomization criterion then transfers every bounded deployment decision. Positive deficiency guarantees a bounded decision problem with a nonzero risk gap. Our fiber rank reversal gives an explicit MoE decision witnessing that gap rather than attempting to estimate deficiency from one finite dataset.

Budgeted selection. If a nonempty bounded simultaneous compatibility set C contains every transport error allowed by the observation contract, then $\widetilde{v} + C$ outer-bounds the identified value set, and for regret budget $\tau \geq 0$, Equation (1) gives the standard check

$$B_C := \max_b \{h_C(e_b - e_{\widehat{a}}) - g_b\} \leq \tau, \tag{7}$$

with $h_C(u) = \sup_{\delta \in C} u^\top \delta$ and e_b a coordinate vector: necessary and sufficient over C, possibly conservative for the actual fiber, with $\tau = 0$ recovering optimality and only paired errors entering.

Let $\Delta_{\mathsf{O}} = \{v(\theta) - \widetilde{v} : \theta \in \Theta_{\mathsf{O}}(P_0)\}$ be the jointly realizable transport errors. If structural assumptions yield $\Delta_{\mathsf{O}} \subseteq C$, then $\widetilde{v} + C$ is an outer bound, not necessarily the sharp identified set $\widetilde{v} + \Delta_{\mathsf{O}}$. Equality requires attainability of every element of C by an admissible world. For candidate a, robust optimality is

$$\inf_{\delta \in C} \min_b \{\widetilde{v}_a + \delta_a - \widetilde{v}_b - \delta_b\} \geq 0.$$

Moving the infimum inside the finite minimum and using the support function reduces this condition exactly to the zero-budget case of Eq. (7). Thus the paired support certificate is both necessary and sufficient relative to the declared compatibility set, but need not be necessary for optimality on the actual fiber. For example, with $\widetilde{v} = (1, 0)$, $\Delta_{\mathsf{O}} = \{(0, 0)\}$ and $C = [-1, 1]^2$, action 1 is identified as optimal, although its margin 1 is smaller than $h_C(e_2 - e_1) = 2$. A transport or smoothness

assumption can narrow C; behavior data alone cannot validate that assumption. A set covering only the realized error gives a conditional certificate for that world, not an identified set over all worlds. Random confidence sets require their separate coverage guarantee.

More generally, fix the candidate set, decision unit and replay choice $\widehat{a}$, and let C be nonempty and bounded. For a declared regret budget $\tau \geq 0$, the same finite maximum gives

$$\sup_{\delta \in C} \mathrm{Regret}_{\widetilde{v}+\delta}(\widehat{a}) = \sup_{\delta \in C} \max_b \{\delta_b - \delta_{\widehat{a}} - g_b\} = \max_b \{h_C(e_b - e_{\widehat{a}}) - g_b\} = B_C.$$

The self comparison is zero, so $B_C \geq 0$. Hence $B_C \leq \tau$ is equivalent to accepting that budget for every vector in C; for the actual fiber it is sufficient but can be conservative. A sharp worst-regret bound on that fiber recovers equivalence. The preceding optimality derivation is its $\tau = 0$ case. Additional cross-action restrictions, including a justified conditional transport model (Lin et al., 2026), can narrow C but are not supplied by an observed route-mismatch rate.

Two elementary checks explain the distinction. With replay values $(1, 0)$, arbitrarily large common errors (K, K) preserve the winner and give $B_C = 0$ for any bounded set of such shifts. Conversely, $C = \{(0, 0), (0, 2)\}$ fixes the winner's error at zero but gives $B_C = 1$: its exact value cannot certify optimality, although a budget of one passes. If the realized error is $(0, 0)$, the actual choice is optimal despite the larger outer-set bound. Thus a failed robust check is not itself a witnessed deployment failure.

For a random set $C(D)$ covering the realized vector jointly with probability at least $1 - \alpha$, pointwise validity instead implies

$$\Pr\{B_{C(D)} \leq \tau,\ \mathrm{Regret}(\widehat{a}) > \tau\} \leq \Pr\{\delta \notin C(D)\} \leq \alpha.$$

The coverage must hold for the same candidates and selection procedure; separate marginal intervals do not imply it. This bounds a false-acceptance event, not conditional risk given acceptance. Adaptive stopping or choosing among decision units requires corresponding simultaneous/time-uniform coverage. Our complete-matrix checks do not estimate such a cheap confidence set, and ex-post measured state displacements do not establish one.

A.3 A REALIZATION WITH INTEGER EXPERT-LOAD COST

The arbitrary-(m, g) construction above uses an external two-point utility. The following separate realization uses actual integer cache misses and one common no-action warmup from empty caches. It does not assert arbitrary independent margins or an autoregressive KV-cache serving implementation.

Proposition 3 (Logical-load realization)**.** *For $1/2 < p < 1$, there exist two same-checkpoint three-block top-1 MoE worlds with identical contracted behavior observations and the same reached capacity-one per-layer caches, for which behavior replay has strict fixed-action margin $2p - 1$ and oracle advantage $1 - p$, with zero oracle-to-Bayes gap. Deployment fixed-action rankings are opposite, and the minimax regret over behavior-only input-dependent policies, relative to each world's conditional Bayes policy, is exactly $p/2$ integer expert loads per request. Both first-layer experts occur in the behavior experiment.*

Proof. The first gate has scores $(2x_1 - 1, 1 - 2x_1)$. Expert weights are $(1, 0, 0, 1)$ and $(M, -M, e, 1)$, with biases $0, -b$, respectively; use the representable $0 < b < e < U_{\rightarrow}/2$ conditions of Corollary 1. Both subsequent blocks have gate $(h, -h)$ and constant expert outputs $+1/-1$. Starting with empty caches, the no-action warmup $w = (1, 1, 1, -2)$ gives route $(0, 1, 1)$ and costs three loads in either world. Measured inputs are $x_+ = (1, 1, 1, 0)$ and $x_- = (0, 0, 0, -1)$, with probabilities $p, 1 - p$. Behavior executes expert 0 on x_+ and expert 1 on x_-; action 1 swaps the first selected expert. All behavior and warmup records agree across worlds. On action 1's x_+ input, expert 1's left/right reductions have opposite signs; on its observed behavior input x_-, both reductions instead agree. Suffix blocks propagate sign, with no router ties. Charging one load per absent dispatched expert and holding behavior suffix routes fixed gives replay costs below; actual dispatch gives deployment costs. Rows are x_+, x_-, columns are actions $0, 1$, and the common warmup cost is excluded:

$$\widetilde{C} = C_L = \begin{pmatrix} 2 & 3 \\ 1 & 0 \end{pmatrix}, \qquad C_R = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}.$$

Replay fixed-action costs are $(1+p, 3p)$, whereas both the behavior oracle and the available input policy $\pi_0(x_+) = 0, \pi_0(x_-) = 1$ attain cost $2p$. Left deployment equals replay; right fixed-action costs are $(1+p, p)$, proving strict reversal for $p > 1/2$.

For any randomized behavior-only policy, let q_+, q_- be its probabilities of action 1 conditional on x_+, x_-, averaged over common-law behavior history. Relative to the conditional deployment optima, the regrets are

$$R_L = pq_+ + (1-p)(1-q_-), \qquad R_R = p(1-q_+) + (1-p)(1-q_-).$$

Their maximum is at least their average, $p/2 + (1-p)(1-q_-) \geq p/2$. Taking $q_- = 1, q_+ = 1/2$ attains this bound in both worlds. The fully realizable behavior policy has regret 0 on the left and p on the right. This is a continuous-policy algebraic argument, not a finite-grid inference. □

The executable artifact verifies the cost matrices using capacity-one LRU objects across 16 floating-point settings, 64 setting/probability combinations, and 1,600 policy-grid checks. Every decision cost is in $\{0, 1, 2, 3\}$; adding the actual common three-load warmup leaves all comparisons unchanged. This is an executed warmup in the synthetic sequence model, not a demonstrated autoregressive serving prefix. The earlier stipulated-cache variant remains in the artifact, separately labeled. The action swaps a first-layer expert, not native W8A16 precision. Expert-level behavior coverage does not identify unobserved expert-input or action-transition values.

Within this specified noiseless two-world family, a single action 1 query on x_+ observes cost 3 or 1 and identifies the correct conditional policy; queries on x_- do not distinguish the worlds. This construction-specific statement is not a general one-query certification or noisy-risk guarantee. Disclosing the reduction tree also resolves the ambiguity. The strengthened witness replaces the custom utility with cache misses and obtains the common initial cache through an executed warmup.

Task-relevant runtime evidence. For one decision in this load witness, let Q_L^+, Q_R^+ denote the laws of all extra information conditional on current input x_+. At x_- both worlds prefer action 1, so the exact minimax regret is

$$p \inf_{q:\mathcal{S}\to[0,1]} \max\{\mathbb{E}_{Q_L^+} q(S), \mathbb{E}_{Q_R^+}[1-q(S)]\}.$$

This is the classical minimax testing risk restricted to the input where optimal actions differ. If the channel reveals the world with probability r on x_+ and otherwise emits a common symbol, the value is exactly $p(1-r)/2$: follow the disclosed world, randomize equally on erasure, and always choose action 1 on x_-. The common erased part proves the matching lower bound. This calculation specifies which evidence resolves the ambiguity in this construction.

A.4 Anytime-valid witness auditing

The following standard conditional-null formulation is distinct from the prospective finite-frame test used in the main text. Let $X_t = 1$ for a positive-regret source pair and assume $H_0 : \mathbb{E}[X_t \mid \mathcal{F}_{t-1}] \leq \gamma$. For fixed probability weights,

$$E_t = \sum_{q>\gamma} w_q \prod_{i=1}^{t} \left(\frac{q}{\gamma}\right)^{X_i} \left(\frac{1-q}{1-\gamma}\right)^{1-X_i}. \tag{8}$$

Theorem 2 (Standard optional stopping and audit complexity)**.** *Under this conditional null, (E_t) is a nonnegative supermartingale and $\Pr(\sup_t E_t \geq 1/\alpha) \leq \alpha$. For i.i.d. failure probability $\gamma < p < 1$, every test with type-I error α and power $1-\beta$, where $0 < \alpha < 1-\beta \leq 1$, obeys*

$$\mathbb{E}_p \tau \geq \frac{\mathrm{kl}(1-\beta, \alpha)}{\mathrm{kl}(p, \gamma)}. \tag{9}$$

For fixed $\gamma < p < 1$, uniform mixing on $(\gamma, 1)$ gives power one and $\mathbb{E}_p \tau_c \sim \log(1/\alpha)/\mathrm{kl}(p, \gamma)$ as $\alpha \downarrow 0$.

These results use established sequential inference tools (Cook et al., 2024; Wang et al., 2017). A random-order finite frame need not obey the conditional null; its hypergeometric construction appears below.

For fixed $0 < \gamma < q < 1$, let

$$E_t(q) = \prod_{i=1}^{t} \left(\frac{q}{\gamma}\right)^{X_i} \left(\frac{1-q}{1-\gamma}\right)^{1-X_i}.$$

If $p_i = \Pr(X_i = 1 \mid \mathcal{F}_{i-1}) \le \gamma$, then

$$\mathbb{E}\left[\frac{E_i(q)}{E_{i-1}(q)}\middle|\mathcal{F}_{i-1}\right] = p_i\frac{q}{\gamma} + (1-p_i)\frac{1-q}{1-\gamma} \le 1,$$

because the right side is increasing in p_i and equals one at $p_i = \gamma$. Nonnegative mixtures preserve the supermartingale property, proving Eq. (8). Ville's inequality gives $\Pr_{H_0}(\sup_t E_t \ge 1/\alpha) \le \alpha$.

For the lower bound, compare i.i.d. Bernoulli laws with parameters p and γ. Under the usual finite-expectation condition, Wald's identity for the stopped log likelihood ratio gives

$$D_{\mathrm{KL}}(P_p^\tau \| P_\gamma^\tau) = \mathbb{E}_p[\tau]\,\mathrm{kl}(p,\gamma).$$

Data processing through the terminal reject/accept decision lower-bounds the left side by $\mathrm{kl}(1-\beta,\alpha)$ when $\alpha < 1-\beta$, proving Eq. (9). If $\mathbb{E}_p\tau = \infty$, the lower bound is immediate. If the mixture assigns mass $w_p > 0$ to p, the correct pathwise relation is

$$\log E_t \ge \log w_p + \log E_t(p),$$

not an equality with that component. Hence mixture rejection occurs no later than when $E_t(p)$ crosses $1/(\alpha w_p)$. For fixed $0 < \gamma < p < 1$, bounded log increments and positive drift give

$$\mathbb{E}_p\tau_{\mathrm{mix}} \le \frac{\log(1/\alpha) + \log(1/w_p) + O(1)}{\mathrm{kl}(p,\gamma)}.$$

This crossing test has power one under p; taking $\beta = 0$ in the lower bound matches its leading coefficient as $\alpha \downarrow 0$. For fixed nonzero β, the general lower bound retains its stated $\mathrm{kl}(1-\beta,\alpha)$ numerator rather than claiming the same constant. A finite grid that omits p need not attain this rate. This lower bound concerns between-pair stopping. The within-pair first-witness rule saves action executions pointwise and does not alter X_i.

Continuous-mixture completion (standard). An atom at an unknown p is unnecessary if a fixed continuous alternative is used. With $S_n = \sum_{i=1}^n X_i$, take the standard conjugate mixture (Howard et al., 2021)

$$\overline{E}_n = \frac{1}{1-\gamma}\int_\gamma^1 (q/\gamma)^{S_n}\{(1-q)/(1-\gamma)\}^{n-S_n}\,dq.$$

Tonelli's theorem and the same one-step inequality preserve the conditional-null supermartingale and Ville guarantees. For fixed $\gamma < p < 1$, choose $\delta > 0$ with $[p, p+\delta] \subset (\gamma, 1)$. The log-likelihood derivative per observation is bounded on this interval, so integrating just over $[p, p+\delta/n]$ gives, for every data path,

$$\log \overline{E}_n \ge \log E_n(p) - \log n - C_{p,\gamma}.$$

The strong law and the maximum-likelihood upper bound imply $n^{-1}\log\overline{E}_n \to \mathrm{kl}(p,\gamma) > 0$ almost surely. For $\tau_{\mathrm{c}} = \inf\{n : \overline{E}_n \ge 1/\alpha\}$, set $h = \log(1/\alpha)$ and $d = \mathrm{kl}(p,\gamma)$. For any fixed $\epsilon > 0$ and all sufficiently large h, $n \ge (1+\epsilon)h/d$ makes $\{\tau_{\mathrm{c}} > n\} \subseteq \{\log E_n(p) < h + \log n + C_{p,\gamma}\}$ a fixed-fraction lower deviation of bounded i.i.d. log-likelihood increments. Hoeffding's bound is exponentially summable in n, hence $\limsup_{\alpha\downarrow 0} \mathbb{E}_p\tau_{\mathrm{c}}/h \le (1+\epsilon)/d$. Letting $\epsilon \downarrow 0$ and applying the power-one lower bound proves the stated leading rate for every fixed interior p, without knowing it. This is pointwise asymptotic efficiency, not a uniform finite-budget guarantee.

The frozen experiments retain their original finite grid. For such a grid, the asymptotic log-growth is $\max_q[p\log(q/\gamma) + (1-p)\log\{(1-q)/(1-\gamma)\}]$, which can be negative throughout a near-null interval despite valid type-I control. The artifact includes an exact-rational continuous-mixture implementation and a retrospective support diagnostic; it does not replace any frozen endpoint.

Empirical sampling contract and retrospective correction. The validation audit samples 29 distinct pairs without replacement from a 120-pair frame, then sorts the selected parent indices before execution. We verified the parent/sample hashes, seeded selection, text membership, all 29 outcomes per family, and strict-witness arithmetic. A uniform subset does not make its sorted prefixes i.i.d. or establish the conditional null of Theorem 2. Even random-order sampling without replacement needs its own finite-population treatment (Waudby-Smith & Ramdas, 2020). Thus the reported first crossings and permutations are descriptive; all 29 pairs were actually executed. This qualification supersedes the earlier interpretation of those prefixes as population invalidity certificates.

The complete data still provide a sampling-assumption-free conclusion on the declared frame: 16/15/13 distinct witnessed pairs imply respective frame-risk lower bounds $16/120 = 13.33\%$, $15/120 = 12.50\%$, and $13/120 = 10.83\%$. Each exceeds $\gamma = .10$, whatever the unobserved labels or sampling order. These are deterministic frame bounds conditional on fixed audited runtime/action labels, not lower bounds for all OASST or future executions. Under the additional declared simple-random-subset design, exact fixed-horizon hypergeometric inversion gives one-sided 95% lower totals 49/45/37 out of 120. Specifically, for observed s positives, retain totals K with $\Pr\{\mathrm{Hypergeom}(120, K, 29) \geq s\} > .05$ and take the smallest retained K. The legacy binomial endpoints are retained as historical, assumption-dependent numbers requiring i.i.d. Bernoulli source units at a fixed horizon, not relabeled as exact finite-population bounds. This qualification applies wherever those historical endpoints are reproduced; a numerical endpoint alone is not a certificate under the actual sorted-subset design.

An explicit hypothetical counterexample illustrates why this matters. Put the 12 positives first in a 120-element frame and sort a uniform 29-subset. The frozen grid crosses after two initial positives; the probability of sampling at least two is 0.83990, despite frame risk exactly .10. This is not an estimate of the actual experiment's false-positive rate. Exact small-population enumeration checks the corrected interval's coverage. Later SHA-ordered, deduplicated OASST and HH cohorts likewise do not by themselves establish i.i.d. source streams: their paired-bootstrap inference remains approximate. Pointwise witness soundness, complete-batch timings and matched cost attribution do not require the disputed between-pair stopping interpretation. No frozen data, gates or endpoints are rerun or replaced; the artifact includes the correction and its limitations.

A.5 Action-dependent trace transport

A.5.1 Proof and boundary-mass form of the cascade certificate

Theorem 3 (Conditional action-induced cascade envelope)**.** *Index router events in causal execution order by q. Let s_q include all live hidden states, KV entries, and action state needed by the transition, and choose a norm with $r_q \leq D_q := \|s_q^a - s_q^0\|$. Suppose valid bounds L_q, B_q, J_q respectively control same-proposal state propagation, direct action injection (including execution errors), and a change of proposal. Then*

$$D_{q+1} \leq L_q D_q + B_q + J_q \mathbf{1}\{g_q \leq 2(M_q D_q + \epsilon_q)\}. \tag{10}$$

Starting from a valid $R_{q_0} \geq D_{q_0}$, if $R_{q+1} = L_q R_q + B_q$ and $g_q > 2(M_q R_q + \epsilon_q)$ throughout a suffix, its global proposal sets are unchanged. Exact accounting additionally requires equal actual ordered dispatch and matched accounting state/transition.

For any selected behavior index i and excluded index j, actual score perturbation bounded by d_ℓ leaves separation at least $\hat{z}_{\ell,i}^0 - \hat{z}_{\ell,j}^0 - 2d_\ell$. Thus $g_\ell > 2d_\ell$ preserves the global top-k set. For actual floating-point scores, Cauchy–Schwarz instead gives $d_\ell \leq M_\ell r_\ell + \|\eta_\ell^a - \eta_\ell^0\|_\infty$. No assumption of zero numerical error is valid merely because the checkpoint is unchanged.

Use full states s_q in causal event order; a single current-token hidden vector does not specify historical KV or action residency state. For transitions $F_q^a(s, S)$, add and subtract $F_q^a(s_q^0, S_q^a)$ and $F_q^a(s_q^0, S_q^0)$. Under the stated valid bounds, the resulting terms are at most $L_q D_q$, J_q if proposals differ, and B_q. Since $r_q \leq D_q$, proposal disagreement implies $g_q \leq 2(M_q D_q + \epsilon_q)$, proving Eq. (10). The injection bound includes any numerical transition residual not controlled by a continuous-state Lipschitz bound. Such bounds are assumptions, not quantities estimated by the empirical margin check. Induction with $R_{q+1} = L_q R_q + B_q$ gives the conditional suffix result.

For deterministic $R_q \geq D_q$ and valid numerical bounds, define $Q_q(u) = \Pr\{g_q \leq 2(M_q u + \epsilon_q)\}$. Then

$$\Pr(S_q^a \neq S_q^0) \leq Q_q(R_q), \qquad \mathbb{E}D_{q+1} \leq L_q \mathbb{E}D_q + B_q + J_q Q_q(R_q).$$

With a fixed common input-token sequence, equal initial prefix state, and deterministic common pre-action computation, induction over tokens proves first-acted-depth proposal equality: its KV entries precede that action. Free generation can instead feed a different token back into all layers. Proposal-set equality also does not imply ordered dispatch equality. For example, logits $(100, 99, 0)$ and $(99, 100, 0)$ satisfy $g = 99 > 2d = 2$ but swap the top-two order. From an empty capacity-two LRU, access sequences ABCA and BACA have four and three misses respectively. Exact value transport therefore also requires matching the action's actual ordered dispatch and the initial accounting state and transition, or an explicitly order-invariant objective.

Native-route analyses in Table 11 are restricted to positions where each architecture's native selection equals global top-k of the recorded linear scores (92.4–100% eligibility), while the exact theorem is checked on global top-k itself.

Proposition 4 (One-sided witness audit)**.** *Let D_j denote complete-sweep decision failure and let W_j denote that a prespecified partial search executed a challenger strictly better than the transported choice. Then $W_j \leq D_j$ pointwise, and every valid lower confidence bound for $\Pr(W_j = 1)$ is a valid lower bound for $\Pr(D_j = 1)$. If a no-witness pair exhausts every challenger, then $W_j = D_j$ while allowing within-pair stopping at the first witness.*

Proof. A strictly better executed challenger is itself a full-sweep failure witness, so $W_j = 1$ implies $D_j = 1$. Probability monotonicity transfers any lower bound. Under exhaustion, $D_j = 1$ also implies that the search eventually executes a better challenger, proving equality. The argument permits stopping inside a deployment unit but does not justify optional stopping over the number of sampled units. □

Fixed-budget label-validity audit. Under the i.i.d. Bernoulli model, the 10% target at 95% confidence requires 29 independent source pairs with zero failures, or 174 on-policy executions for three non-null actions and two views per pair. The actual design samples without replacement from a finite frame; it does not establish that i.i.d. premise (Appendix A.4). We sampled 29 OASST1 pairs before the Qwen and DeepSeek outcomes and reused exactly this frozen paired sample for the prospectively specified GPT-OSS replication. A pair fails when either view has strictly positive deployment regret under the replay-selected action; a different action with tied deployment value is recorded as choice disagreement but not failure. Qwen, DeepSeek, and GPT-OSS respectively yield 20, 18, and 11 failing pairs. The retained one-sided binomial lower endpoints exceed 10% only as diagnostics conditional on the extra i.i.d. premise, not as sampling-free population certificates. All three also have at least one numerically inexact candidate label on every pair. The GPT-OSS adapter passed the native-output gate before the full run; its local PyTorch runtime dequantizes the checkpoint's MXFP4 expert bank.

The common sample permits paired descriptive comparisons but not pooling into one architecture-independent rate. Seven pairs fail in every family, 25 in at least one, and four in none. Qwen versus GPT-OSS has ten Qwen-only and one GPT-OSS-only failures (legacy two-sided exact McNemar $p = .0117$, conditional on independent exchangeable discordant pairs, not guaranteed by frame selection alone). GPT-OSS nevertheless has nearly the same early-window route-set mismatch as DeepSeek (9.69% versus 9.87%) while its pair failure rate is lower (37.9% versus 62.1%). Hence aggregate trajectory mismatch is not a calibrated surrogate for selection risk; on-policy decision auditing remains necessary.

The exact boundary audit decomposes each stored value error as $\delta_a = v_a - \widetilde{v}_a$ and verifies Eq. (1) on 756 views from PAWS, ParaSCI, and OASST1: the maximum identity error and the number of failure-sign disagreements are both zero. On the common OASST sample, Qwen minus GPT-OSS mean replay margin is $+2.586$ (paired bootstrap interval $[0.207, 4.931]$), but centered transport radius is $+6.810$ $[4.987, 8.728]$ and pairwise span is $+10.483$ $[7.828, 13.086]$. DeepSeek minus GPT-OSS centered radius is only -0.125 $[-1.319, 1.056]$, while replay margin is -1.224 $[-2.414, -0.017]$. This separates the Qwen high-error regime from the DeepSeek small-margin regime. Descriptive centered-radius failure AUC is 0.704/0.845/0.799 for Qwen/DeepSeek/GPT-OSS, compared with 0.655/0.746/0.697 for negative replay margin; these are mechanism checks on stored audits, not frozen predictors.

Table 7: Historical pair-level diagnostics of transported action labels. Exact error means that any candidate action has unequal replayed and deployed value; decision failure means that either paired view has positive deployed regret. Bounds retain one-sided 95% binomial endpoints, exact only under i.i.d. Bernoulli source units at a fixed horizon; the frozen cohorts do not establish this premise (Appendix A.4). The historical verdicts use the frozen 10% target conditional on that premise, not a sampling-free population certificate.

Model	Source/action class	Pairs	Exact error	Decision fail	95% interval	Verdict
Qwen	PAWS windows	80	80/80	37/80	[36.7%,56.0%]	invalid
DeepSeek	PAWS windows	80	80/80	35/80	[34.3%,53.6%]	invalid
Qwen	ParaSCI windows	80	80/80	40/80	[40.3%,59.7%]	invalid
DeepSeek	ParaSCI windows	80	80/80	31/80	[29.6%,48.5%]	invalid
Qwen	TaPaCo tolerances	20	20/20	0/20	[0.0%,13.9%]	uncertain
DeepSeek	TaPaCo tolerances	20	20/20	0/20	[0.0%,13.9%]	uncertain
Qwen	OASST natural windows	29	29/29	20/29	[52.1%,82.8%]	invalid
DeepSeek	OASST natural windows	29	29/29	18/29	[45.1%,77.1%]	invalid
GPT-OSS	OASST natural windows	29	29/29	11/29	[22.9%,54.9%]	invalid

Table 8: Two frozen 29-pair OASST audits. Earlier fail is complete-sweep failure on the first sample; other columns use a disjoint validation sample. Frame lower counts verified witnesses among all 120 frame pairs, without a sampling assumption, conditional on fixed audited runtime/action labels. Saved counts action executions, not wall time. All 29 pairs were executed.

Family	Earlier fail	Witness	Frame lower	Actions saved
Qwen3-MoE	20/29	16/29	16/120	21.84%
DeepSeek-V2-Lite	18/29	15/29	15/120	32.76%
GPT-OSS-20B	11/29	13/29	13/120	21.26%

Frozen witness confirmation. The within-pair order (smaller replay margin view first, then challengers by transported value) was selected on PAWS matrices. We then sampled 29 unused OASST1-validation pairs and completed the fixed budget in every family. Exact witness counts are 16 Qwen, 15 DeepSeek, and 13 GPT-OSS. Among the fixed 120-pair parent frame these imply lower rates of $16/120$, $15/120$ and $13/120$, each above .10 without a sampling assumption, conditional on fixed runtime/action labels. The legacy one-sided 95% binomial lower endpoints 38.44%, 35.20%, and 28.93% require the additional i.i.d. premise; they are not the basis for this finite-frame conclusion or an early-stop claim. Stopping a pair at its first witness uses 136/117/137 non-null actions instead of 174, saving 21.84%/32.76%/21.26%. No-witness pairs are exhaustive, so these indicators equal complete-sweep failures rather than relying on a learned search-recall estimate.

Second logical action class. Before action outcomes, we froze 20 additional OASST1-validation pairs that exclude every cache-audit source pair. The mandatory action chooses one of three cross-depth router masks, balanced to within one layer, for fake-W4A4 execution (Appendix B.3). Transported value counts direct recorded router-slot changes on BF16 behavior states; deployment value counts such changes over the complete incremental action-induced execution. All 120 view-level value vectors are numerically inexact, and the exact regret identity has zero error. Pair failures are 5/20 Qwen, 11/20 DeepSeek, and 15/20 GPT-OSS; their one-sided lower endpoints are 10.41%, 34.69%, and 54.44%. These legacy binomial endpoints exceed the frozen 10% threshold under the extra i.i.d. premise; the without-replacement sample alone does not justify that population inference. Qwen also supplies the useful mechanism contrast: its mean absolute error radius is largest (211.55), but mean replay margin is 31.63 versus 9.15/4.50 in DeepSeek/GPT-OSS. Common action shift cancels, leaving Qwen with fewer boundary crossings. The fake quantize–dequantize path makes no packed-kernel, memory, latency, or task-quality claim.

Proposition 5 (Causal onset)**.** *Let layer ℓ be the earliest MoE block whose expert output is modified. If its router and residency rule receive the same pre-expert hidden and cache state in replay and deployment, their selected set at layer ℓ is equal. Feedback divergence can begin only after the modified expert output enters a downstream hidden state.*

This follows directly from equality of the router inputs and deterministic selection rule. It supplies an implementation falsifier: disagreement at the earliest acted-on layer indicates alignment, precision, or cache-order error, not hidden-state feedback.

Table 9: Descriptive A800 cost calibration for the on-policy audit. Each family has 12 timed actions; projections multiply the frozen action counts by the measured median and exclude model loading. GPU-minutes are in parentheses.

Family	Action s [IQR]	Full s	Witness s	Saved s (GPU-min)
Qwen3-MoE	1.702 [1.357,2.206]	296.2	231.5	64.7 (1.08)
DeepSeek-V2-Lite	1.115 [0.887,1.448]	194.1	130.5	63.6 (1.06)
GPT-OSS-20B	0.814 [0.644,1.046]	141.6	111.5	30.1 (0.50)

Table 10: Frozen fake-W4A4 router precision-mask audit on 20 cache-audit-disjoint OASST1-validation pairs. Gap and centered error are route-slot units; lower retains the legacy one-sided 95% binomial endpoint, requiring i.i.d. pairs rather than following automatically from finite-frame sampling.

Family	Pair fail	Lower	Gap	Centered err.	Regret
Qwen3-MoE	5/20	10.41%	31.63	21.43	1.675
DeepSeek-V2-Lite	11/20	34.69%	9.15	10.03	3.525
GPT-OSS-20B	15/20	54.44%	4.50	8.64	4.900

Controlled-router mechanism audit. The prospective local score-stability audit executes 216 real windowed actions on new OASST1 views. The first acted depth remains aligned at every continuation token. Outside $g_\ell \leq 2d_\ell$, observed mismatch is zero; inside it, mismatch is 16.69–17.92%. Negative behavior gap ranks downstream mismatch with AUROC .821–.874. An exploratory ex-post row-norm reanalysis covers 5.5–15.6% of all positions but only 0.16–0.64% with nonzero hidden perturbation. One Qwen score perturbation exceeds the uncorrected $M_\ell r_\ell$ floating-point bound, although no observed norm-qualified global set changes. These checks establish neither useful pre-execution suffix certification nor the full-state constants.

On the frozen 20-pair Qwen sample, every one of the model's 48 router gates is instrumented during incremental KV-cache execution. Across 291,840 calls, gate weights, logits, softmax, and top-k are FP32; experts and the remaining checkpoint retain their deployed BF16 precision. An action-free repeated run has zero absolute mean-NLL difference and identical argmax outputs. For each early/middle/late intervention, the first acted-on layer has 0/400 route-set mismatches. Strictly downstream mismatch is 3,686/18,800, 2,104/12,400, and 535/6,000, with paired-bootstrap lower bounds 18.62/15.98/7.95%. Replay's tie-broken choice differs in 9/40 views, including one equal-cost alternative; eight views have strictly positive regret. The choice-disagreement pair-bootstrap interval is 12.5–32.5%, and mean deployment regret is 1.775 loads with interval 0.375–3.925. This rules out BF16 gate arithmetic as a necessary cause in this execution class. The observation ladder in Table 12 combines this mechanism sample with the previously frozen fixed-route audits and is not an independent replication.

Complete winner-value acceptance partition. Table 13 reconstructs this same 40-view matrix against the per-action integer cost ledgers. Exactly three selected values agree with replay; one has a better executed challenger. The other 37 selected values are inexact, but that fact alone does not establish a selection error. The partition is retrospective, uses no fitted tolerance, and does not estimate a population error rate conditional on passing a winner check.

The historical downstream localization curve inspects the replay-selected and the eventual deployment-optimal actions, then reports coverage conditional on choice disagreement. The latter action is unavailable to a prospective probe. We retain this only as a post-hoc mechanism diagnostic. An additive non-oracle reconstruction inspects replay's selected action alone, or all three actions with three times the observation count. At two layers, the selected-only alarm has 5 true positives, 23 false positives and 3 false negatives against strictly positive regret; at four layers these counts are 7, 31 and 1. Full-continuation layer logs are not early-token deployment observations, and no partial-execution wall-time saving is measured.

Calibration boundary. The absolute-LCB experiment uses PAWS pairs 0–19 for action-specific shrunken bias fitting, pairs 20–39 for source-pair split-conformal penalties, and pairs 40–79 once for testing. Qwen's safe-minus-naive regret is $+1.2875$ loads/request with a paired interval $[0.0125, 2.5753]$ and only 72.5% simultaneous pair coverage. DeepSeek's difference is $+0.5125$

Table 11: Frozen action-induced cascade audit on 24 unused OASST1 views and three depth windows (72 nontrivial runs/family). "In/out" is native-route mismatch inside/outside $g_\ell \le 2d_\ell$; exact violations count positions where $g_\ell > 2d_\ell$ but global top-k changes. Norm columns report ex-post qualification over all/nonzero-perturbation positions; the latter is an exploratory correction, and neither establishes a pre-execution certificate.

Family	Substitutions	Gap AUROC [pair 95% CI]	In/out mismatch	Norm all/nonzero	First/exact viol.
Qwen	3,097	0.831 [0.825,0.837]	17.92/0.00%	5.5/0.16%	0/0
DeepSeek	1,019	0.874 [0.860,0.886]	16.69/0.00%	15.6/0.64%	0/0
GPT-OSS	1,256	0.821 [0.806,0.833]	17.50/0.00%	15.5/0.53%	0/0

Table 12: Observation diagnostics with distinct targets, not an established cost-efficient acquisition ladder. The mechanism rows reuse Qwen data; complete-trace exactness comes from separate fixed-route audits.

Added observation	Result
Behavior summary	10 choice/gap collisions cover 32/58 views.
Declared FP32 router	9/40 different choices; eight positive-regret views, one equal-cost tie.
First acted-on route	0/1,200 first-layer mismatches; feedback has not occurred.
Post-hoc localization	55.6/88.9/100.0% coverage at 1/2/4 layers, using the eventual best action; not an executable probe.
Complete ordered trace	Fixed-route accounting is exact for 108/108 and 197/197 actions.

Table 13: Winner-value acceptance versus selection on the complete controlled-FP32 Qwen cohort: 40 views from 20 source pairs. The retrospective partition uses exact integer-value equality, without a tolerance search. Regret is relative to the best executed candidate; percentages use each view's native loads before averaging. Mean/max include zero-regret views in each row. The first row is the union of the next two; neither views nor conditional counts are independent population trials.

Winner-value check	Views	Strict errors	Ties*	Loads: mean/max	Native %: mean/max	$> 1\%$
All views	40	8	1	1.775/37	0.094/1.969	1
Winner value exact	3	1	0	1.000/3	0.053/0.159	0
Winner value inexact	37	7	1	1.838/37	0.098/1.969	1

*Different selected actions with equal executed cost. The sole strict error passing the winner check has regret 3/1,881 native loads; this is a missed challenger, not a material deployment correction.

$[-0.100, 1.250]$ despite 92.5% coverage. Neither rule abstains from the underlying action, so the failure is conservative reordering rather than a zero-action artifact.

The paired repair refits Ridge on PAWS pairs 0–59 and calibrates the maximum candidate-wise advantage error on pairs 60–79. The rule is then frozen to 80 disjoint ParaSCI-arXiv pairs with two views. Penalties are 15.749 Qwen and 9.090 DeepSeek loads; proposal rates are 12.5% and 16.9%, but no advantage lower bound is positive. Pair coverage is 97.5% and 93.75%. Thus the frozen rule is no-harm but non-actionable. Removing the penalty is only a post-hoc diagnostic: it gives $+0.788$ $[-0.063, 1.700]$ in Qwen and -0.588 $[-1.069, -0.181]$ in DeepSeek, so it is not promoted as a method.

B Deployment-consistency protocol details

B.1 Fixed-route capture and admission audit

The paired execution-mode audit contains 256 SICK paraphrase pairs (512 inputs), a fixed eight-token continuation, and 18 aligned prompt/continuation positions at three sparse layers per family. Set flips discard top-k internal order. A 5,000-repeat cluster bootstrap resamples source records. The low-margin AUROC uses the full-sequence k versus $k+1$ gap. The FP32 mechanism check reuses captured hidden states and gate weights; it recomputes only the router decision and is not an end-to-end FP32 execution.

For action-value validation, the baseline trace establishes the initial cache, then all 128 single-expert admissions are enumerated per batch/layer cell. The first real comparison exposes the transport failure between full-sequence capture and incremental offload. The repaired collector uses BF16

Table 14: Non-oracle route alarms on the existing 40-view FP32 Qwen audit. "Selected" inspects only replay's action; "all" inspects all three. Depth is measured after each action's first layer across the complete fixed continuation, not a prospective token-prefix execution. Counts are TP/FP/FN/TN.

Policy/depth	Action observations	Positive regret	$> 1\%$ regret
selected/0	40	0/0/8/32	0/0/1/39
selected/1	40	1/9/7/23	0/10/1/29
selected/2	40	5/23/3/9	1/27/0/12
selected/4	40	7/31/1/1	1/37/0/2
selected/8	40	8/32/0/0	1/39/0/0
selected/16	40	8/32/0/0	1/39/0/0
selected/32	40	8/32/0/0	1/39/0/0
all/0	120	0/0/8/32	0/0/1/39
all/1	120	5/15/3/17	0/20/1/19
all/2	120	7/30/1/2	1/36/0/3
all/4	120	8/32/0/0	1/39/0/0
all/8	120	8/32/0/0	1/39/0/0
all/16	120	8/32/0/0	1/39/0/0
all/32	120	8/32/0/0	1/39/0/0

Nine tie-broken choice disagreements contain eight positive-regret views and one material-positive view. The original oracle-action-pair coverage curve is post-hoc localization, not this observation policy. Partial-execution wall times were not measured. No inference of signed value from divergence.

Table 15: Trace semantics determine counterfactual validity. Full and incremental executions are aligned by request and position. Exactness after repair is scoped to ordered, action-invariant, deterministic cache replay.

Audit	Frozen result	Implication
Route transport	DeepSeek flip 6.65% [6.35%, 6.96%]; Qwen flip 12.07% [11.68%, 12.44%]	Full-sequence routes are not exact deployment routes
Full-trace action labels	Only 34.65% of 127 cache action values reproduce exactly; admission agrees on 98.43%	Invalid for fine-grained exact-value claims; coarse admission is more stable
Matched ordered trace	108/108 exact unseen actions; 197/197 exact in the external capacity/continuation audit	Validates dense counterfactual supervision under the stated runtime assumptions
Materiality oracle	Best deployment-consistency increment is 0.356% of baseline loads; 0/216 plausible settings reach 0.5%	Measurement contribution, not a cache-speed or latency claim

token-at-a-time KV-cache inference through the same expert-offload runtime, stores the ordered top-k, and includes prompt-prefix accesses. The 108-action prospective test excludes all previously executed actions and balances positive, near-zero, and deliberately harmful strata. The external validation adds capacities 24, 32, and 40 and three continuations, for 197 real actions.

The materiality sweep compares

$$\text{incremental deployment oracle value} - \text{deployed value of the full-trace oracle action.}$$

It enumerates all 128 admissions plus abstention across capacities 12/16/24/32/40/64, batch sizes 1/2/4/8, admission costs 0.5/1/2 in the primary grid, and positive-value thresholds 0/1/2. Native router order and canonical expert-set order are both evaluated. The canonical result rules out an explanation based only on within-set order and LRU tie-breaking.

An audit of a legacy summary found that grouping by `(batch_id, layer)` omitted continuation and undercounted old multi-continuation oracle totals. No legacy file was overwritten. Those aggregated totals are excluded from the main paper; real-run artifacts and the execution-mode and materiality conclusions are unaffected.

B.2 Exact specification of the route-changing actions

Scope and token protocol. The resident-aware substitution audits use a fully resident model and a *logical* routed-expert cache; no weight transfer is timed here. Each source view is tokenized

independently with the checkpoint tokenizer, `add_special_tokens=True`, retaining its last at most 96 tokens. There is no chat template. The continuation is the literal string

```
␣Please explain this sentence carefully and concisely.
```

where the visible space denotes exactly one leading ASCII space, not an extra token. It is tokenized separately with `add_special_tokens=False`: 10 tokens in Qwen and GPT-OSS, 9 in DeepSeek. These are distinct from the eight-token capture-mode audit above. All tokens, including the prefix, execute one at a time with KV caching, batch size one, evaluation mode and no sampling. Every candidate starts from the same native prefix state, recomputed or exactly restored, and does not inherit another candidate's cache or KV. The intervention begins before the first continuation token is *processed*. Continuation NLL averages the next-token losses over these fixed tokens, including the first token predicted by the untouched prefix. The final continuation token is processed and charged even though its output predicts no further scored token.

Selection rule. At an active MoE layer and continuation token, let p be the FP32 full-softmax router probabilities and R the resident expert set *before* that token's accesses. Initialize U to all expert indices sorted by decreasing p_e and S to the empty ordered list. Repeat k times:

1. Let n be the first index in U and set $q = (1 - \tau)p_n$.
2. Choose the first $e \in U$ with $e \in R$ and $p_e \geq q$; if none exists, choose $e = n$.
3. Append e to S and remove only e from U.

Thus S has exactly k distinct experts. R stays fixed during this selection loop. An unchosen high-score nonresident expert remains in U and can be considered again at the next slot; this is not one-to-one replacement of an already fixed native top-k list. Probability ties inherit the frozen runtime's `torch.argsort(descending=True)` order, not a separately imposed expert-ID tie rule. Outside the active window, and throughout the prefix, the native gate supplies the route. The abstain action calls the native branch, rather than emulating it by a zero-tolerance sorted route.

Expert weights and numerical dispatch. Qwen uses the gate's linear logits and FP32 softmax. DeepSeek uses its FP32 linear logits for substitution probabilities, while retaining the native gate's indices and weights for unchanged dispatch. GPT-OSS uses FP32 affine logits including bias for the substitution probabilities and native router outputs for unchanged dispatch. If the selected ordered list is unchanged, native weights are retained. Otherwise Qwen and GPT-OSS normalize $w_e = p_e / \sum_{j \in S} p_j$; this DeepSeek checkpoint has `norm_topk_prob=False` and routed scale 1, hence $w_e = p_e$ without top-k renormalization. Weights are cast to the hidden-state dtype. For DeepSeek and GPT-OSS, an unchanged *set* delegates to the native MoE forward even if its order differs, avoiding a new rounded accumulation chain. On genuinely changed sets, DeepSeek uses its native expert-inference helper and adds the unchanged shared-expert branch; GPT-OSS passes selected IDs and weights to its expert module. Qwen accumulates expert contributions in increasing expert-ID order with the selected weights. DeepSeek's two shared experts remain active and are excluded from routed-expert load accounting. The controlled-FP32 Qwen experiment changes all gate linear maps to FP32; expert computation and hidden states remain BF16, and the same substitution and cache rules apply.

Cache transition and objective. Each routed MoE layer has its own empty LRU cache at the start of the request. Native prefix accesses warm it; at the continuation boundary we retain its ordered state and reset the miss/admission counters. Deduplicate each route in first-occurrence order. Protect all selected experts already resident; load missing experts in route order, evicting the least recent unprotected, unpinned expert whenever full, and protect each newly loaded expert. Finally move all selected experts to most-recent status in route order. Each load costs one logical unit. There are no pins in the substitution arms. Replay clones the native boundary state and applies this same transition to the stored behavior logits, updating its *own* residency after every token; it does not reuse recorded per-token resident sets. Execution applies the rule to action-induced hidden states and recomputes all affected layers. For native continuation cost C_0, the two values are $\widetilde{v}_a = C_0 - \widetilde{C}_a$ and $v_a = C_0 - C_a$, summing loads over all routed layers, including those outside the window. The fixed-route negative control instead pins the highest-prefix-frequency boundary-nonresident expert per layer (ties: lower

Table 16: Parameters of the archived substitution audits. Layer indices are zero-based decoder indices, with inclusive endpoints. $E/k/c$ denotes routed expert count, active expert count and per-layer logical capacity.

Checkpoint	$E/k/c$	MoE layers	Early	Middle	Late
Qwen3-30B-A3B	128/8/16	0–47	0–15	16–31	32–47
DeepSeek-V2-Lite	64/6/12	1–26	1–8	10–17	19–26
GPT-OSS-20B	32/4/8	0–23	0–7	8–15	16–23

expert ID), if one exists. Its one-time admission costs one, pinned experts cannot be evicted, and neither expert outputs nor routes change.

Windows and experimental mapping. For sorted MoE indices L of length N, use width $w = \lfloor N/3 \rfloor$ and slices $L[0 : w]$, $L[\lfloor (N-w)/2 \rfloor : \lfloor (N-w)/2 \rfloor + w]$, and $L[N-w : N]$. DeepSeek layer 0 is dense; routed layers 9 and 18 are outside all three windows. The windows have equal layer counts *within* each model, not equal numbers of actual substitutions or identical perturbation energy. Attention is SDPA for Qwen/GPT-OSS and eager for DeepSeek. GPT-OSS uses the local dequantized expert-bank path, not native MXFP4 serving kernels. The main comparisons map to these settings as follows:

- Table 2, fixed-route MAE and route-drift range: 20 TaPaCo pairs per Qwen/DeepSeek; the pin control above and all-MoE-layer substitution at $\tau \in \{.05, .10, .20\}$, with the listed capacities.
- The same table's window choice, regret and NLL columns: 20 PAWS pairs per Qwen/DeepSeek, $\tau = .10$ and the three windows above. The controlled-FP32 Qwen audit reuses this PAWS sample and these actions.
- The PAWS/ParaSCI and OASST window audits retain $\tau = .10$ and the listed windows/capacities. Table 4 uses the exhaustive reference matrices of the 32-pair-per-family HH strong-baseline audit, not its stopped subsets; the separate prospective 64-pair materiality audit is DeepSeek with capacity 12 and the same action/token protocol.

Window selection maximizes savings over the ordered candidates `[abstain, early, middle, late]`, using the first maximizer for ties; abstain has value zero. A different tied action is not positive regret. NLL is measured after selection, not used as a quality constraint. Its harm rate denotes $\Delta\mathrm{NLL} > 0$ on this continuation, not task failure.

B.3 Router-precision masks and recorded-slot objective

The auxiliary fake-W4A4 experiment uses the same prefix/continuation protocol but no resident-aware substitutions. For sorted MoE indices L, mask r is $L[r :: 3]$, $r \in \{0, 1, 2\}$. The three mask sizes are 16/16/16 for Qwen, 9/9/8 for DeepSeek, and 8/8/8 for GPT-OSS. They span depths and are balanced to within one layer; DeepSeek is not an exactly equal-size comparison. Only masked routers change, only during continuation; expert computation and routers outside the mask follow the original model.

For each activation row or router-weight output row x, define

$$s(x) = \max(\|x\|_\infty, 10^{-8})/7, \qquad Q_4(x) = s(x)\ \mathrm{clip}(\mathrm{round}(x/s(x)), -7, 7).$$

Arithmetic is FP32 with the runtime's round-to-nearest-even operation; weights are quantized per output row and activations per token. The fake logits are $Q_4(h)Q_4(W)^\top + b$, with any bias unquantized in FP32. Qwen returns these logits cast to the native router output dtype; DeepSeek applies full-softmax/top-k and its native scaling policy; GPT-OSS normalizes the selected affine logits and returns native-format indices/weights. This is quantize–dequantize reference arithmetic, not packed integer kernels.

Let $R^0_{\ell t}$ be the native recorded set, $R^{q,0}_{\ell t}$ the fake router set evaluated on behavior states, and $R^a_{\ell t}$ the recorded set from the acted run. With $d(A, B) = |A \setminus B|$, the archived objectives are

$$\widetilde{v}_a = -\sum_{\ell \in a,t} d(R^0_{\ell t}, R^{q,0}_{\ell t}), \qquad v_a = -\sum_{\ell,t} d(R^0_{\ell t}, R^a_{\ell t}).$$

Choose among the three masks, without abstention, minimizing recorded-slot change (equivalently maximizing these values), with phase order 0,1,2 for ties. The on-policy run includes downstream feedback outside the mask. For Qwen, the auditor records fake top-k IDs from FP32 logits *before* the router return is cast back to its native dtype; the actual expert module receives the cast logits. Thus this auxiliary endpoint is the auditor's router-reference slot count, not separately certified post-cast dispatch parity, task quality, memory or latency. It must not be conflated with the ordered logical-load endpoint of the substitution audits.

C CACHE-MEMORY MECHANISM AND DIAGNOSTIC SCOPE

Why a local route check can miss a later cost. Let R_t^0, R_t^a be the ordered expert accesses and S_t^0, S_t^a the pre-access ordered cache states in replay and action execution. For the declared logical cache, event cost is $c(R, S)$, the number of selected experts absent from the resident set; selected resident experts are protected during the event. Its error has the elementary decomposition

$$e_t = c(R_t^a, S_t^a) - c(R_t^0, S_t^0) = \underbrace{c(R_t^a, S_t^0) - c(R_t^0, S_t^0)}_{D_t} + \underbrace{c(R_t^a, S_t^a) - c(R_t^a, S_t^0)}_{B_t}.$$

The intermediate reference affects attribution to D_t and B_t, so this decomposition is not a general claim that state amplifies or damps error. However, if $R_t^a = R_t^0$, the direct demand term is zero under either reference: nonzero error then requires different resident states. Equal route sets and resident sets suffice for this event's cost, but not its successor's state. LRU order can affect later evictions. An equal cache state alone also does not certify future neural routes; hidden state, KV, and subsequent actions matter.

Discovery example. Figure 2 uses the original view of discovery OASST pair zero, an early intervention, and downstream layer 22. At continuation event one the replay accesses are $(74, 30, 26, 35, 113, 112, 98, 62)$ and actual accesses are $(74, 30, 26, 35, 113, 112, 98, 50)$; each loads one expert. At event two both access $(74, 112, 30, 26, 84, 98, 110, 62)$, but replay loads only 84 whereas actual execution loads 84 and 62. The intervention acts earlier than layer 22; the figure depicts its downstream consequence, not a direct modification at the displayed layer.

Frozen follow-up and units. The follow-up fixes pairs 4–7, both views, from each of PAWS and OASST. Normalized-text overlap with discovery pairs 0–3 is zero. The corpora were used in earlier experiments. Qwen3-30B-A3B uses logical capacity 16 and SDPA; DeepSeek-V2-Lite uses capacity 12 and eager attention. The three original early/middle/late windows, substitution tolerance 0.1, maximum prefix length 96 and literal continuation *Please explain this sentence carefully and concisely.* are fixed. The tokenizers produce ten Qwen and nine DeepSeek continuation tokens. Eight distinct source pairs supply sixteen inputs reused across two models: 32 model–views, 96 non-null action labels and 34,272 layer–token–action events. Events and repeated model evaluations are not independent source pairs. The models are fully resident; costs are logical loads, not H2D timings.

Diagnostic corrections and strong baseline. Starting from replay cost, a correction adds the *exact* observed e_t only at flagged events. The route-set-only ablation flags unequal sets; the stronger route-order-only ablation flags unequal ordered lists. The state-aware variants additionally flag unequal resident sets or ordered cache states, respectively, checking each event rather than skipping a suffix after one match. All variants already receive executed action traces. They are diagnostic ablations, not cheap pre-action predictors or verified external implementations. Ordinary full ordered-route replay from the exact boundary state is the strong baseline, independently reconstructed and checked against recorded action costs.

There are 709 equal-ordered-route events with nonzero cost error, affecting all four pairs in each cell of Table 3. Route-order-only correction leaves 74/96 labels wrong and 3/32 positive-regret views, but no residual regret above 1% of native loads. Original replay has respectively 1/8, 4/8, 2/8, 2/8 positive-regret views in Qwen/PAWS, Qwen/OASST, DeepSeek/PAWS, DeepSeek/OASST, including 1, 1, 0, 1 above-threshold views. Thus both route-only corrections already remove the observed larger errors; residual label failures cannot establish additional material decision benefit. The state-aware variants and ordinary replay recover all 96 labels exactly. All aggregate choices agree (early, middle, late, late in cell order).

Table 17: Frozen natural-response pilot: eight source trees reused across models, capacities and nested token horizons. Same-route events have identical ordered accesses but different cost. Ratio columns average per-source ratios. All aggregate winners agree; all zero-denominator counts are zero.

Family	Cap.	Tokens	Same-route cost events	Wrong labels	Positive regret	$> 1\%$ native	Mean R (loads)	Mean R/C_0 (%)	Mean R/S (%)
Qwen	16	16	344	22/24	1/8	0/8	1.375	0.064	2.218
Qwen	16	128	3324	21/24	2/8	0/8	15.875	0.083	2.442
Qwen	32	16	265	21/24	0/8	0/8	0.000	0.000	0.000
Qwen	32	128	2645	24/24	0/8	0/8	0.000	0.000	0.000
DeepSeek	12	16	181	23/24	2/8	0/8	1.125	0.074	4.172
DeepSeek	12	128	1788	24/24	0/8	0/8	0.000	0.000	0.000
DeepSeek	24	16	219	22/24	3/8	1/8	2.375	0.241	6.554
DeepSeek	24	128	2554	23/24	1/8	0/8	0.625	0.008	0.198

What this predicts and what remains open. The mechanism predicts where a route-local check can miss cost: after current accesses agree but resident states do not. It does not predict which requests will exceed a decision budget. The paired-error identity in Equation (1) still requires challenger errors and margins. A subsequent fixed replay-only stress screen on these small archived cohorts did not improve on propagation-based ordering in Qwen and missed the sole DeepSeek material-positive pair. It supplies neither a calibrated risk predictor nor an independently confirmed selective re-evaluation rule.

C.1 A BOUNDED NATURAL-RESPONSE HORIZON AND CAPACITY CHECK

To test beyond the common artificial continuation, this pilot fixes eight English OASST validation user–assistant pairs from distinct conversation trees before execution. It selects nondeleted, rank-zero responses marked nonsynthetic, at least 220 words long, with user prompts of at least eight words. Historical local JSON trees and normalized texts are excluded; the first eight distinct trees in a seeded SHA256 order are retained. This selects new sources within OASST. Prompts are capped at 96 tokens, without a chat template; each recorded response supplies a fixed 128-token continuation whose first 16 tokens define the short condition. Qwen capacities are 16/32, DeepSeek 12/24; tolerance 0.1 and the three windows remain unchanged. No action search, free generation or additional tolerance sweep is performed.

The short condition is recovered from its causal prefix of the long run. For source zero at both capacities, all four actions are additionally executed for 16 tokens independently: ordered router records, boundary caches, stored prediction logits, token NLL, argmax and cost all agree with the long-run prefix. Native-wrapper parity passes at both capacities; ordered replay independently reconstructs all 192 non-null action costs. Models share the same eight sources; horizons share tokens. The 64 source–condition observations are not 64 independent samples.

All eight cells contain same-ordered-route cost discrepancies, affecting all eight sources per cell. Nevertheless, 180/192 inexact labels correspond to only 9/64 positive-regret observations and one above 1% of native loads. That exception is DeepSeek, capacity 24, 16 tokens: source two loses 11 loads out of 923 native loads, or 29.73% of its available saving of 37. All eight aggregate choices agree (early in Qwen, late in DeepSeek), with no observed native-load harm. These counts describe this small frame, not a population safety certificate or an independence-based significance test.

Changing the budget denominator. A retrospective scan of the same matrices compares thresholds of 0.5, 1, 2, 5, 10 and 20% against native cost and available saving. At 1%, the counts above threshold are 1/64 and 9/64 respectively; 5/64 exceed 10% of available saving. All denominators are positive. The absolute losses and all eight aggregate choices are unchanged. A saving-relative budget asks how much of the candidate set's opportunity may be lost; a native-relative budget limits additional cost relative to the unmodified execution. The same percentage therefore represents different tolerances. The scan is descriptive, uses the same eight sources, and does not calibrate either tolerance to latency or quality. The evidence entry in Appendix J reports every cell and threshold alongside absolute regret.

Table 18: Occupancy-conditioned demand discrepancies at 128 tokens. Each action trace is held fixed across capacities. Counts are events with $D_t \neq 0$, not total label error or independent samples. The exploratory eight source trees suggested Qwen-down/DeepSeek-up; twelve excluded trees then tested these fixed directions. The predeclared rule required at least 10/12 strictly same-direction sources and a matching pooled direction per family; both pass. This is a directional replication rule, not a significance level.

Cohort	Family	Sources	Capacity	Low count	High count	Same direction
Exploratory	Qwen	8	$16 \to 32$	3936	3235	8/8
Exploratory	DeepSeek	8	$12 \to 24$	1022	1192	8/8
Held-out	Qwen	12	$16 \to 32$	5267	4330	12/12
Held-out	DeepSeek	12	$12 \to 24$	1438	1697	11/12

Neither horizon nor capacity has a uniformly monotone effect on selection loss. Longer Qwen sequences at capacity 16 increase mean absolute regret from 1.375 to 15.875 loads while native-relative regret changes only from 0.064% to 0.083%. DeepSeek's longer sequences instead reduce mean regret at both capacities. Higher capacity removes Qwen's observed errors but does not do so uniformly in DeepSeek. Candidate margins and paired errors both change: for example, in DeepSeek at capacity 12 their respective means change from 8.125/8.375 loads at 16 tokens to 59.5/20.75 at 128 tokens. These are ex-post summaries; mean margins do not certify individual choices. Actual substitution counts are retained because capacity and depth also change intervention strength, precluding a pure matched-strength causal interpretation of these contrasts.

C.2 Holding the intervention fixed: occupancy and held-out confirmation

A capacity-independent intervention. The preceding resident-aware actions change when capacity changes. To remove that confound, a separate control applies exactly one expert substitution at continuation token zero, at the first layer of each early/middle/late window. For each candidate, the lowest-probability selected expert is replaced by the highest-probability unselected expert, with expert-ID tie-breaking and the original slot order and weight normalization retained. All three plans are recorded from the native run before any action outcome. No further substitution is applied, although altered hidden/KV states can change subsequent native routes. This matches intervention count, not output-perturbation magnitude across depths. These are diagnostic actions, not a proposed deployment policy.

Each actual route stream is replayed under both capacities, with the appropriate initial cache independently reconstructed from its full ordered prompt accesses. The eight natural-pilot source trees supply an exploratory cohort. A second cohort fixes twelve distinct English OASST validation trees after excluding historical tree IDs and normalized texts, including the pilot. The same text eligibility and tokenization rules apply; a new fixed SHA256 ordering selects twelve of fifteen eligible trees without looking at model outcomes. This is a held-out source cohort within the same sampling frame. Both models reuse the twelve trees and nested 16/128-token continuations.

The original capture implementation is unchanged. Native-wrapper parity, exactly one selected-set substitution, and pre-injection route/logit agreement pass. On source zero, native and all three actions are independently executed for sixteen tokens at the other capacity: recorded router/prediction logits, routes, NLL and argmax match the long-run neural prefix; each capacity's cache boundary and cost are verified separately. Ordinary ordered replay reconstructs all captured costs. No truncated input pair overlaps the exploratory cohort.

Why occupancy can reverse cost sensitivity. For equal-cardinality selected sets, the elementary current-demand term is

$$D_t = |(R_t^0 \setminus R_t^a) \cap S_t^0| - |(R_t^a \setminus R_t^0) \cap S_t^0|,$$

where intersections refer to resident membership. Different experts can have equal cost because both sides miss, both hit, or their hit counts balance. Increasing capacity can expose a previously balanced pair of misses, or remove a discrepancy by making both sides hit. The identity is an accounting check; the new empirical test concerns the direction suggested by the exploratory cohort. At 128 tokens, the held-out cohort confirms Qwen-down in 12/12 sources and DeepSeek-up in 11/12 (Table 18); the opposite DeepSeek source is retained. The frozen rule requires at least 10/12 strict

Table 19: Selection consequences in the twelve-source held-out pulse cohort. All rows retain abstain; sources are reused across models and nested horizons. Ratios average per-source values: parentheses give the number of defined R/S ratios out of twelve; zero-saving rows remain in all other summaries, and ratios are not clipped. The last column counts positive regret among unique replay winners. All unique winning margins are one logical load; only DeepSeek capacity 24 has the predeclared minimum of six such sources per cell, with the same six reused at both horizons.

Family / cap.	Tokens	$R > 0$	$R/C_0 > 1\%$	Mean R loads	Mean R/C_0 (%)	Mean R/S (%; defined n)	Non-tie $R > 0$ / unique
Qwen 16	16	10/12	0/12	4.167	0.169	115.56 (9)	2/2
Qwen 16	128	9/12	0/12	16.000	0.081	96.67 (10)	2/2
Qwen 32	16	9/12	0/12	2.917	0.236	90.00 (10)	1/2
Qwen 32	128	7/12	0/12	10.500	0.110	137.50 (8)	1/2
DeepSeek 12	16	6/12	0/12	3.000	0.201	164.29 (7)	3/5
DeepSeek 12	128	9/12	0/12	6.417	0.055	198.33 (10)	3/5
DeepSeek 24	16	10/12	1/12	5.833	0.597	105.00 (10)	5/6
DeepSeek 24	128	9/12	0/12	11.417	0.155	224.63 (9)	5/6

source-wise directions and a matching pooled direction for each family, not a binomial significance claim. Qwen has 1,384 both-miss-to-unbalanced transitions versus 2,325 unbalanced-to-both-hit transitions; DeepSeek has 679 versus 420. All mixed-cancellation transitions are retained in the artifact.

Equal-ordered-route cost discrepancies still affect every held-out source in each cell. A separate cache-only branch at token sixteen feeds replay and action caches identical future accesses, isolating inherited state from new route differences. Full-state coalescence time and miss-cost exposure are distinct: the larger cache's slower coalescence follows ordinary LRU nesting. Neither that property nor a cache hit rate alone predicts the direction of total error or its candidate-wise cancellation.

Selection consequences and limits. The held-out cohort has 264/288 inexact non-null labels, 69/96 positive-regret source–condition observations, and only 1/96 above 1% of native loads (Table 19). These are not 96 independent source trees. Unique replay winners occur on only two Qwen sources per cell and five DeepSeek capacity-12 sources. At DeepSeek capacity 24, six sources have unique winners, with five positive regrets at each horizon; the same six sources are reused and every winning margin is one load. Thus errors are not exclusively tie-breaking effects, but no large-margin generalization is established.

The sole above-threshold case is DeepSeek, capacity 24, sixteen tokens, held-out source five. Replay selects early at 1,014 loads against native's 1,015; execution costs 1,043 and native is optimal: regret and native harm are 28 loads (2.76%), with zero available saving. Its demand-only correction already selects native, so this example does not establish incremental decision value from the state term. Four of eight aggregate cells change their optimum with strict regret, still at small logical-load cost. The occupancy direction is independently confirmed within this frame; cheap pre-action budget-risk prediction, material serving benefit, free-generation quality, and transfer to larger checkpoints or other runtimes remain untested here.

D Invalidity versus aggregate method-selection consequence

A retrospective check asks whether the existing complete labels change a concrete method-selection decision. For each family, separately minimize the aggregate replay and deployed costs on the 48-source OASST cohort over the four declared constant actions (abstain/early/middle/late). Apply both choices to the already evaluated 48-source HH cohort, without using HH labels for selection. This retrospective split comparison reuses known archives. Each source averages its two views; timing repeats are verified equal in labels and are not counted as new observations.

Replay and exact labels choose middle/middle for Qwen and late/late for both DeepSeek and GPT-OSS. Thus primary held-out selection gain is exactly zero in all three families (Table 20). The request-adaptive replay selector also retains lower point-estimate mean deployed cost than the exact-selected constant in every family; its Qwen interval includes zero. Per-request oracle regret is only

Table 20: Exact invalidity need not change aggregate method selection. Choose one constant window on the 48-source OASST cohort using replay or complete deployed costs; evaluate both on the existing 48-source HH cohort. Gain is replay-selected minus exact-selected true cost. Last column instead compares the request-adaptive replay selector minus the exact-selected constant; negative means adaptive replay remains better. Units are logical loads/request; brackets are descriptive source-bootstrap intervals, not risk certificates.

Family	Replay/exact choice	Constant gain	Adaptive minus constant [95%]
Qwen	middle/middle	0.00	$-0.33\,[-2.03, +1.45]$
DeepSeek	late/late	0.00	$-1.65\,[-2.60, -0.75]$
GPT-OSS	late/late	0.00	$-1.48\,[-2.60, -0.33]$

3.0625/1.28125/0.90625 loads in Qwen/DeepSeek/GPT-OSS, and access to that evaluation oracle is not an implementable selection method.

The paired-error identity also applies to aggregate costs. If a minimizes mean replay cost $\bar{R}$, set $\bar{\delta}_j = \bar{C}_j - \bar{R}_j$. Its mean deployed margin against j is $(\bar{R}_j - \bar{R}_a) + (\bar{\delta}_j - \bar{\delta}_a)$. For example, Qwen's OASST early-versus-middle replay margin is 5.8646 loads. The mean signed error contrast is -2.5000, leaving a positive deployed margin of 3.3646, even though the mean absolute request-level error contrast is 14.2292. Large individual errors can cancel in aggregate and need not reverse the selected method.

The exact-choice screen requires complete development labels: per-repeat full audit costs are approximately 513/314/238 seconds for Qwen/DeepSeek/GPT-OSS, including captures and action execution, versus 364/265/204 seconds for the witness-only audit. The cheaper witness does not identify the optimum. Logical loads cannot be converted to deployment-time payback without a separately measured transfer model; these archives also lack intervention task-quality labels. The declared retrospective advance gate (at least 1% of no-action loads and a positive descriptive lower bound on primary gain) fails in all families. The audit detects invalid labels but leaves these aggregate choices unchanged.

E PROSPECTIVE MATERIAL-RISK AUDIT AND APPLICATION BOUNDARY

The frame contains 64 previously unused HH roots with mutually unique request texts and exclusions against all earlier data. A single saved Fisher–Yates permutation defines the acquisition order. Checkpoint, capacity 12, tolerance .1, 96-token prefix limit and fixed continuation are frozen before outcomes. The source event is $100(C_{\hat{a}} - \min_b C_b) > C_0$ for either view; equality is negative. Small positive regret must not prematurely stop material-witness search. This threshold measures logical loads, not an established latency SLO.

For s positives among n observations, let $L_K = \binom{K}{s}\binom{N-K}{n-s}$, zero for infeasible counts. The standard uniform-prior confidence sequence used here is

$$C_n = \left\{ K : \frac{\sum_{j=0}^{N} L_j}{(N+1)L_K} < 1/\alpha \right\}, \qquad N = 64, \quad \alpha = .05.$$

Zero-denominator totals are excluded. For each candidate total K, the mixture-to-P_K likelihood ratio is a nonnegative test supermartingale on histories supported by P_K. Its inverted confidence sequence simultaneously covers the fixed true K with probability at least $1 - \alpha$ over the initial random ordering (Waudby-Smith & Ramdas, 2020). Rejecting $K \leq 6$ excludes that true total on a null frame, so the same error bound applies without a union bound over possible totals. At census the set is the exact singleton. The frame need not represent all future HH requests. Order-invariant measurement is an assumption; repeat parity supports but does not universally prove it.

Sequential execution writes a durable terminal receipt before the independent census starts. Both arms use the same ordinary stopped/lazy terminal-KV evaluator and pay capture, copies and actual queries. Historical all-action logits/KV/logical-state byte parity and untouched native-wrapper parity pass; fresh repeated rows are checked for scalar-label and query-path parity, not claimed as an unarchived all-row byte comparison. All null totals at $N = 64$ are also checked by exact probability recursion; this verifies code, not an empirical false-positive rate from one live trial.

Table 21: Prospective DeepSeek finite-frame audit at the frozen 1% native-load regret threshold. Uniform random order of 64 excluded HH source pairs; the null permits at most six material-positive pairs. Both arms use ordinary stopped/lazy terminal-KV evaluation. Sequential execution ends and saves a terminal receipt before the independently paid census begins.

Arm	Pairs	Views	Queries	Evaluator s	Wall s
Sequential	64	126	375	316.54	332.41
Independent census	64	126	375	314.99	331.82

Census: 4/64 material positives; terminal confidence set $\{4\}$. Unexecuted sources/queries: 0/0. No early stopping or cross-source work saving. Fixed outer arm order makes wall-time differences descriptive, not an algorithmic advantage. Model loading 11.23s; historical/native preflight 27.26s; total research wall time 702.92s. The finite-frame result does not certify future requests or task quality.

Table 22: Retrospective logical-load admission screen on existing HH records. Admit replay's action if it predicts at least 1% savings versus native; otherwise abstain. "Breach" means actual savings below that promise, "harm" means more loads than native. These are not latency or quality guarantees.

Family	Admitted	Breach	Harm	Regret > 0	Regret $> 1\%$
qwen	96/96	1	0	36	3
deepseek	96/96	1	0	30	3
gptoss	96/96	0	0	20	11

All requests are admitted; the gate changes no choices. Lower cost than a development-selected constant is already present in replay's policy, not caused by this audit. No new held-out study or corrected deployment gain.

Checking a native-relative saving promise. The retrospective admission screen asks whether a one-percent logical-load saving promised by replay survives execution. It reuses the existing OASST/HH archives and keeps all families. All 96 views per family are admitted, with 1/1/0 promise breaches and no native-load harm in Qwen/DeepSeek/GPT-OSS. There are 36/30/20 positive-regret views, but only 3/3/11 above one percent of native loads. This distinction explains why finding many oracle mistakes is not itself a consequential evaluation correction. The policy's mean advantage over a development-selected constant is already present without our audit. This screen measures only the load-saving promise; it has no accompanying task-quality labels.

Related-work boundary. An existing trace-driven study already obtains policy-ranking reversals by correcting event-atomic accounting on a fixed route stream (Zhang, 2026). Its reported scope excludes expert substitution and reduced-precision fallback; our distinction is the action-induced stream, not the general observation that replay semantics matter. Public cache-aware rerouting code executes modified expert outputs and scores resulting logits; REAP's pruning pipeline likewise evaluates the modified model (Suram, 2026; Lasby et al., 2026). Both acquire intervention-specific quality evidence by executing the modified model.

F SAME-COHORT MATERIALITY AND QUERY-ORDER HEADROOM

This retrospective diagnostic uses the complete verification-only action matrices from the fresh strong-reference comparison (Table 46): 32 Qwen and 32 DeepSeek source pairs, with two text views per pair. The two families use different sources. Timing repeats do not increase the task sample size. The threshold grid was specified for this retrospective analysis after the original data evaluation.

For native cost $C_0 > 0$, replay-selected action $\widehat{a}$ and challenger b, the event is $10000[C(\widehat{a}) - C(b)] > tC_0$, where t is an integer basis-point threshold. Thus $t = 100$ is the earlier one-percent logical-load anchor. It specifies cost tolerance, independently of intervention strength; latency and quality would require separate calibration. Integer comparisons retain strict endpoints. Fixed action order breaks minimum-cost ties; different selected names with equal actual cost do not count as positive regret.

Paid observations and comparator class. The ordinary policy captures view zero first, queries its replay-selected action, and visits challengers by replay cost and action name. The native abstain value is available only after paying capture. A witness stops the pair; otherwise both views and all six

Table 23: Same-cohort retrospective materiality–query profile. Q/D denotes Qwen/DeepSeek, each with 32 source pairs (64 views). Every threshold is strict and relative to native logical loads; the full grid is retained. A complete census costs 192 non-null queries per family. Oracles optimize counts within the selected-first, exhaustive-negative protocol, not elapsed time.

Threshold (%)	Positive pairs Q/D	Fixed Q/D	Oracle: fixed views	Oracle: free views
0.00	20/13	148/155	145/152	112/140
0.01	20/13	148/155	145/152	112/140
0.05	20/13	148/155	145/152	112/140
0.10	17/13	153/155	151/152	124/140
0.25	14/10	162/167	157/164	136/152
0.50	11/5	164/179	160/178	148/172
1.00	1/2	188/184	188/184	188/184
2.00	0/0	192/192	192/192	192/192
5.00	0/0	192/192	192/192	192/192
10.00	0/0	192/192	192/192	192/192

Table 24: Paid native captures for the same-cohort query profile. Zero denotes strictly positive regret, not acceptance of zero-regret ties. Captures are separate costs; fewer queries do not by themselves establish a time saving.

Family, threshold	Fixed	Oracle: fixed views	Oracle: free views	Census
Qwen, 0%	55	55	44	64
Qwen, 1%	63	63	63	64
DeepSeek, 0%	55	55	51	64
DeepSeek, 1%	62	62	62	64

non-null actions are exhausted. The fixed-view oracle enumerates all 6×6 challenger permutations; the free-view oracle also permits reversing the views (72 schedules). Both pay selected-action queries and require an executed challenger to witness failure. Full labels choose the ordering only post hoc, not as free evidence. The objective minimizes queries, then captures, not elapsed time. Exhaustive-negative stopping is a restriction of this protocol class, not a universal lower bound for algorithms using structural certificates.

Table 4B in the main text shows selected thresholds and aggregate choices; the complete query grid and corresponding capture counts follow here.

Results and mechanism. Strict positive regret occurs in 22/64 Qwen and 16/64 DeepSeek views; two additional DeepSeek choice changes are cost ties. At the one-percent anchor only one Qwen and two DeepSeek views qualify, in distinct pairs. Ordinary ordering uses 188 and 184 of 192 queries; both oracles give exactly the same counts. Every qualifying witness is already the first challenger in view zero. Each positive therefore needs one capture and two queries, while each negative needs two captures and six queries in this protocol. With $N = 32$ and $k = 1$ or 2 positives, the corresponding minima are $6(N - k) + 2k$ queries and $2(N - k) + k$ captures, attained by the ordinary order. These counts apply to this protocol; elapsed time also depends on capture and action-execution costs.

The full grid also preserves the remaining small-error opportunity. At zero threshold the fixed-view oracle saves only three additional queries per family, while free view ordering saves 36 in Qwen and 15 in DeepSeek relative to ordinary ordering. This is nondeployable hindsight about which view has a witness, not a demonstrated cheap predictor. Raising the threshold cannot make the first qualifying witness earlier along a fixed schedule.

No selected action has observed native-load harm in these cohorts. The maximum view-level regret is 1.477% of native loads in Qwen and 1.214% in DeepSeek. Retrospective constant-window selection agrees between replay and actual labels: middle in Qwen, late in DeepSeek. This is not a held-out training/evaluation split or a future-risk guarantee. A witness demonstrates failure of the selected action at the threshold; it does not by itself identify the optimal action or certify task quality.

Regret units without changing the data or decision problem. For each view let $R = C(\widehat{a}) - \min_a C(a)$, $C_0 = C(\text{abstain})$ and $S = C_0 - \min_a C(a)$. Table 25 reports R, R/C_0 and R/S side

Table 25: Three regret units on the complete same-cohort matrices (32 source pairs, 64 views per family). Means average per-view quantities; P95 uses linear empirical interpolation. Available saving includes the native abstain option.

	Qwen			DeepSeek		
Metric	Mean	P95	Max	Mean	P95	Max
Regret (loads)	3.062	16.000	24.000	1.078	6.850	11.000
Regret / native (%)	0.173	0.867	1.477	0.126	0.813	1.214
Regret / available saving (%)	6.298	30.432	41.026	5.761	35.646	68.750
Available saving (loads)	52.188	71.550	83.000	19.531	26.850	30.000

Undefined native/saving denominators: 0/0 in Qwen, 0/0 in DeepSeek. No rows are removed or ratios clipped. All three regret medians are zero in both families. These are the same HH matrices as Table 4, not a new experiment.

Table 26: Ratios of requestwise totals are distinct from mean per-view ratios and from selecting one constant action for the cohort. R denotes regret, C_0 native loads and $S = C_0 - \min_a C_a$ available saving. The constant-action regret is separately recomputed from aggregate candidate costs.

	$\sum R$	$\sum C_0$	$\sum S$	$100 \sum R / \sum C_0$	$100 \sum R / \sum S$	Constant regret
Qwen	196	112197	3340	0.175	5.868	0
DeepSeek	69	54265	1250	0.127	5.520	0

by side; Table 26 separately reports ratios of totals. Including abstain ensures $S \geq 0$. At $C_0 = 0$ or $S = 0$ the corresponding ratio is undefined, not zero: the row remains in absolute regret and harm counts, with undefined counts reported. Ratios are not clipped, including when an action harms native performance and $R/S > 1$. All 64 views per family are retained; here neither denominator is zero.

The requestwise loss totals are 196 Qwen and 69 DeepSeek loads, respectively 5.87% and 5.52% of their summed requestwise available savings, but only 0.175% and 0.127% of native loads. The mean of per-view saving-relative ratios is instead 6.30% and 5.76%. Thus a small native-relative loss can consume a noticeable fraction of a small intervention opportunity. This comparison expresses the same losses against two cost scales. Constant-action selection still has zero regret in both cohorts; summing requestwise regrets does not define that decision.

Verification and timing limit. Complete source hashes and the original strong-reference summaries are rechecked. At zero threshold, reconstructed capture/query paths match all three measured arms and both repeats. Oracle enumeration matches an independent count formula; synthetic tests and full-grid monotonicity checks pass. Terminal-KV timings are censored after the original first witness, so unqueried actions lack measured terminal time. Complete prefix-reference timings cannot replace them. Tables 23 and 24 therefore report counts only. This diagnosis connects error size and acquisition headroom within each cohort without combining unrelated studies into a speed curve.

G Observed NLL constraints: protocol and complete results

We re-analyze the archived candidate matrices to ask how much saving remains after imposing an NLL budget. All values come from the original executions; this analysis adds the selection comparison.

Data. Five frozen cohorts are used: the PAWS 20-pair test matrices for Qwen and DeepSeek, and the OASST random-29-pair matrices for Qwen, DeepSeek and GPT-OSS (254 views in total; each source pair contributes two views, one per member request as intervention target). Each selection row stores, for every declared candidate window, the transported (replay) logical-load value, the executed logical-load value, and the executed continuation-NLL change relative to the unmodified continuation; abstain is always available with zero saving and zero NLL change. Dataset and per-cohort protocol hashes are recorded in the analysis artifact (`quality_endpoint_results.json`). The continuation is the same literal sentence for every request, " Please explain this sentence carefully and concisely." (10 tokens for Qwen/GPT-OSS, 9 for DeepSeek; Appendix B.2). NLL therefore measures the likelihood

Table 27: Observed fixed-continuation NLL screening, PAWS/OASST matrices (retrospective; five cohorts, 254 views; $\varepsilon = 0$): per-cohort ranges; saving relative to the quality-blind oracle. Zero violations are enforced on observed NLL, not a held-out guarantee. Protocol and complete results: Appendix G.

Selection rule	Executes	NLL violations	Abstain	Saving kept
Replay argmax	none	.448–.603	0	.89–.95
Winner-only check	winner	0	.448–.603	.38–.52
Complete audit	all candidates	0	.138–.275	.63–.79

of this prescribed continuation. The 254 model–view observations reuse source prompts across families; we analyze uncertainty separately within each cohort.

Decision problem and selection rules. An evaluator accepts a window's saving only if its executed continuation-NLL change stays within budget ε (primary analysis $\varepsilon = 0$; ties count as within budget). Four rules are compared per view, with candidate order abstain/early/middle/late breaking ties. *Replay argmax* selects the stored replay winner (the transported argmax) with no executed evidence. *Winner-only check* executes only the replay winner and abstains if its NLL change exceeds ε; this needs one intervention execution per view in addition to the shared no-action reference. *Complete audit* evaluates all candidates and selects the one with the largest executed saving among those within budget. Abstention is included and wins when feasible interventions have negative savings. We use the complete executed matrix already collected for C2. *Cost oracle* selects the executed-cost argmax regardless of NLL, providing the quality-blind reference ceiling.

Metrics and uncertainty. Per cohort and rule we report the harm rate (fraction of views whose selected action exceeds the budget; abstention never harms), the abstain rate, the mean executed saving in logical loads (abstentions contribute zero), and the retained fraction of the cost oracle's mean saving. Intervals are 95% percentile intervals from a cluster bootstrap that resamples source pairs with both views (B=10,000, fixed seed). Both budget-enforcing rules use the same NLL values for selection and reporting, so their observed violation rates are identically zero. Bootstrap intervals describe variation across the sampled pairs; they provide no quality guarantee for new continuations.

Results. Table 28 gives the complete grid. Cost-ranked replay exceeds the observed NLL budget on 44.8–60.3% of views while retaining 89–95% of oracle saving; the winner-only check retains 38–52%; the complete audit retains 63–79% with 14–28% abstention. Even the cost oracle exceeds the NLL budget on 44.8–58.6% of views, showing why correcting costs alone leaves this acceptance question unresolved. Full screening has a built-in advantage: it can choose every feasible action available to winner-only screening, plus the other windows. The measured retention rates quantify how much those additional choices recover. Its selected mean NLL change ranges from -0.011 to -0.052 on the same records used for selection. Table 29 varies the budget: at $\varepsilon = 0.01$ replay harm persists at 30–50% while the audit retains 83–88% with abstention at or below 10.4%; at $\varepsilon = 0.05$ the budget-violation rate is 0–10% outside GPT-OSS (22.4%). Among views where the replay and executed argmax windows differ, the replay choice's mean NLL change is positive in four of five cohorts but small throughout.

Separating candidate coverage from cost ranking. An additional control measures the same complete feasible set using executed NLL, but ranks its members by replay saving. Changing only that ranking to executed saving isolates the conditional increment from correcting cost order. Table 30 compares it with the full increment over winner-only screening, and reports how often screening recovers a non-null candidate after the winner was rejected. At $\varepsilon = 0$, cost-order correction explains 6.3–16.7% of the increment. Thus most of this example's gain comes from inspecting alternatives, not repairing their cost ranking. This decomposition depends on the stated comparison order. Every full-set rule has already acquired all candidate NLL values. Paired intervals use the source-pair bootstrap described above.

Scope. This retrospective comparison uses a prescribed continuation and a budget uncalibrated to downstream task quality (cf. the loss thresholds in Appendix F). Savings exclude the cost of evaluating candidates. The result measures available savings after a finite NLL check; deployment would require accounting for that evaluation cost and testing quality on the intended workload.

Table 28: Complete quality-endpoint results at $\varepsilon = 0$: per-cohort harm rate (accepted action with executed continuation-NLL change above budget), abstain rate, mean executed saving in logical loads, and the fraction of the quality-blind cost oracle's mean saving retained. Brackets are 95% cluster-bootstrap percentile intervals over source pairs (B=10,000); the two budget-enforcing rules have zero harm by construction, so no interval is shown.

Selection rule	Harm rate	Abstain	Saving (loads)	Retained
PAWS / Qwen (20 source pairs, 40 views)				
Replay argmax	.500 [.350, .650]	.000	48.8 [45.8, 51.6]	.949
Winner-only check	0	.500	25.1 [17.3, 32.8]	.487
Complete audit	0	.150	38.6 [31.9, 44.8]	.750
Cost oracle (quality-blind)	.525 [.375, .675]	.000	51.4 [48.4, 54.4]	1
PAWS / DeepSeek (20 source pairs, 40 views)				
Replay argmax	.500 [.325, .675]	.000	16.2 [14.9, 17.4]	.943
Winner-only check	0	.500	8.0 [5.2, 10.8]	.464
Complete audit	0	.275	11.2 [8.2, 14.2]	.656
Cost oracle (quality-blind)	.525 [.375, .675]	.000	17.2 [15.9, 18.3]	1
OASST / Qwen (29 source pairs, 58 views)				
Replay argmax	.448 [.328, .569]	.000	47.2 [43.4, 51.0]	.894
Winner-only check	0	.448	27.7 [21.1, 34.4]	.523
Complete audit	0	.138	41.6 [36.7, 46.8]	.788
Cost oracle (quality-blind)	.448 [.345, .552]	.000	52.8 [49.7, 56.2]	1
OASST / DeepSeek (29 source pairs, 58 views)				
Replay argmax	.603 [.448, .759]	.000	18.9 [17.3, 20.5]	.906
Winner-only check	0	.603	7.9 [4.8, 11.0]	.377
Complete audit	0	.259	13.0 [10.6, 15.4]	.626
Cost oracle (quality-blind)	.586 [.448, .724]	.000	20.8 [19.6, 22.1]	1
OASST / GPT-OSS (29 source pairs, 58 views)				
Replay argmax	.534 [.431, .638]	.000	24.8 [23.4, 26.1]	.954
Winner-only check	0	.534	11.2 [8.4, 14.3]	.432
Complete audit	0	.138	20.1 [17.4, 22.7]	.775
Cost oracle (quality-blind)	.483 [.362, .603]	.000	26.0 [24.7, 27.2]	1

Table 29: Budget sensitivity. Per cohort: replay-argmax harm rate, complete-audit abstain rate, and the complete audit's retained fraction of the quality-blind oracle saving, as the declared budget ε varies. These are fixed-continuation NLL thresholds, not budgets calibrated to downstream task quality; "harm" denotes an observed NLL-budget violation.

	$\varepsilon = 0$			$\varepsilon = 0.01$			$\varepsilon = 0.05$		
Cohort	Harm	Abstain	Retained	Harm	Abstain	Retained	Harm	Abstain	Retained
PAWS/Qwen	.500	.150	.750	.425	.100	.834	.100	.000	.989
PAWS/DeepSeek	.500	.275	.656	.300	.025	.876	.000	.000	1
OASST/Qwen	.448	.138	.788	.310	.103	.852	.052	.000	.986
OASST/DeepSeek	.603	.259	.626	.414	.069	.844	.000	.000	.993
OASST/GPT-OSS	.534	.138	.775	.500	.086	.830	.224	.052	.924

Local reconstruction. `python quality_endpoint_20260918/verify_quality_endpoint.py` recomputes the point estimates, pair-bootstrap intervals and budget sweep from the five local matrices, checks their archived dataset/protocol hashes, and rejects missing or nonfinite candidate values. It runs entirely on the stored matrices. The original September 18 analysis and outputs are retained for provenance.

Table 30: Retrospective NLL-screening attribution at $\varepsilon = 0$. All savings are mean logical loads per view. Δ_{full} compares complete audit with winner-only screening (descriptive pair-bootstrap 95% intervals). Δ_{rank} changes replay to executed-cost ranking within the same fully checked feasible set. Recovered counts views where the rejected winner is replaced by a feasible non-null candidate. No candidate-acquisition cost is deducted.

Cohort	Δ_{full} [95% interval]	Δ_{rank}	Share (%)	Recovered
PAWS/Qwen	13.500 [8.100, 19.075]	0.850	6.3	14/40
PAWS/DeepSeek	3.300 [1.625, 5.050]	0.550	16.7	9/40
OASST/Qwen	13.983 [9.707, 18.483]	1.672	12.0	18/58
OASST/DeepSeek	5.172 [3.448, 6.914]	0.638	12.3	20/58
OASST/GPT-OSS	8.897 [6.103, 11.879]	0.879	9.9	23/58

H AUDIT ACQUISITION COST AND ORDINARY-REFERENCE COMPARISONS

The final two-corpus comparison appears in Table 34; the full ordinary terminal-KV comparison is in Table 46. The studies below document the development comparisons, endpoint changes and controls underlying them.

Development comparisons. Retrospective strong-baseline accounting uses complete candidate matrices, separately from the censored first-witness audit. On 29 pairs per family, gap/replay uses 111/118/142 non-null actions in Qwen/DeepSeek/GPT-OSS, compared with 114/130/154 for original-view-first replay, 125.83/132.50/148.75 in expectation under all 72 random view/challenger schedules, and 174 for complete sweeps, on previously evaluated records. A PAWS source-grouped cross-fitted scenario-KNN scheduler needs 757 versus 743 development actions and 375 versus 371 retrospective OASST actions and fails to improve on the baseline. Earlier GPU-minute estimates use single-action timing; the following studies measure the whole pipeline.

Direct whole-pipeline cost and lazy acquisition. The two-capture timing study freezes 24 new OASST1 training user-successor pairs, excluding historical trees and normalized texts. These are natural requests, not an assumption of paraphrase equivalence. Three families execute four independent policies twice, with rotated arm order, GPU-synchronized pipeline timing, immutable prefix KV/LRU cloning, and matched native-adapter and prefix-reuse checks. All completed action costs, replay values, and full-sweep failure indicators agree across policies and repeats. Model loading is accounted for separately.

Figure 3 makes the original/lazy policy explicit. Logical-load accounting determines the audit indicator; wall time measures the cost of obtaining it. Gap/eager instead pays for both captures before ordering views; random/lazy freezes random view and challenger priorities before observing action outcomes.

The original median-pair gate requires a positive descriptive interval lower endpoint in at least two families. Only Qwen passes, so the gate fails despite positive aggregate savings. A stronger-baseline extension allows ordinary original-view-first and random policies to acquire the second behavior trace only if the first has no executed witness. It retains the same complete-sweep target and independently reruns all arms on the same 24 pairs. Its aggregate endpoint was declared after early two-capture rows were visible but before the extension ran. It reuses the sample; the original median endpoint remains failed.

Original/lazy lowers aggregate cost in all three families, but gap/eager does not demonstrate superiority over either simple lazy baseline. The cost mechanism includes avoided native captures, not merely fewer action executions. For example, GPT-OSS original/lazy executes 236 rather than gap/eager's 224 actions but takes 195.84 rather than 209.56 seconds. Measured-section accounting attributes 18.36 seconds of its 38.55-second saving versus exhaustive to capture/replay and 20.18 seconds to action execution; this is an accounting decomposition, not a factorial causal effect. Savings concentrate in witness-containing pairs; no-witness pairs still require both captures and all six non-null actions. Thus the evidence supports cheaper rejection, not cheaper certification of valid labels. Intervals resample whole source pairs with both repeats kept together and are descriptive, not finite-sample performance guarantees.

Simple lazy first-witness protocol (original-view-first). Input: two requests, a fixed finite action set including no-op 0, a common continuation, and a matched deterministic execution/accounting contract.

1. Visit the original request, then its paired request only if needed. For the current request, capture the native trace and immutable prefix state; record its exact no-op cost $Q[0]$ and compute replay costs $r(a)$.
2. Set $\widehat{a} = \arg\min_a r(a)$ using the frozen canonical tie rule. If $\widehat{a} \neq 0$, execute it from a clone of the prefix state and cache its exact cost $Q[\widehat{a}]$.
3. Visit challengers $a \neq \widehat{a}$ in frozen replay-cost order. If $Q[a]$ is unknown, pay for its matched execution from a fresh prefix clone. After each challenger, return $X = 1$ if $Q[a] < Q[\widehat{a}]$.
4. If no challenger wins, continue to the paired request. Return $X = 0$ only when both requests' challenger sets have been exhausted.

Output: the exact complete-sweep pair-failure indicator X and the paid query ledger. Unacquired traces/actions remain unknown, not negative labels.

Figure 3: First-witness stopping with lazy acquisition, as measured in the direct-cost study. It preserves the full audit's pair-failure indicator while omitting queries after a witness. The timing studies evaluate every source pair. Using these indicators for between-pair stopping requires the sampling conditions in Appendix A.4.

Table 31: Two-capture audit pipelines on 24 new source pairs, two repeats. Gap/replay savings versus exhaustive are percentages. Brackets summarize descriptive source-bootstrap uncertainty in the median, not the total. The original median-pair promotion gate is retained.

Family	Median pair saving [95%]	Total saving	Witness pairs
Qwen	$+0.34\,[+0.05, +20.99]$	$+10.51$	11/24
DeepSeek	$+0.03\,[-0.03, +0.15]$	$+5.02$	5/24
GPT-OSS	$+0.10\,[-0.01, +23.83]$	$+10.59$	10/24

Frozen new-pair total-cost confirmation. The new-pair confirmation freezes the same four policies before executing 48 new OASST training trees, with two repeats and no sample extension. Sampling excludes 304 prior trees and 14,388 normalized text hashes from 63 historical JSON manifests; an additional all-string scan finds zero exact normalized-text/tree overlap. Exclusions use exact text and source identifiers. All 288 pair-repeat rows pass adapter/prefix parity, independently recomputed witness labels, shared action-cost equality and timing-component checks.

The frozen primary method is original/lazy. At least two complete families must have total saving at least 5%, a positive nominal bootstrap lower bound at percentile $0.05/3$, and positive saving in both repeats. All three pass (Table 36); 100,000 resamples of source pairs keep repetitions and policies paired. These are approximate bootstrap bounds for the total-cost endpoint; the earlier median-pair criterion remains failed.

Witness-containing pairs number 38/25/19 of 48 in Qwen/DeepSeek/GPT-OSS. Original/lazy saves 298.91/98.14/68.39 seconds per measured two-repeat batch; including one common model load retains 28.51/15.32/14.12% savings. Gap/eager versus original/lazy is $-15.97/-3.58/-5.09\%$ in total-cost saving. With the same policy on both samples, differing witness prevalence helps explain the different savings. No-witness pairs require full evaluation.

Matched-order stopping-by-capture attribution. The attribution experiment freezes a 2×2 intervention on the reused 24 source pairs with two repeats: full versus first-witness-stopped evaluation, crossed with eager versus lazy second-view capture. Every arm starts with the original view and uses the same replay-selected/challenger order. Full arms execute the same six paid actions and two captures; stopped arms execute the same prefix of the action sequence. Each arm independently captures and clones immutable prefix state; no outcomes or captures are shared between arms. A balanced cyclic arm order controls fixed position effects. All 144 pair-repeat rows pass independently reconstructed query, cost, witness and timing checks.

The two primary conditional effects use 100,000 paired source-bootstrap draws and nominal lower percentiles $0.05/6$, with no overall promotion gate. At eager capture, first-witness stopping saves 44.47/10.10/20.25 seconds in Qwen/DeepSeek/GPT-OSS, with adjusted lower bounds 17.26/0.25/7.02 seconds. At the same stopped action sequence, lazy capture saves 32.49/6.27/18.48 seconds, but

Table 32: Lazy acquisition on the same source pairs, with independent executions. Aggregate audit-cost savings (%) and descriptive source-bootstrap intervals. Positive values favor the first named strategy. This is evaluation cost, not serving latency; the extension is not a new independent sample.

Family	Original/lazy vs exhaustive	Gap/eager vs original/lazy
Qwen	$+14.86\,[+6.56, +24.13]$	$-5.00\,[-14.41, +2.44]$
DeepSeek	$+5.38\,[+0.64, +11.73]$	$-0.57\,[-6.85, +4.68]$
GPT-OSS	$+16.45\,[+6.52, +27.97]$	$-7.01\,[-20.08, +2.38]$

Table 33: Whole-audit costs for all four independently executed policies on the same 24 pairs, two repeats. Savings and brackets are percentages and descriptive source-bootstrap intervals versus exhaustive. Capture counts include native tracing and replay; actions use cloned immutable prefix state. Loading is excluded from these totals and listed separately below.

Family	Policy	Time (s)	Captures	Actions	Saving [95%]
Qwen	Exhaustive	513.25	96	288	—
Qwen	Gap/eager	458.85	96	224	$+10.60\,[+5.50, +16.13]$
Qwen	Original/lazy	436.98	84	236	$+14.86\,[+6.56, +24.13]$
Qwen	Random/lazy	423.64	82	228	$+17.46\,[+8.54, +27.02]$
DeepSeek	Exhaustive	309.27	96	288	—
DeepSeek	Gap/eager	294.29	96	256	$+4.84\,[+1.01, +9.44]$
DeepSeek	Original/lazy	292.64	92	266	$+5.38\,[+0.64, +11.73]$
DeepSeek	Random/lazy	291.84	91	268	$+5.63\,[+0.94, +11.24]$
GPT-OSS	Exhaustive	234.39	96	288	—
GPT-OSS	Gap/eager	209.56	96	224	$+10.59\,[+5.23, +16.64]$
GPT-OSS	Original/lazy	195.84	84	236	$+16.45\,[+6.52, +27.97]$
GPT-OSS	Random/lazy	201.84	83	236	$+13.89\,[+6.52, +22.10]$

Model loading: Qwen 22.91s, DeepSeek 12.76s, GPT-OSS 7.01s. Gap/eager versus random/lazy: Qwen $-8.31\,[-19.16, +0.29]$; DeepSeek $-0.84\,[-3.89, +1.34]$; GPT-OSS $-3.83\,[-8.76, +0.07]$. These contrasts do not measure serving speed.

adjusted lower bounds 7.82/-0.081/3.10 support this component only in Qwen and GPT-OSS. The DeepSeek unadjusted 95% interval is barely positive and does not override its failed primary rule. Both repeats have positive point effects for both components in every family.

The full/eager-minus-full/lazy negative control is 0.104/0.091/0.055 seconds; all descriptive intervals span zero, which is not an equivalence certificate. The secondary combined savings are 14.85/5.32/16.48%. Saved seconds add, but component percentages have different denominators and do not. A post-hoc diagnostic finds only 2/24 DeepSeek sources permit capture omission, compared with 6/24 in GPT-OSS; these two sources account for 5.77 of 6.27 seconds. The concentration explains the uncertain component estimate: capture omission is available only after executing a witness. The experiment measures conditional audit-cost effects on the reused sources.

Independent-corpus confirmation with an unchanged protocol. The independent-corpus confirmation freezes the same four policies, three checkpoints/capacities, fixed continuation, two repeats and total-cost decision rule before any new action outcomes. The source is Anthropic's HH-RLHF helpful-base test split (Bai et al., 2022), pinned at revision `09be8c5bbc57cb3887f3a9732ad6aa7ec602a1fa`. We remove the final assistant response and require chosen/rejected records to share their complete preceding prompt. Adjacent human turns supply two standalone requests, not a paraphrase pair or a concatenated chat history. Neither preference outcomes nor assistant responses are used as model targets. The existing length rule and deterministic hash ordering select 48 distinct normalized first-turn roots and 96 distinct texts. An all-string exclusion scan over 66 historical data JSON files covers 127,124 normalized hashes; no selected text overlaps exactly. This is not semantic or pretraining deduplication or proof of population independence. Source-level inference uses the selected root pair as its resampling unit.

All 288 pair-repeat rows pass frozen adapter/prefix checks, shared action-cost and repeat parity, and independent reconstruction of each policy's actual query sequence. All three family gates pass without sample extension, endpoint changes or family removal (Tables 39 and 40). As in the OASST confirmation, the 100,000-draw source bootstrap keeps repeats and policies paired.

Table 34: Original/lazy versus exhaustive: same-corpus new-pair confirmation (OASST) and external-corpus replication (HH-RLHF helpful-base), each with 48 source pairs and two repeats. Entries are total audit-time savings (%) with descriptive 95% source-bootstrap intervals and nominal adjusted lower bounds (LB, percentile $0.05/3$ within each study). These are not simultaneous bounds over both corpora or exact risk certificates.

	OASST: new pairs		HH-RLHF: new corpus	
Family	Saving [95%]	LB	Saving [95%]	LB
Qwen	$+29.13\,[+21.31, +37.12]$	$+20.65$	$+23.86\,[+16.15, +31.95]$	$+15.56$
DeepSeek	$+15.64\,[+9.31, +22.51]$	$+8.84$	$+21.42\,[+14.22, +29.00]$	$+13.63$
GPT-OSS	$+14.34\,[+8.11, +21.18]$	$+7.63$	$+12.76\,[+6.59, +19.59]$	$+6.15$

Passing families: OASST 3/3; HH-RLHF 3/3. Each study requires at least two, with at least 5% saving, positive LB, and positive saving in both repeats, plus integrity in all three.

Table 35: All independently executed policies on the 48-pair confirmation. Time includes capture, replay, prefix cloning, action execution and Python overhead, but excludes model loading. Savings are versus exhaustive; intervals are descriptive source-bootstrap summaries. The action target is exact logical-load regret, not measured latency or task quality.

Family	Policy	Time (s)	Captures	Actions	Saving [95%]
Qwen	Exhaustive	1026.31	192	576	—
Qwen	Gap/eager	843.55	192	358	$+17.81\,[+13.69, +21.99]$
Qwen	Original/lazy	727.39	144	366	$+29.13\,[+21.31, +37.12]$
Qwen	Random/lazy	765.24	148	399	$+25.44\,[+18.98, +32.11]$
DeepSeek	Exhaustive	627.55	192	576	—
DeepSeek	Gap/eager	548.35	192	410	$+12.62\,[+8.62, +16.77]$
DeepSeek	Original/lazy	529.40	166	456	$+15.64\,[+9.31, +22.51]$
DeepSeek	Random/lazy	534.97	167	470	$+14.75\,[+9.59, +20.25]$
GPT-OSS	Exhaustive	476.91	192	576	—
GPT-OSS	Gap/eager	429.32	192	454	$+9.98\,[+6.05, +14.18]$
GPT-OSS	Original/lazy	408.51	168	472	$+14.34\,[+8.11, +21.18]$
GPT-OSS	Random/lazy	404.45	167	476	$+15.19\,[+9.14, +21.70]$

Nominal adjusted lower percentiles address three primary contrasts WITHIN each study, not a joint six-contrast guarantee across the two corpora. Bootstrap inference remains approximate, separate from the anytime-valid invalidity test.

Witness-containing source pairs number 30/27/18 of 48 in Qwen/DeepSeek/GPT-OSS. Total savings are 23.86/21.42/12.76%, whereas median pair-repeat savings are 9.04/7.44/0.09%; including one common model load retains 23.27/20.97/12.54%. The result extends whole-audit cost evidence to another curated corpus under the same runtime and action class. No-witness pairs remain exhaustive. All policies and negative comparisons are reported, and the earlier failed median-pair gate is unchanged. Gap/eager's total-time contrasts against original/lazy are $-7.10/-8.61/-0.32\%$, and against random/lazy $-2.21/-4.91/-0.33\%$. DeepSeek's descriptive intervals exclude zero in the unfavorable direction for both comparisons; those of Qwen and GPT-OSS span zero. Thus the external corpus does not establish an improved gap-ordering algorithm.

I Exact action evaluation with ordinary computation reuse

Given the runtime and the dependencies changed by an action, exact queries can reuse unaffected computation and execute the affected suffix. The following conditions specify the state that this reuse must preserve.

Table 36: New-pair confirmation of original/lazy versus exhaustive on 48 previously excluded OASST trees, two repeats. Savings are total audit-time percentages; brackets are descriptive 95% source-bootstrap intervals. The adjusted lower bound uses percentile $0.05/3$ for the three primary contrasts; this bootstrap approximation is not an exact risk certificate. The frozen gate additionally requires at least 5% point saving and positive saving in both repeats, in at least two complete model families.

Family	Total saving [95%]	Adjusted lower	Repeat1/2	Gate
Qwen	$+29.13\,[+21.31, +37.12]$	$+20.65$	$+29.14/+29.11$	Pass
DeepSeek	$+15.64\,[+9.31, +22.51]$	$+8.84$	$+15.57/+15.71$	Pass
GPT-OSS	$+14.34\,[+8.11, +21.18]$	$+7.63$	$+14.34/+14.34$	Pass

Table 37: Matched-order stopping-by-capture ablation on the reused 24-source batch, two repeats. All effects are seconds saved, with descriptive 95% source-bootstrap intervals. Primary conditional contrasts are full/eager minus stop/eager and stop/eager minus stop/lazy; their nominal adjusted lower bounds use percentile $0.05/6$. Schedule control is full/eager minus full/lazy; interaction subtracts this control from the lazy-given-stopping contrast. Combined is full/eager minus stop/lazy. Seconds are additive, unlike savings percentages with different denominators. These are audit costs, not serving gains; approximate bootstrap bounds are not exact risk certificates.

Family	Contrast	Saving (s) [95%]	Adj. lower	Repeat1/2
Qwen	Stop (eager)	$+44.47\,[+22.21, +69.98]$	$+17.26$	$+22.30/+22.17$
Qwen	Lazy (stopped)	$+32.49\,[+10.51, +57.37]$	$+7.82$	$+16.11/+16.39$
Qwen	Schedule control	$+0.10\,[-0.13, +0.35]$	—	$+0.03/+0.07$
Qwen	Interaction	$+32.39\,[+10.33, +57.31]$	—	$+16.07/+16.31$
Qwen	Combined	$+76.97\,[+33.21, +125.73]$	—	$+38.41/+38.56$
DeepSeek	Stop (eager)	$+10.10\,[+1.23, +21.59]$	$+0.25$	$+4.85/+5.25$
DeepSeek	Lazy (stopped)	$+6.27\,[+0.03, +15.00]$	-0.08	$+3.48/+2.80$
DeepSeek	Schedule control	$+0.09\,[-0.22, +0.53]$	—	$+0.19/-0.10$
DeepSeek	Interaction	$+6.18\,[-0.01, +14.87]$	—	$+3.29/+2.89$
DeepSeek	Combined	$+16.37\,[+1.85, +36.34]$	—	$+8.33/+8.05$
GPT-OSS	Stop (eager)	$+20.25\,[+9.32, +32.66]$	$+7.02$	$+10.14/+10.11$
GPT-OSS	Lazy (stopped)	$+18.48\,[+5.03, +36.30]$	$+3.10$	$+9.26/+9.22$
GPT-OSS	Schedule control	$+0.06\,[-0.02, +0.14]$	—	$+0.05/+0.00$
GPT-OSS	Interaction	$+18.43\,[+4.97, +36.23]$	—	$+9.21/+9.21$
GPT-OSS	Combined	$+38.73\,[+14.83, +66.87]$	—	$+19.41/+19.32$

Proposition 6 (Conditional exact substitution and query-path invariance)**.** *Fix tokens and deterministic numerical transitions, with actions confined to decoder layers at or after ℓ. Suppose initial request states agree, actions cannot write upstream state, and substituted upstream outputs restore complete attention KV, positional/mask metadata and ordered logical accounting. Reusing native computation strictly before ℓ and executing the affected suffix returns the same action outputs and logical values as full execution. An acquisition policy depending only on contracted summaries and completed labels, with coupled randomness, then has the same query path and stopping decision under either implementation.*

Proof. Induct over fixed tokens and then decoder layers. Before ℓ, unchanged input, prior KV and numerical transitions imply equal outputs and state updates. Restoring that complete state gives the acted suffix the same inputs as full action execution. Ordered logical accesses from equal cache states give equal costs. Inducting over acquisition operations then gives equal observed histories, next queries and strict challenger decisions. A policy branching on elapsed time is excluded. These are sufficient conditions, not a characterization of minimal state; equivalent transition semantics do not require literal copying of every object.

For captured views V and executed queries Q, an equal-block accounting baseline already sharing request-prefix KV uses $L[\sum_{v\in V}(P_v + T_v) + \sum_{q\in Q} T_{v(q)}]$ blocks; reuse removes $\sum_{q\in Q} \ell_q T_{v(q)}$ blocks. This is a path-specific work prediction, not a wall-time bound: capture, cloning, tape storage and replay bookkeeping remain paid. Autoregressive token divergence breaks the common-input premise. The implementations cover the inspected Qwen DynamicLayer and DeepSeek legacy-to-DynamicLayer paths; unsupported offloaded/sliding caches fail rather than silently reuse state.

Table 38: Independent whole-pipeline costs in the matched-order 2x2 ablation. Both full arms execute the same six paid actions per pair; both stopped arms execute the same prefix of that sequence. Eager arms pay for both captures. The lazy stopped arm omits the second capture only when the first has a witness. Model loading is excluded. All no-witness pairs are exhaustive.

Family	Arm	Time (s)	Captures	Actions
Qwen	full/eager	518.27	96	288
Qwen	stop/eager	473.80	96	236
Qwen	full/lazy	518.17	96	288
Qwen	stop/lazy	441.31	84	236
DeepSeek	full/eager	307.80	96	288
DeepSeek	stop/eager	297.70	96	266
DeepSeek	full/lazy	307.71	96	288
DeepSeek	stop/lazy	291.43	92	266
GPT-OSS	full/eager	235.08	96	288
GPT-OSS	stop/eager	214.83	96	236
GPT-OSS	full/lazy	235.02	96	288
GPT-OSS	stop/lazy	196.35	84	236

Table 39: External-corpus audit-cost confirmation on 48 distinct-root HH-RLHF helpful-base user-successor pairs, two repeats. These are standalone related requests, not paraphrases or preference-learning examples. Original/lazy is compared with exhaustive evaluation under the unchanged OASST protocol. Savings are total-time percentages; brackets give descriptive 95% source-bootstrap intervals. The nominal adjusted lower uses percentile $0.05/3$. A family passes with at least 5% total saving, positive lower, and positive saving in both repeats; replication requires at least two passing families and complete integrity in all three. Approximate bootstrap inference is not an exact risk certificate.

Family	Total saving [95%]	Adjusted lower	Repeat1/2	Gate
Qwen	$+23.86\,[+16.15, +31.95]$	$+15.56$	$+23.81/+23.91$	Pass
DeepSeek	$+21.42\,[+14.22, +29.00]$	$+13.63$	$+21.46/+21.39$	Pass
GPT-OSS	$+12.76\,[+6.59, +19.59]$	$+6.15$	$+12.79/+12.74$	Pass

Witness-containing source pairs: Qwen: 30/48, DeepSeek: 27/48, GPT-OSS: 18/48.

Activation caching and suffix-entry execution are established capabilities, including TransformerLens's `start_at_layer` interface (TransformerLens Contributors, 2026). The measured question is whether a state-consistent exact evaluator reduces complete action-evaluation cost in this audited setting.

New-request confirmation and explicit validation amendment. After four development sources, the reuse study froze 32 unused HH-RLHF requests from distinct normalized first-turn roots, excluding historical exact text and every root from the independent-corpus audit. These are standalone requests, not another corpus or paraphrases. Checkpoint, capacity 16, prefix cap 96, fixed continuation and three actions are unchanged; each arm pays capture and copies. The primary rule requires at least 5% total-time saving, a positive nominal one-sided 95% lower, and positive saving in both repeats. The 100,000-draw source bootstrap keeps repeats paired; inference is approximate, not distribution-free. Table 41 reports all results.

During the original run, comparator review found that `torch.equal` does not distinguish signed zero or require matching dtype. It therefore did not establish the protocol's stronger word "bitwise." We preserve the original implementation and timing and explicitly amend validation with a follow-up on the same requests, two repeats, unchanged measured computation, and a separate frozen check of tensor dtype, shape and logical-value bytes. All 192 paired actions in that follow-up pass, and logical labels match the original run. Byte parity is established for the repeated executions; the original timings and endpoints remain unchanged. The combined amended criteria pass.

The first acted depth 0 is a zero-skipped-work control, not a speedup. Later cuts save computation while native tape construction adds cost. These results concern exact full-action evaluation in one Qwen/HuggingFace/A800 environment. The separate fresh-pair study in Table 42 now confirms the increment within first-witness/lazy acquisition. It freezes 32 new roots and 64 distinct texts

Table 40: All independent policy executions on the HH-RLHF external corpus. Whole-pipeline time includes tokenization, capture, replay, cloning, action execution and Python overhead; model loading is separate. This is audit cost, not serving latency. No-witness pairs remain exhaustive. The corpus changes, but model checkpoints, runtime, hardware and action class do not.

Family	Policy	Time (s)	Captures	Actions	Saving [95%]
Qwen	Exhaustive	877.03	192	576	—
Qwen	Gap/eager	715.22	192	382	$+18.45\,[+13.45, +23.55]$
Qwen	Original/lazy	667.79	154	412	$+23.86\,[+16.15, +31.95]$
Qwen	Random/lazy	699.77	160	439	$+20.21\,[+14.03, +26.76]$
DeepSeek	Exhaustive	534.20	192	576	—
DeepSeek	Gap/eager	455.91	192	414	$+14.66\,[+10.27, +19.13]$
DeepSeek	Original/lazy	419.76	158	426	$+21.42\,[+14.22, +29.00]$
DeepSeek	Random/lazy	434.59	162	451	$+18.65\,[+13.14, +24.36]$
GPT-OSS	Exhaustive	403.12	192	576	—
GPT-OSS	Gap/eager	352.77	192	446	$+12.49\,[+7.78, +17.51]$
GPT-OSS	Original/lazy	351.66	174	488	$+12.76\,[+6.59, +19.59]$
GPT-OSS	Random/lazy	351.62	171	491	$+12.77\,[+7.53, +18.50]$

Table 41: Exact full-action evaluation on 32 newly excluded HH-RLHF requests, Qwen/HuggingFace/A800, two repeats. The full baseline already reuses request-prefix KV; the other arm additionally reuses invariant continuation-layer computation. Whole-pipeline time includes independent native capture, tape construction, copies and all three action evaluations. Savings are percentages; the original row gives a descriptive source-bootstrap 95% interval. The second row repeats the SAME requests with a stricter validation check, not a second independent confirmation. Its timing is secondary and cannot rescue the original gate.

Study	Prefix reuse (s)	Ancestor reuse (s)	Saving [95%]	Repeat1/2
Original timing	309.00	270.84	12.35 [11.50, 13.22]	12.44/12.26
Strict-byte follow-up	308.57	270.46	12.35	12.43/12.27

Original nominal one-sided lower: 11.64%; archive-inclusive saving: 11.71%. Original numeric/timing criteria plus strict follow-up validation: Pass. No first-witness/lazy incremental or serving claim.

after excluding all 80 prior HH roots and historical exact texts. Both repeats and all methods are resampled jointly within source roots using 100,000 draws; the sole primary contrast and its gate are fixed before pretrained execution. All 64 rows pass payload-coverage, byte-parity, independent policy reconstruction and actual MoE-call guards. Identical query paths isolate computation cost; they do not reduce required label information. Neither study establishes a cheap population-validity certificate or an audit-caused serving gain.

Ordinary suffix-entry reference. A separately frozen comparison retains the four measured arms and adds direct entry at the first affected decoder layer. It uses four historical development pairs, not the fresh confirmation requests. Method order is reversed across repeats to equalize mean ordinal position. The full and stopped policies are reconstructed from complete action values; direct entry preserves all returned payload bytes and the stopped query path. Decoder-module counters distinguish genuinely bypassing upstream calls from returning cached values inside those calls. Both methods execute the same affected MoE operations.

The direct implementation is descriptively 0.17% faster than hook reuse, with no significance or equivalence inference from four sources. It has slightly more paid native metadata and slightly lower action execution time. Both implementations preserve the state needed for exact suffix execution. Timing boundaries differ from the confirmation, so compare methods only within each measured study.

Second-architecture fresh-source confirmation. The DeepSeek evaluator uses a separate adapter for one dense and 26 routed layers, shared experts, fixed-scaling YaRN, eager expanded K/V, legacy cache conversion and FP32 output logits. Native before-token masks/positions, upstream KV, hidden boundaries and ordered logical records are restored. Actual dense-MLP and routed-MoE calls are counted separately. Untouched native execution is checked before the timed arms; model loading and this preflight are reported separately, not included in the timing comparison.

Table 42: Exact-query computation reuse within a matched audit on 32 fresh HH-RLHF source pairs, Qwen/HuggingFace/A800, two repeats. Each arm pays its own native capture, tape, copies, actual queries and verification-payload collection. Prefix KV is already reused in the baseline. The primary comparison is the last two rows, with identical query paths and strict native/action output parity. These are total audit seconds, not serving latency.

Evaluator	Total (s)	Captures	Queries
All queries, prefix KV	602.65	128	384
All queries, ancestor reuse	527.74	128	384
Stopped/lazy, prefix KV	456.73	104	276
Stopped/lazy, ancestor reuse	409.37	104	276

Primary saving: 10.37% [9.22, 11.35]; nominal one-sided lower: 9.42%. Repeat1/2: 10.37/10.37%; archive-inclusive: 9.89%. Frozen gate: Pass. Source-bootstrap intervals are approximate, not risk certificates.

Table 43: Ordinary suffix-entry reference on four historical development pairs, two repeats. Five arms independently execute with identical contracted labels. The outer timer includes wrapper/counter setup and teardown in every arm; these absolute times must not be pooled with the fresh-pair study's different timer boundary. This is descriptive development evidence, not independent confirmation, an equivalence test or a novel reuse algorithm.

Evaluator	Total (s)	Captures	Queries	Decoder calls
All queries, prefix KV	93.845	16	48	48864
All queries, hook reuse	83.098	16	48	48864
Stopped/lazy, prefix KV	78.215	14	36	40992
Stopped/lazy, hook reuse	71.712	14	36	40992
Stopped/lazy, direct entry	71.593	14	36	35872

Against stopped/lazy prefix-KV: direct entry 8.47%; hook reuse 8.31%. Direct versus hook: 0.17%; repeat1/2: 0.02/0.31%. Archive-inclusive direct versus hook: 0.08%. All eight rows pass strict integrity.

A four-historical-pair development gate passes before selecting 32 new HH roots, excluding all 112 earlier HH roots and historical exact text. The 133-file freeze retains the unchanged evaluator, capacity 12, prefix cap 96, fixed continuation and action rule. Rotated method order is reversed across two repeats. The primary gate and 100,000-draw source-bootstrap procedure match the Qwen criterion, but are a separate study, not a joint confidence statement.

All 64 paired-repeat rows pass strict native/action output-byte coverage, policy reconstruction and repeated-label checks. Both stopped arms use 114 captures/322 queries. Both repeat savings and the nominal primary lower bound are positive; archive-inclusive saving is 12.91%. This confirms the audit-cost benefit of standard reuse in a second model architecture under the same serving runtime.

Strong ordinary baselines and scaling boundaries. Two already completed development studies delimit what the fresh-source reuse results establish (Table 45). Neither changes the frozen implementations or endpoints in Tables 42 and 44. The first compares full execution, per-token native-state snapshots, and an ordinary terminal-KV representation on two historical DeepSeek pairs. The latter stores one final append-only native KV and reconstructs the needed prefixes, avoiding duplicate growing snapshots while still paying prefix-copy costs. Six controlled prefix/continuation lengths use identical tiled/truncated tokens across arms, not natural long-context examples. Terminal storage reduces tape tensors by 86.69–97.73%, but whole-audit time improves only 0.60–3.49% relative to snapshot reuse. At $P = 96, T = 64$, peak allocated GPU memory falls from 32.87 to 29.95 GiB; allocator-reserved memory is unchanged. Tape reduction is not whole-model memory reduction.

The second study compares a state-aware dynamic execution frontier to the stronger ordinary terminal-KV baseline on six historical DeepSeek pairs. The frontier skips later computation only while unchanged execution inputs and operators are verified, tracking persistent KV effects. All stopped arms retain 24 captures and 70 action queries with exact labels. It saves 3.81% whole-audit time, below its original 5% advancement threshold (archive-inclusive 3.57%). Action execution saves 2.872 s, but additional capture costs 0.297 s; total net saving is 2.540 s. Native capture accounts for 63.32% of the baseline time. Frontier tape memory rises from 19.60 to 27.85 MiB. Thus cheaper suffix execution does not remove the dominant native-capture cost, and the development gate fails against terminal-KV reuse.

Table 44: Fresh-source DeepSeek exact-audit confirmation: 32 HH-RLHF natural user-successor pairs, two repeats and four independently executed arms. The unchanged development evaluator handles dense/shared experts, eager expanded KV, fixed-scaling YaRN and legacy cache conversion. Prefix KV is already reused by the baseline; capture, tape and copies are charged. This is a second architecture in the same research-audit/HuggingFace/A800 setting, not a second serving runtime.

Evaluator	Total (s)	Captures	Queries
All queries, prefix KV	380.04	128	384
All queries, direct reuse	330.26	128	384
Stopped/lazy, prefix KV	326.20	114	322
Stopped/lazy, direct reuse	282.33	114	322

Primary stopped/lazy saving: 13.45% [12.74, 14.17]; nominal one-sided lower: 12.86%. Repeat1/2: 13.37/13.52%; archive-inclusive: 12.91%. Frozen gate: Pass. Source-bootstrap intervals are approximate, not risk certificates or a joint guarantee with Qwen. These are audit costs, not serving latency.

Table 45: Historical strong-reference development, not fresh confirmation. Panel A uses two DeepSeek sources, two repeats, three independently paid full-query evaluators, and tiled/truncated tokens. All six length cells are shown. Panel B uses a separate six-source, two-repeat DeepSeek study with ordinary terminal-KV and dynamic-frontier evaluators; all six arms are shown. Seconds and percentages must not be pooled across panels or other studies.

A. Whole-audit time (s); tape tensor memory (MiB)

P/T	Full	Snapshot	Terminal	Saving	Tape S/T
32/9	29.50	26.39	26.09	1.13%	124.99/16.63
32/32	70.76	60.85	58.73	3.49%	580.98/28.79
32/64	129.01	107.41	104.51	2.70%	1541.95/45.70
96/9	55.83	52.75	52.44	0.60%	338.74/40.38
96/32	97.14	87.13	84.92	2.53%	1340.98/52.55
96/64	154.87	133.71	131.09	1.95%	3061.97/69.47

B. Matched evaluators within each query policy

Evaluator	Total (s)	Capture (s)	Actions (s)	Queries
All/prefix KV	78.19	41.59	36.59	72
All/terminal KV	67.74	42.26	25.41	72
All/frontier	65.17	42.55	22.51	72
Stopped/prefix KV	77.01	41.49	35.50	70
Stopped/terminal KV	66.73	42.25	24.40	70
Stopped/frontier	64.19	42.55	21.53	70

A: saving is terminal versus snapshot, not versus full execution. B: frontier versus stopped terminal saves 3.81%, below the original 5% advance gate; repeat savings 3.83/3.78%. All integrity checks pass. Neither panel supplies a confidence interval, free-generation result, candidate-count sweep, or new caching primitive.

These data provide controlled length/memory profiles and per-window action timings. The fresh-source comparison against ordinary first-witness/lazy capture plus direct terminal-KV entry is reported in Table 46; the ordinary implementation accounts for the reuse benefit.

Fresh complete ordinary-reference comparison. The frozen comparison independently charges capture, tapes, copies, accounting, and action execution. Qwen's snapshot returns ancestors through hooks; DeepSeek's already enters the suffix directly. The ordinary reference stores final append-only native KV and materializes its required prefixes before suffix entry: standard primitives in the research runtime, not a separately optimized production serving system. All 128 source-repeat rows pass complete native/action payload, independent query reconstruction and decoder/MLP/MoE work checks. Each Qwen arm uses 110 captures/296 queries; each DeepSeek arm uses the same 310 queries.

The 64 newly selected HH roots are disjoint across Qwen and DeepSeek, with 32 per family and exclusions against previous normalized texts and root IDs; there is no cross-family paired inference or semantic-deduplication claim. Before the GPU freeze, eight actual tiny-model CPU conditions (two architectures, FP32/BF16, two lengths, two repeats) pass full native/action payload and query/work checks. The first preflight's validator incorrectly expected DeepSeek snapshot reuse to invoke skipped

Table 46: Fresh strong ordinary-reference comparison: 32 excluded source pairs per family, two repeats, identical first-witness/lazy acquisition in all arms. Prefix, snapshot and ordinary terminal-KV/direct-entry times are whole-audit seconds. The primary saving is snapshot versus terminal, not versus prefix; brackets are approximate paired-source 95% bootstrap intervals.

Family	Prefix (s)	Snapshot (s)	Terminal (s)	Primary saving (%)
Qwen	478.96	429.23	420.17	-2.16 [-2.33, -1.98]
DeepSeek	302.41	263.21	258.64	-1.77 [-1.90, -1.63]

Table 47: All arms of the fresh strong-reference study. Capture includes paid tape construction; total additionally includes copies/accounting, equally collected verification payloads and wrapper overhead. Archival and independent checks are charged separately, not hidden in the speed ratio. Tape memory is not whole-model allocation.

Arm	Total (s)	Capture (s)	Action (s)	Archive (s)	Tape (MiB)	Queries
Qwen/Prefix	478.96	216.32	250.51	23.22	0.00	296
Qwen/Snapshot	429.23	226.20	190.65	23.25	39.84	296
Qwen/Terminal	420.17	221.81	185.02	23.22	6.76	296
DeepSeek/Prefix	302.41	139.82	154.82	13.81	0.00	310
DeepSeek/Snapshot	263.21	145.92	109.11	13.68	111.63	310
DeepSeek/Terminal	258.64	143.28	107.27	13.65	15.15	310

Qwen. Primary gate fails; repeat savings -2.08/-2.23%; archive-inclusive -2.05%. Verification-only exhaustive references cost 293.21 s plus 12.62 s archival; independent checks 24.31 s; model load/preflight 21.51/12.95 s. Overall research-process wall time is 1789.32 s (including serialization, collection and untimed housekeeping).

DeepSeek. Primary gate fails; repeat savings -1.80/-1.74%; archive-inclusive -1.69%. Verification-only exhaustive references cost 183.45 s plus 7.05 s archival; independent checks 14.08 s; model load/preflight 12.27/4.64 s. Overall research-process wall time is 1107.42 s (including serialization, collection and untimed housekeeping).

decoder modules; that adapter already uses direct entry. A recorded amendment corrects only this expected count and reruns all conditions before GPU outcomes, preserving the failed preflight.

One exhaustive native matrix is independently executed per fresh source and never supplied to the stopped policy. Within each of two repeats, six method orders rotate across sources with counts differing by at most one. Full logits/KV/routes/ordered logical state match native action execution; scalar values and query paths also repeat. Each family's 100,000-draw bootstrap resamples all methods and repeats together. The .025 primary lower quantile applies a two-family Bonferroni adjustment to an approximate one-sided bootstrap bound; it is not a distribution-free certificate. Both primary snapshot-over-terminal 5%/positive-lower/both-repeats gates fail. The full native verification matrix is separately charged, not a timed policy arm.

Bounded migration screen. A preceding CPU screen exercises the existing single-token restrictions: both Qwen and DeepSeek auditors reject both tested multi-token shapes (four refusals). Supported tiny-DeepSeek cache-representation conversion passes two repeats in each of FP32 and BF16. No new silently accepted wrong-label case is found, so the prespecified independent-confirmation trigger is not met. The screen confirms the existing refusal and compatibility boundaries.

I.1 Adapting the audit to a new intervention

Interface and backend obligations. The artifact's `audit_reference` separates the standard first-witness schedule from two model-specific callbacks: native capture and exact action execution. Declare an ordered action set, no-action baseline, cost orientation, source unit, token policy, initial-state/reset policy and runtime identity. A capture returns replay costs and a separately measured native baseline cost; an execution returns one action's exact value and matching source, capture, action and execution-contract identifiers. These identifiers guard against accidental mixing, not a falsely declared or incorrectly implemented runtime. The backend must reset candidate state independently and validate untouched native execution, full-versus-reused logits/KV/routes, ordered accounting, attention metadata and actual executed operations. Unsupported semantics must raise an error rather than return a default cost. State preservation is a sufficient contract, not a proof of globally minimal storage.

Three distinct outcomes. The reference explicitly distinguishes:

1. An acquired exact challenger strictly beats replay's choice. This is an invalidity witness, not certification of the best action or task quality.
2. Every declared action and view has been evaluated without such a witness. This is a finite-source negative, not population validity or equality of every replayed value.
3. A query budget expires before either conclusion. This is incomplete evidence, not an observed zero-failure source. Backend errors also cannot be converted into negative labels.

The paper's uncensored source-pair e-process uses the first two outcomes. Dropping incomplete sources or treating them as fully observed negatives changes that observation mechanism and needs a separate statistical analysis. A strict comparison of noisy timing observations likewise does not certify an expected latency advantage.

Validation scope. The extracted interface reproduces scalar labels and query paths from all archived records in the two-corpus acquisition and two-architecture reuse confirmations, with malformed-receipt and incomplete-budget tests. The original frozen evaluators retain their own executed parity and timing evidence; their measured speedups cannot be transferred to the extracted API without a new charged live test. The artifact guide specifies the callback contract, state checks and interpretation of each outcome.

I.2 Live contract checks and the limits of local exactness

Numerical dependency preservation. Proposition 6 is constructive but conditional: replace an operation only when its complete read state and numerical transition agree, and restore its output and state effects. Induction over the actual operation dependencies then preserves subsequent execution; real-arithmetic algebra alone does not. In a native ordered MoE reduction, write the computed weighted expert responses as $u_{t,e}(a_e)$ and the per-token accumulation as

$$s_{t,0} = 0, \qquad s_{t,j} = \mathrm{fl}\{s_{t,j-1} + u_{t,e_j}(a_{e_j})\}.$$

Preserving each response's input rows, row ordering, GEMM shape, parameter and weighting/cast semantics, and this rounded chain preserves its output bits. Independent token chains may be scheduled differently; their additions may not be reassociated. This argument requires the listed dependencies to be preserved; a runtime identifier alone does not establish that condition.

Runtime guard and its assumptions. The optional Torch reference guard binds a native *whole-block output* memo entry to observed input bytes/layout, registered parameters/buffers, inspectable source/configuration, selected runtime flags and payload integrity. It returns conditional match, changed dependency requiring re-execution, or unsupported scope/mode. It assumes an inspected deterministic, stateless backend without hidden relevant state, concurrent mutation or adversarial tampering. Matching these fields does not itself prove those assumptions. In particular, the guard does not automatically certify the separate stateful prefix/suffix or expert-response composition adapters. Requesting NLL from a block-output entry is unsupported: the remaining action-dependent suffix is still needed. Full byte hashing is charged at each check. The guard is a debugging tool; its measured overhead is reported below.

Frozen live conformance. Four historical texts (not independent quality data), 32 tokens each, supply new native batch inputs to Qwen layer 24 and DeepSeek layer 13. Per family, repeat, byte-identical input copy and separately captured fake-W8 expert 0 entries must pass with byte-identical fresh native outputs. Changed input, parameter bytes, layout, TF32 flag and payload must be refused; NLL scope, training mode and opaque instance-forward overrides must be unsupported. All 22 verdicts and all six accepted-output comparisons pass (Table 48). These are designed conformance cases, not 22 independent source observations or an estimate of universal error rate. Two compatibility amendments preserve the same frame and endpoints: typed integer configuration keys, then a harness diagnostic for inference parameters without version counters. Both failed preliminary runs are retained.

The Qwen TF32-flag change leaves native bytes unchanged but is still refused: the guard conservatively checks its declared contract, not numerical inequality as a classifier target. Layout comparison

Table 48: Frozen live-contract conformance, not a population or efficiency test. Each family has three accepted exact replays, five changed-dependency refusals and three unsupported cases. Times are medians over the three accepted cases, not repeated timing trials; replay includes the full live hash check.

Family	Cases passed	Exact accepts	Refusals	Guarded replay (s)	Native (s)
Qwen	11/11	3/3	8/8	1.1784	0.01836
DeepSeek	11/11	3/3	8/8	1.0808	0.00567

includes shape, so its refusal is not itself evidence of changed numerical values. Guarded replay costs over one second because full registered weight bytes are hashed; it is far slower than these native block calls. No check overhead is hidden in historical audit savings, and this experiment establishes no efficient validity certificate.

Strong-baseline and target-cost checks. Separate local Qwen checks fix layer 24, eight expert candidates and all 70 four-of-eight fake-W8A16 masks. Native-order response reuse preserves all tested output bytes; in a 16-historical-text follow-up, a conventional fused ordered cache beats our token-order implementation in all three repeats (61.46 versus 82.94 ms median for 70 configurations). The fused reference is implemented by us, not a reproduction of an external quantization pipeline. This fails the incremental-method gate despite a gain over the unfused cache.

A further native batch-16-by-128 execution captures its own layer inputs and runs the true downstream model for all 70 masks. Three prespecified full-stock versus suffix-entry anchors have byte-identical logits/NLL, and all 70 native suffix NLL vectors repeat exactly across three passes. A shared measured suffix ledger with independently checked local parity gives 42.498 s for ordinary caching and 42.401 s for fused caching, including prefix capture but excluding common loading. This is paired component accounting, not three independently timed end-to-end implementations. Eliminating the ordinary local evaluator entirely could save only 0.368% with this prefix/suffix workload fixed; actual fused saving is 0.228%, with first-use compilation separately costing 1.112 s. The local-only optimization branch therefore stops. A different local-error and full-NLL winner is a proxy/target distinction, not invalid replay: both objectives are evaluated under the same valid contract. No held-out quality or new serving benefit is inferred.

J Reproducibility entry points

The supplementary archive contains CPU checks and selected archived evidence. The supported commands below run from its extracted root. Tables are generated or checked against frozen JSON records. Manifests bind those records to the checked tables and manuscript, establishing file identity rather than independently validating the experiments.

CPU execution and archived evidence. The unified entry is `python reviewer_evidence_20260926.py --case all`. It executes the synthetic integer-load construction on CPU and reconstructs Figure 1, the complete winner-acceptance partition, the cache-state swap and confirmation labels, and the NLL matrices and attribution comparison. It also reconstructs HH regret units and query counts, the prospective audit's recorded query and confidence paths, and the natural-response budget scan. These latter checks read archived outputs. They require neither model weights nor a GPU; Table 49 distinguishes these checks from the materials needed to rerun the pretrained experiments.

The archive is a reduced evidence package, not the full experiment repository. Its scalar export preserves numeric observations, source/view identifiers and the saved acquisition order while omitting request text and infrastructure metadata. Source hashes and the field-extraction script document this projection.

The retrospective NLL-budget tables have an additional entry: `python quality_endpoint_20260918/verify_quality_endpoint.py`. It independently reconstructs the five frozen matrices' point estimates, cluster-bootstrap intervals and budget sweep from the fixed-token observations.

Table 49: Coverage of the supplementary archive. All listed entry points run from its extracted root.

Item	Included material and supported check
CPU theory execution	Synthetic inputs and executable witness construction; runs ordinary CPU operators.
Archived neural evidence	Selected candidate matrices, cache records and scalar exports; recomputes reported values, verdicts and recorded timing sums.
Source data	Selected source/view identifiers and protocol descriptions; not a complete input-text and preprocessing bundle for GPU reruns.
Model and environment	Architecture and execution settings in Appendix B; some archives retain version metadata. No model weights or complete per-run immutable model/tokenizer revision manifest.
Software	CPU/archive verifiers and dependencies used by them; `ENVIRONMENT.txt` records their tested environment, not the original GPU environment.
Pretrained GPU rerun	Full capture/evaluator pipeline and a locked GPU environment are not included. A rerun additionally needs the exact inputs, model/tokenizer revisions and per-run configurations.

A unified diagnostic report. The minimal entry reports the opening example's three verdicts together with the complete 40-view partition. Its HH and prospective sections name their separate cohorts and retain incomplete observations as incomplete. The unified entry includes the pinned cache example, its event-local state swap, all 96 confirmation labels, and the winner-only versus complete-candidate acceptance comparison. The underlying calculations use ordinary replay and paired-value checks. The report keeps cache and FP32 cohorts separate and preserves the absence of additional above-budget gain after route-only correction. Its three verdicts mean supported by the stated finite evidence, refuted by a checked counterexample, or inconclusive because required evidence is missing. Neither a finite pass nor artifact-hash agreement certifies a deployment population.

`LABEL_ACCEPTANCE_GUIDE.md` specifies a second, scalar-input mode. It accepts one request's explicit candidate order, integer logical-load savings, native denominator and regret budget. Missing executed candidates are not filled from replay; positive challengers can refute a claim, but incomplete negative evidence cannot accept it. This mode reports the execution contract as declared but unverified: inspecting scalar values cannot establish neural runtime correctness. Unsupported units and aggregate inputs are rejected; latency and task quality require their own evaluation rather than a relabeling of logical loads. Tests include missing challengers, exact-budget equality, tie-breaking, malformed inputs and refusal to overwrite existing receipts.

K Evidence index

Table 50 links each conclusion to its comparison unit and supporting experiment. Different rows use different cohorts and action sets. Numerical reconstruction and runtime validation are described separately in Section J.

L Expanded related work

Related and concurrent evaluation studies. The arXiv first-release dates are July 30, 2026 for Zhu (2026), August 8 for both Zhang (2026) and Shihab et al. (2026), and August 31 for Lee et al. (2026). Table 51 compares their evaluation targets and evidence. The earlier expert-importance audit (Engmann et al., 2026) appeared on June 9.

Zhang isolates fixed-event accounting and workload effects; Engmann et al. test population statistics against token-conditioned ablation losses; Zhu localizes numerical reduction forks and reconstructs persistent state. These studies overlap with our work in their numerical mechanisms and execution diagnostics. Our construction fixes the checkpoint, all contracted behavior observations, replay ordering and realizable opportunity while changing deployment order. In real models, we compare candidate values under replay and execution, then measure the loss at different thresholds and selection units.

Table 50: Evidence index. Comparisons use their own action sets, cohorts and decision units; rows are not a pooled experiment.

Question	Comparison	Evidence location
When is fixed-route replay exact?	Matched ordered accesses, initial state and accounting; unseen cache actions	Matched labels in Section 6.1; sufficient conditions in Appendix A.
Why does an intervention break replay?	Controlled-FP32 routing; first acted layer versus downstream layers	Intervention feedback in Section 6.2; action definitions in Appendix B.
Why can equal routes have different costs?	Event-local cache-state swap and complete confirmation labels	Delayed loads and capacity tests in Appendix C.
Does checking the winner validate the choice?	Complete controlled-FP32 candidate matrices	Exact values with better challengers and inexact values with optimal choices in Table 13.
When does error exceed budget?	Same-cohort threshold grid, aggregate choices and prospective finite-frame audit	Budget and decision-unit comparisons in Section 6.3.
What does an NLL check add?	Complete feasible sets versus winner-only checks	Candidate coverage and cost-ranking attribution in Appendix G.
What reduces audit cost?	Whole-pipeline timings and complete stopped/lazy terminal-KV reference	Ordinary stopping and reuse; negative custom-method comparisons in Appendix H.

Routing stability and intervention. Classical sparse MoE work studies learned routing and load balance (Shazeer et al., 2017; Fedus et al., 2022; Dai et al., 2022); recent work examines routing consistency, causal boundaries, and quantization-induced route changes (Park et al., 2026; Engmann et al., 2026; Gu, 2026). R3 (Ma et al., 2025), first released in October 2025, identifies discrepancies between rollout and training routers and uses recorded rollout masks while recomputing training-side logits and expert outputs. It reports improved RL stability and task performance, establishing numerical route mismatch as a practical concern before this study. R3 deliberately trains under fixed rollout routes. Our evaluator instead follows interventions that permit downstream rerouting, and tests the resulting candidate rankings and losses. Positive task-specific rerouting methods (Li et al., 2025; Yoon et al., 2026) use supervision unavailable to our cheap behavior-trace replay. CoR uses virtual ablation and counterfactual expert impact to reallocate a fixed activation budget for factuality (Hu et al., 2026). These methods acquire intervention-specific evidence through evaluation or supervision. Our question concerns the values that can be inferred before acquiring that evidence.

MoE serving. Expert caching, prefetch, mixed precision, and scheduling are explored by MoE-Infinity, HOBBIT, ExpertFlow, FATE, and pre-attention predictors (Xue et al., 2024; Tang et al., 2024; He et al., 2024; Fang et al., 2025; Zhu et al., 2025). Cache-conditional experts, ReMoE, and hardware-specific cache controllers (Skliar et al., 2025; Zhu et al., 2026; Suram, 2026) evaluate different actions, runtimes, or available state from our resident-aware substitution. PagedWeight dynamically quantizes MoE weights as KV demand changes, and WiSP treats resident experts and KV cache as competing working sets (Yang et al., 2026; Zhang et al., 2026). Both report results from direct deployment. Huang et al. (2026) formalize online/offline GPU assignment of experts with reconfiguration costs. These serving objectives include scheduling and transfer effects beyond the logical-load accounting used in our mechanism experiments.

Counterfactual identification and adaptive certification. Execution nondeterminism and counterfactual/selective learning motivate execution-matched labels and explicit risk (Yuan et al., 2025; Bottou et al., 2013; Swaminathan & Joachims, 2015; El-Yaniv & Wiener, 2010). Decision-dependent distributions and no-overlap partial identification are already established (Perdomo et al., 2020; Khan et al., 2024), as are time-uniform inference and constrained Pareto identification (Cook et al., 2024; Kone et al., 2025). Khan et al. obtain sharp no-overlap value bounds under Lipschitz smoothness. We use the general compatibility-set formulation to describe a specific missing transition in one MoE checkpoint: a previously observed expert receives a new input under the intervention. Conditional optimal transport can add cross-action structure (Lin et al., 2026); applying it here would require assumptions about those transitions and evidence that the assumptions hold.

Table 51: Closest-work comparison by evaluated experiment and decision target. Fixed-trace accounting, expert ablation, numerical conformance, and direct serving measurements supply different evidence. Our tests connect action-induced feedback to candidate selection and its loss-budget consequences. Zhang, Zhu and Opportunity are concurrent studies; first-release dates appear in the accompanying text.

Work	Experiment / established result	Evidence and deployment boundary	Additional question addressed here
R3 (Ma et al., 2025)	Rollout masks reused in training, with logits and expert outputs recomputed; RL stability and task gains	Deliberately route-constrained training; practical routing discrepancy predates this study	Does behavior evidence support values, rankings and budgets for an action whose natural suffix can reroute?
Zhang (2026)	Event-atomic versus sequential accounting on fixed routes reverses policy rankings; workload construction changes conclusions	Corrects own simulated evaluation; public-data negative control plus synthetic positive; oracle gaps do not imply causal gains	Our expert-output interventions change the subsequent trace, not just fixed-event accounting
Engmann et al. (2026)	Population expert statistics tested against token-level ablation losses across three families	Diagnostic failures and a narrow late-layer control signal; not an end-to-end pruning-and-recovery result	We test candidate selection and loss thresholds, not only proxy–ablation association
Zhu (2026)	Controlled expert reductions in DeepSeek-V4-Flash cause delayed generation forks; exact persistent-state reconstruction	Numerical mechanism and compatibility diagnostics on the evaluated states and schedules	Does identical contracted behavior evidence and realizable opportunity determine deployment order when action semantics are unobserved?
Opportunity (Shihab et al., 2026)	Separates oracle opportunity from realizable value in a declared candidate-outcome experiment	Selection-valid opportunity and realizability audits	Its ladder remains valid; our construction changes the action experiment despite identical contracted behavior evidence
Task-specific routing; ReMoE (Yoon et al., 2026; Hu et al., 2026; Zhu et al., 2026)	Supervised or directly evaluated expert interventions; reuse-aware router updates	Positive task-specific quality or multi-runtime serving results on modified models	These action-specific measurements are compatible with, not counterexamples to, our label-validity boundary
PagedWeight; WiSP (Yang et al., 2026; Zhang et al., 2026)	Expert-weight precision/residency and KV capacity are jointly managed	Directly measured memory–quality and serving tradeoffs	They evaluate deployed configurations; we check whether behavior-derived values justify a candidate choice
R-AutoEval+ (Park et al., 2025)	Combines real and automatic labels for reliable candidate evaluation, including quantization selection	Reliability and efficiency guarantees under its assumptions	Proxy correction and risk control are established; our discrepancy concerns action-induced execution
Ours	Two failure mechanisms; same-checkpoint information separation; complete candidate and loss-budget audits	Fixed-route repair; feedback under controlled-FP32 routing; selection witnesses with threshold-dependent consequences	Which execution evidence validates a candidate comparison, and when does a replay error change the deployed decision?

What deterministic execution and FP32 can repair. Fixing dispatch order, initial state, kernels and collection shape removes specific sources of execution mismatch. FP32 reduces rounding error; it does not make floating-point reductions associative or guarantee identical results across batched and sliced evaluation (PyTorch Contributors, 2026). Our controlled-FP32 audit keeps router arithmetic

matched while changing expert outputs. Later routers then receive different inputs. Fixed-route labels can be repaired with matched state-propagating replay; output-changing actions need the affected computation to be executed or described by a justified invariance or transition model.

Pre-action limits and the connection to off-policy evaluation. Importance-weighted off-policy evaluation requires suitable support and identified probability ratios (Bottou et al., 2013; Wang et al., 2017); observing an expert somewhere in a behavior trace does not supply its output at a new action-induced hidden state. A known transition model can supply additional information, and the full action law resolves our witness. Partial identification describes the remaining uncertainty as a set of compatible values. Equation (7) gives the selection check once a simultaneous error set is available. Estimating a useful set cheaply before execution is an open algorithmic problem.

Reliable automatic evaluation. R-AutoEval+ (Park et al., 2025) adaptively combines real labels with automatic evaluation, provides finite-sample reliability and sample-efficiency guarantees under its assumptions, and includes LLM quantization selection. Its automatic evaluator predicts labels for candidate models. In our setting, the intervention changes the computation from which a replayed label was obtained. We query executed challengers to test the resulting choice. Prediction-powered methods would be relevant comparators for extending this finite comparison to mean-risk certification. Our current query strategies use ordinary first-witness stopping and gap/replay order.

Opportunity ordering. Opportunity ordering conditions on a specified candidate-outcome experiment. Our construction asks whether behavior-replay outcomes determine deployment outcomes. The behavior oracle and input-conditioned Bayes policy already agree in the example, so increasing policy capacity cannot recover the missing action-path information. The ranking reverses between two runtimes despite that fully realized behavior opportunity.