SAME COUNTS, DIFFERENT COST: MEASURING THE ORDER GAP IN SPARSE MIXTURE-OF-EXPERTS ROUTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapt a sparse Mixture-of-Experts model to a new task and you can easily measure how much its routing changed: how often each expert fires, how far the choices have moved, which experts fire together. Those summaries are then used to decide what to keep resident in memory. Every one is a count, a rate or a set overlap, and none records the order in which experts are accessed. We call the cost of that omission a trace's *order gap*: what randomizing token order does to simulated miss rate while every one of the trace's own counts is preserved exactly. With 16 of 64 experts resident per layer it is 0.135 on full traces, holding to within 0.005 across two checkpoints, an independently adapted second run and a sixfold increase in coverage, while adaptation itself moves the same metric by 0.0008. A separate comparison at twice the optimizer budget, on its own population, shifts the mean order gap by 0.000232. Replayed on real expert-sized tensors on an RTX A6000, randomizing token order adds 14.2 seconds of expert transfer to the actual order's 38.9, a 36.6% increase. Order gaps also vary across real checkpoints, by more than adaptation moves the metric over 225 checkpoint-layer cells from four adapted runs; each carries its own count-matched baseline, so that comparison is descriptive. The insufficiency is then exact rather than comparative. A constrained kernel additionally holds each prompt's aggregate overlap with the pre-adaptation checkpoint fixed, so per-prompt expert counts, per-expert rank counts and aggregate overlap are identical on both sides, and the configuration it reaches over 120 prompts and all 15 eligible layers, invariants checked on each of the 1,800 cells, differs by $+0.015465$ in miss@16; replayed on a 20-prompt, five-layer subset it adds 2.25 seconds of expert-loading time. One such pair shows that no function of those summaries alone reproduces the simulated cost; our sampler does not mix, so those figures describe the configuration reached rather than a population effect. Behind the constructions is the measurement program that calibrates them. Under an empirical null stratified by task, router rank, resident-set size and token position, newly selected experts show excess recent-window residency of $+0.037129$ and $+0.037296$ in two OLMoE-1B-7B trajectories and three to six times less in Qwen1.5-MoE-A2.7B, while matched comparisons give no evidence that the longitudinal residual exceeds that between two independently adapted runs. The claim is narrow: these summaries do not determine the cost they are used to reason about, and the gap is worth a third of this workload's expert transfer time on real hardware. The replay measures transfer in isolation, so we claim no end-to-end benefit.

1 INTRODUCTION

Suppose you have adapted a sparse Mixture-of-Experts model to a new task and you want to know whether it became more expensive to serve. You cannot easily measure that directly, so you measure what you can. You count how often each expert fires, how far the routing has moved from where it started, which experts tend to fire together. The numbers change, sometimes a lot, and it is natural to read that change as a change in cost: different experts, more swapping, more paging.

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.

This paper is about how far that reading carries. The summaries are widely reported and widely used to decide what to keep resident. Fiddler pins experts in GPU memory by offline-profiled popularity and computes an *expected hit rate* directly from that ranking (Kamahori et al., 2025); MoE-Infinity drives prefetching and cache replacement by cosine-matching a matrix of per-expert token counts against previously seen requests (Xue et al., 2024); Klotski prefetches from a cross-layer co-activation frequency table (Fang et al., 2025); Lina allocates devices by expert popularity (Li et al., 2023). In continual settings the same family of statistics is reported as routing drift and treated as damage (Hou et al., 2026; Xie et al., 2026; Dai et al., 2022), while continual pre-training work reads routing imbalance directly as "a proxy for the worst case latency of an MoE layer" (Thérien et al., 2025).

Every statistic in that list is *order-invariant*: a count, a rate, a pairwise co-occurrence or a set overlap, none of which records the sequence in which experts are accessed. The gap this opens is easiest to see with two experts and a cache that holds one. The access sequence $ABAB$ misses every time, because each expert is evicted just before it is needed again. The sequence $AABB$ misses twice and hits twice. Both use A twice and B twice; every count, rate and co-occurrence anyone could report about them is identical. The cost is not, and nothing in the summary distinguishes them. For some cost models that is harmless, and we are careful not to claim otherwise. All-to-all communication volume and expert-parallel load imbalance are themselves order-invariant, so an order-invariant summary is a legitimate proxy for them (Li et al., 2023; Zhang et al., 2025; Thérien et al., 2025). Cache and paging cost is different: it has been known since Mattson et al. (1970) that miss ratio under LRU is determined by the reuse-distance distribution, which is a property of the access order and not of per-item counts. That is why work optimising locality directly measures it on replayed traces under a temporal policy rather than inferring it from a summary (Suram, 2026; Zhu et al., 2026). Our target is the step from an order-invariant routing summary to an order-dependent locality cost.

Our central result is about the size of what the summaries omit. Define a trace's *order gap* as the cost of randomizing its token order while preserving every one of its own counts exactly: it is what the actual arrangement is worth, and no summary above records it. On full traces the order gap is 0.135 in miss@16, roughly 217 MB of expert traffic per token, while adaptation itself moves the same metric by 0.0008, about 1.3 MB/token (§5). We give the two magnitudes rather than their ratio on purpose: the smaller of them is inside the spread the simulator's own access conventions produce, and a ratio taken against a quantity that small would say more about our noise floor than about either effect.

That comparison is about magnitude. A separate construction settles the logical point exactly: a constrained kernel that additionally holds each prompt's aggregate overlap with the pre-adaptation checkpoint fixed, so that all three preserved summaries are identical on both sides, still moves miss@16 by $+0.015465$ (Figure 3). One such pair is enough to show that no function of those summaries alone reproduces the cost, and it needs no sampler to mix; §7 states what it does and does not require.

The rest of the paper is the measurement program that makes that demonstration interpretable, reported with its limits: the residency estimates are descriptive and family-specific, and the same-stage comparison contains two adapted models, so it cannot identify the effect of removing adaptation.

Contributions.

1. **A calibration ladder** placing adaptation, summary-preserving transformations and the simulator's own modelling choices on one axis (§5), which is what makes a small locality effect interpretable rather than indistinguishable from a dead instrument.
2. **A demonstration that reported routing summaries are insufficient** for the locality cost they are used to argue about, by explicit construction rather than by a null test (§7).
3. **A measurement protocol** for conditional reassignment structure: an empirical null stratified by task, router rank, resident-set size and token position; prompt-clustered intervals; and matched common-position contrasts that make longitudinal and same-stage arms comparable (§4).
4. **Descriptive residency estimates in two architectures** (§6), including the finding that both the estimate's magnitude and the ordering of its comparison arms depend on how the null draws its replacement experts, which bounds what any single such choice establishes.

5. **Two failed tests, reported as limits on interpretation** rather than as evidence of absence (§7). One is a causal substitutability test whose interval includes zero. The other is a decision rule of our own design, which we withdraw: it compares two quantities as a ratio, and their signs differ, so it returns the same verdict whatever their magnitudes.

2 SETUP

Everything that follows runs on stored routing traces, so this section is mostly an account of what was recorded and what it can carry.

Model, data, traces. OLMoE-1B-7B-Instruct (64 experts, top-8, 16 sparse layers) is adapted with LoRA (r=4) on expert projections over TRACE tasks C-STANCE $\rightarrow$ FOMC $\rightarrow$ MeetingBank, producing checkpoints C_0 (pre-adaptation) through C_3. At each checkpoint we replay a fixed set of 441 probe prompts with `output_router_logits`, recording the top-8 expert ids, their probabilities and the top-k boundary margin for every (layer, sequence, position): 2,352,848 routing events per checkpoint. Two router conditions are run: *frozen* (router untouched; routing changes only through hidden-state drift) and *full-router fine-tuning*. Two run-ID-labelled trajectories are available for both. The local source uses the supplied seed for run naming, initialization and data shuffling, but its replay writer incorrectly hardcodes the embedded trace seed to zero. Without the authoritative executed launch/checkpoint record, we retain run-ID labels and do not treat this field as independent confirmation of training-seed identity.

Probe hygiene. A complete, retained normalized-hash audit covering all 441 probe prompts and every training split is required before claiming zero overlap. The FOMC dataset audit reports 16/496 train–test duplicates; this limits what the accompanying task-learning numbers can support.

The statistic. For each position where the top-k set changes between a reference checkpoint and a later one, and for each expert e swapped *in*, we ask whether e was resident in the recent window (the union of top-k sets over the preceding 64 tokens of the same prompt). This is recent-window set membership, not a hit in a capacity-limited cache. The raw membership rate is approximately 0.948 in the original OLMoE comparison. The quantity of interest is the *lift* Δ_{res} of that rate over a null that redraws the swapped-in expert from a matched stratum.

3 CONDITIONAL SUBSTITUTION STRUCTURE

A raw residency rate of 0.95 sounds high until you ask what it would be by chance, and that question is where most of the work in this paper sits. The empirical null redraws a swapped-in expert within declared strata. The strict version conditions on task and router rank, and uses quantile bins of resident-set size and token position. Binned conditioning does not hold the latter quantities exactly fixed; the legacy implementation splits tied values by its event-order/argsort rule. Table 1 reports two comparisons on the same original frozen-router run-0 population, $C_1 \rightarrow C_3$. Intermediate rows from a cheaper regenerated trajectory were removed because they did not share this population.

Table 1: Same-population descriptive null comparison, 1,224,452 substitutions. Historical permutation estimates are reported here; exact-null inference is recorded separately with its provenance.

Null	Mean-layer excess residency
Free permutation	$+0.1195$
Task + rank + size-bin + position-bin	$+0.0371$

Conditioning substantially reduces estimated excess residency. This is a comparison of estimands under different nulls, not a causal decomposition assigning a percentage of the effect to each confound.

Exact empirical expectation. For event k in stratum S, the empirical null expectation is $\mathbb{E}[\text{hit}_k] = \sum_e c_e(S)\mathbb{1}[e \in W_k]/|S|$. Here hit_k denotes recent-window membership, not a finite-cache outcome. The implementation computes the expectation exactly using resident masks, with each

model's expert-pool and top-k dimensions checked against its traces. Earlier intervals used a 20-draw approximation and predate this change; they are retained in the audit record but are not presented as validated exact-null intervals. The new CPU analysis uses exact expectations and 5,000 joint prompt-cluster resamples. Layers 1–15 receive equal weight and swaps within each layer receive equal weight; layer 0 has no eligible swaps. The intervals condition on the fitted strata and donor pools. We make no synthetic-coverage claim.

Run comparisons and estimand. The exact frozen-router $C_1 \rightarrow C_3$ estimates are $+0.037129$ with conditional 95% interval $[+0.035410, +0.038721]$ and $+0.037296$ with interval $[+0.035396, +0.039021]$. An exploratory sensitivity using exact resident-set size and actual window fill in place of quantile bins gives $+0.035628$ and $+0.036581$, again with conditional intervals above zero. This is a different, non-nested recipe: fill $\min(t, 64)$ merges late positions that the absolute-position quartiles distinguish, while exact size removes another binning approximation. It is not uniformly stronger conditioning, and a residual need not decrease between these recipes. Prompt resampling is joint across layers and contrasts. These intervals measure evaluation-prompt uncertainty conditional on fitted empirical nulls; they do not propagate training or null-fitting uncertainty.

For the full-router arm, historical permutation estimates for $C_1 \rightarrow C_2$ are $+0.0249$ and $+0.0244$. The run-1 $C_1 \rightarrow C_3$ estimate is $+0.0307$; run-0 C3 is truncated and excluded.

4 Longitudinal and same-stage comparisons

Four descriptive constructions probe what the statistic measures (Table 2). Their populations differ, so their ratios are not equivalence tests. The pseudo-swap arm samples one selected expert at unchanged positions; the longitudinal arm uses actual swap-ins. Eligible layers and rank compositions can therefore differ.

Table 2: Four ways of asking what the residency statistic measures, on OLMoE run 0. Each row changes one thing: which interval is compared, whether the positions had their routing changed at all, and which window residency is measured in. Populations differ across rows, so these are descriptive comparisons rather than equivalence tests, and both models in the cross-run row were continually adapted.

Construction	Mean lift
Run 0, $C_1 \rightarrow C_3$, C3 recent window	$+0.0371$
Pseudo-swaps at unchanged positions	$+0.0231$
Real swap-ins, C1 recent window	$+0.0381$
Run-0 C3 versus run-1 C3	$+0.0350$

The cross-run point estimate is approximately 94% of the longitudinal estimate. This demonstrates a similar descriptive association for another comparison among adapted models, but does not show that adaptation is unnecessary. C1 is after the first training task, so the C1-window arm precedes the $C_1 \rightarrow C_3$ interval rather than all adaptation. Formal comparison requires common support, consistent weighting and uncertainty on the difference, with any equivalence margin declared separately.

We implement a subsequent exploratory comparison using each run's C3 as the common target and recent window, retaining only token positions whose selected set changes in both comparisons. Residuals receive equal weight within positions, positions within layers, and layers 1–15. The adaptation-minus-cross-run differences are -0.000139 (conditional paired 95% interval $[-0.000681, +0.000386]$) for target run 0 and -0.000906 ($[-0.001601, -0.000150]$) for target run 1. Thus this comparison does not support a larger adaptation-associated residual. It also does not establish equivalence or isolate adaptation: both models were adapted, and each arm retains its own fitted donor pool.

5 Calibrating the locality measurement

Suppose we tell you that adaptation moves simulated miss rate by 0.004. You have no way to judge that. It could mean adaptation leaves locality alone, or it could mean our instrument cannot detect

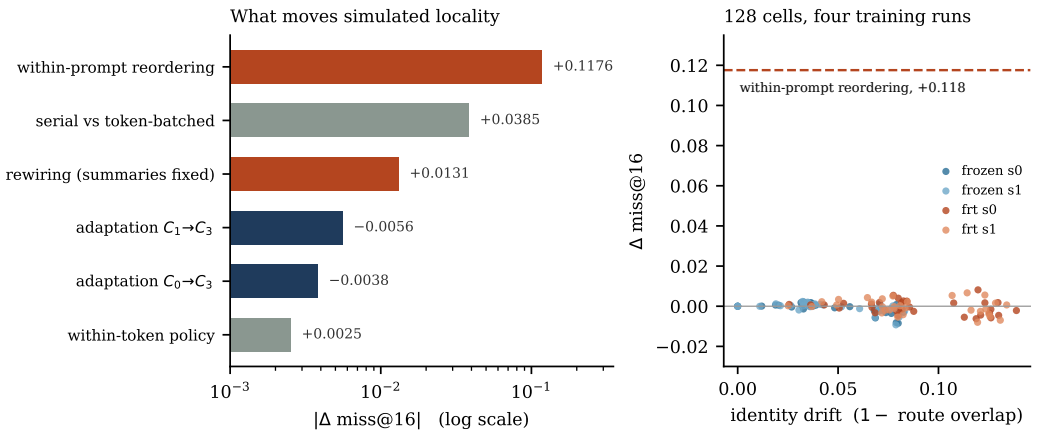


Figure 1: **Left:** everything we can make simulated locality do, on one axis and one configuration. Blue is real adaptation, orange holds the count-style summaries fixed, grey is a choice the simulator's author makes; note the log scale. The reordering bar is the largest, but its invariant is asserted against a co-permuted reference rather than the stored C_0, so the arm carrying the insufficiency claim is the constrained rewiring bar beside it. The serial-versus-batched convention alone moves the metric more than adaptation does. **Right:** identity drift against locality movement for 128 layer-by-checkpoint cells from four training runs. Drift spans 0 to 0.139 while the cells stay near zero on the scale set by the dashed line. The two are correlated within runs; what the figure shows is the difference in magnitude, not independence.

anything at all, and the two readings call for opposite responses. The only way to tell them apart is to find out how far the needle moves when something really does move it. So before interpreting any of our own numbers, we measure the range: the same prompts, the same three layers, the same tie policy and the same prompt-reset cache, with every intervention on one axis (Table 3, drawn to scale in Figure 1).

Table 3: What moves simulated miss@16, same subset and configuration throughout, under both access conventions. Identity drift is $1-$ mean per-token selected-set overlap. For the adaptation rows it is measured between the two checkpoints that row names; for the rewiring and reordering rows it is measured against C_3, the trace being transformed. The reference therefore differs by row. MB/token converts the token-batched column at 12.58 MB per expert and 128 expert accesses per token. Ranges across chains or seeds are given in the text and are not confidence intervals. The bold row is the one that holds all three preserved summaries fixed, including aggregate overlap with the stored C_0; the reordering row below it asserts its invariant against a co-permuted reference instead, and is read only for scale.

	Identity	Δ miss@16		
Intervention	drift	token-batched	serial	MB/token
Adaptation $C_0 \to C_3$	0.067	−0.0038	−0.0044	−6.2
Adaptation $C_1 \to C_3$	0.058	−0.0056	−0.0069	−9.0
Constraint-preserving rewiring	0.044	**+0.0131**	**+0.0143**	+21.2
Within-prompt reordering	0.733	+0.1176	+0.1244	+189.4
Serial instead of token-batched simulation	n/a	+0.0385		+62.1
Rank-ordered instead of expected-tie recency	n/a	+0.0025		+4.1

Three readings follow, and the order matters because the arms are not equally clean.

First, the instrument is not dead: the within-prompt reordering row moves the metric about thirty times further than adaptation does on this subset. That row carries a caveat we state here rather than in a footnote. Its invariant is asserted against a co-permuted reference rather than the stored C_0, so it

preserves counts and rank counts exactly but not overlap with the actual pre-adaptation trace, which is why its identity drift reads 0.733 against adaptation's 0.067. It bounds what the count-preserving degree of freedom is worth; it is not an instance of holding all three preserved summaries fixed. Second, the arm that does hold per-prompt expert counts, per-expert rank counts and aggregate overlap with the stored C_0 fixed is the constrained rewiring kernel. Run on the same nine-prompt subset as the rest of this table it moves miss@16 by $+0.013145$; run on 120 prompts and all 15 eligible layers it moves it by $+0.015465$, with the five-chain range narrowing from ± 0.0007 to ± 0.0002 (§7). That is the arm the insufficiency claim rests on. On the full traces it is $+0.0112$ to $+0.0155$ depending on capacity, against an adaptation effect of -0.00005 to -0.00079. We give both sides in absolute terms rather than as a ratio, because at capacity 8 the adaptation denominator is -0.00005 and any ratio there would divide by our own noise floor. It is nonetheless smaller than the serial-versus-token-batched modelling choice at $+0.0385$, which is the third reading: a locality effect of adaptation's size, taken from a simulator of this kind, is smaller than the simulator's own unvalidated conventions. That is why we report both conventions throughout, and why we treat the ladder as establishing scale rather than as a measurement of adaptation.

The order gap, on full traces. The ladder's subset is small, so we measure the count-preserving degree of freedom where we have the most data. For each checkpoint we compare simulated miss rate under the trace's actual token order against a randomized order that preserves every per-prompt and per-expert rank count exactly, asserted per cell in code. On 120 prompts and all 15 eligible layers, actual order gives miss@16 of 0.415334 at C_0 and 0.414545 at C_3, while randomized order gives 0.550216 and 0.551714. The order gap is therefore 0.1349 at C_0 and 0.1372 at C_3, and it stays within 0.005 of those values on a narrower subset and in an independently adapted second run (Appendix D). At 12.58 MB per expert and 128 accesses per token that is 214 to 221 MB of expert traffic per token, against 1.3 MB/token for the adaptation effect on the same traces. This is the most stable quantity in the paper and the one we would ask a reader to remember.

Does real adaptation move the order gap, or only our kernel? A construction reaching two traces that agree on the three preserved summaries and disagree on cost shows that those summaries do not determine the cost, but not that training travels along the dimension separating them. Over 225 checkpoint-layer cells from four run-ID-labelled trajectories on 60 prompts, holding the layer fixed, order gaps vary across checkpoints and runs with a median within-layer spread of 0.0073, of which 0.0034 falls within a single run. Each checkpoint is compared against its own count-matched baseline, so this does not identify a pure arrangement effect, and these cells are not the 120-prompt population the construction uses, so we form no ratio (Appendix D).

What we cannot decompose. It is tempting to split adaptation's own effect into a part carried by counts and a part carried by arrangement, and we tried. The split is exact by construction but it does not hold still: at our best-powered coverage the count component reverses sign and the two components partly cancel, leaving a total smaller than either (Appendix C). We therefore report no channel attribution for adaptation's own effect, which is a statement about that effect's size rather than about arrangement, whose magnitude is settled above.

What capacity 16 is, in bytes. An OLMoE expert is three 2048×1024 matrices, 12.58 MB at bf16, as recorded in the trace metadata. Capacity 16 therefore holds 0.20 GB per layer and 3.22 GB across the 16 sparse layers, a quarter of the 12.88 GB of expert weight. At the measured C3 miss rate of 0.4286 and 128 expert accesses per token, that is roughly 690 MB of expert traffic per token. We state this to make the operating point legible, not to claim a deployment. A system that cared about this cost would pin a hot set, prefetch a layer ahead, or batch requests, and this simulator does none of those. The comparison the ladder supports is between interventions measured on the same simulator, not between this simulator and a machine, and we make no claim about which operating points are deployable.

The order gap in measured seconds. Every number above is a simulated miss rate, and a simulator is not a machine. So we replayed the traces against real hardware: 64 expert-sized tensors at the true shape in pinned host memory, 16 resident on an RTX A6000 under serial LRU, three sequences interleaved across repeats (Appendix E). The actual order takes 38.85 s; randomizing it costs $+14.20$ s, a 36.6% increase in expert transfer time, and the constrained rewiring, holding those

three summaries fixed, still costs $+2.25$ s, or 5.8%. Repeats agree within 0.25 s. Sustained bandwidth is 19.7 GB/s against the 24 GB/s this paper assumed, and the measured-minus-simulated miss gap of 0.024 to 0.039 is the serial-versus-batched convention difference the ladder already reports. Nothing here overlaps transfer with compute, so absolute times are an upper bound; what survives contact with hardware is the ordering and the relative sizes.

And under a policy that pins hot experts. Under purely static frequency pinning the cost really is a function of the counts, so our constructions could not disturb it, and several systems in Table 5 pin by popularity. Real policies are hybrids, so we sweep the mix: pin the top N per layer by aggregate count, run replacement over the rest. The effect decays gracefully rather than vanishing, retaining 89% of its unpinned value with a quarter of the budget pinned and 75% with half, and reaching exactly $+0.000000$ only when pinning consumes the whole budget, as preserved counts require (Appendix F). The pinned set is identical on both sides because counts are invariant, asserted per layer, and pinned experts never miss: both choices favour the objection. It holds only at that degenerate endpoint.

The same picture on real checkpoints. The ladder uses a small deterministic subset. On the full probe set we compare identity drift against locality movement across 128 layer-by-checkpoint cells drawn from four training runs, two per router arm. Identity drift spans 0 to 0.139; mean absolute miss@16 movement is 0.0022 with a run-clustered 95% interval of $[0.0017, 0.0028]$. Because 32 cells share a trajectory, we resample whole runs rather than cells; resampling cells would overstate precision by treating one trajectory as 32 observations. This is a different subset, reference checkpoint and cache configuration from the ladder, so we quote it separately and do not form ratios across the two.

Within runs the two quantities are not independent: per-run correlations between identity drift and miss@16 movement are -0.63, -0.57, -0.28 and -0.24. The interval for the pooled correlation depends on the resampling unit, and we give both: clustering on runs gives $[-0.60, -0.22]$, which excludes zero, while clustering on layers gives $[-0.51, +0.03]$, which does not. Runs are the unit that varies for our purposes, so we prefer the first, but a reader who treats layers as exchangeable should have the second. The full-router arm drives $1.90\times$ the identity drift of the frozen arm and $1.69\times$ the absolute locality movement, which is closer to proportionality than to independence. The claim here is therefore about magnitude, not about decoupling: identity drift does carry locality information, and the locality movement it carries is small on the scale the ladder establishes.

6 A SECOND ARCHITECTURE

One model family is a case study, not a finding, so we repeated the analysis on a second. We used Qwen1.5-MoE-A2.7B (60 experts, top-4, 24 layers) for two stored runs, using the same 441 probes and the same task sequence, over all 23 eligible layers (layer 0 contributes no events in any contrast). Sparsity and tokenization differ from OLMoE, so nothing here isolates architecture as the cause of any difference.

Under the binned null the longitudinal estimates are $+0.010142$ $[+0.00771, +0.01241]$ and $+0.006448$ $[+0.00376, +0.00901]$, positive in 20 and 19 of 23 eligible layers, while the two cross-run arms sit at or below zero. Under exact size/fill conditioning every arm rises to $+0.017$ or above, including both cross-run arms, so the recipe changes not only the magnitude but which arms look distinguishable. Table 4 gives all eight rows with intervals and swap counts.

Two things follow, and they pull in different directions. The longitudinal residual is positive in this family too, but three to six times smaller than OLMoE's $+0.0371$ under the binned null. And the qualitative relationship between the longitudinal and cross-run arms is not stable across donor-matching recipes: under the binned null the cross-run arms sit at or below zero, while under exact size/fill they reach $+0.017$, within a fifth of the longitudinal estimate. A reader who chose one recipe in advance would draw a different picture of the cross-run arm than one who chose the other. We report both rather than nominate a winner, and we treat the recipe dependence as a limit on what the full-population arms can establish.

The matched comparison is less exposed to that choice. Restricting to token positions whose selected set changes in both arms, sharing the target C3 windows, and weighting layers equally

Table 4: Qwen1.5-MoE conditional residuals over all 23 eligible layers, with 5,000 shared prompt-cluster draws. Two donor-matching recipes are reported because the choice changes both the magnitude and the ordering of the arms. *exact size/fill* is not a nested strengthening of *binned*: it replaces the absolute position quartiles, and fill saturates at position 64.

Arm	Null	Swaps	Estimate	95% interval	Pos. layers
Longitudinal, run 0	binned	895,752	$+0.010142$	$[+0.00771, +0.01241]$	20/23
Longitudinal, run 1	binned	932,817	$+0.006448$	$[+0.00376, +0.00901]$	19/23
Cross-run, target 0	binned	650,644	-0.003213	$[-0.00648, -0.00012]$	11/23
Cross-run, target 1	binned	650,644	-0.001806	$[-0.00484, +0.00114]$	11/23
Longitudinal, run 0	exact size/fill	895,752	$+0.021446$	$[+0.02012, +0.02277]$	23/23
Longitudinal, run 1	exact size/fill	932,817	$+0.020574$	$[+0.01923, +0.02190]$	23/23
Cross-run, target 0	exact size/fill	650,644	$+0.017086$	$[+0.01573, +0.01847]$	23/23
Cross-run, target 1	exact size/fill	650,644	$+0.017611$	$[+0.01633, +0.01891]$	23/23

gives, for Qwen under the binned null, an adaptation-minus-cross-run difference of -0.001978 $[-0.00259, -0.00138]$ with target run 0 and -0.000953 $[-0.00156, -0.00034]$ with target run 1. The corresponding OLMoE differences are -0.000139 $[-0.00068, +0.00039]$ and -0.000906 $[-0.00160, -0.00015]$.

Switching to exact size/fill conditioning keeps the Qwen differences negative (-0.002305, -0.001991) and keeps OLMoE target 1 negative (-0.001984 $[-0.00257, -0.00136]$), but turns OLMoE target 0 positive: $+0.000174$ $[-0.00033, +0.00066]$. We report that cell rather than the three that agree, because it is the one a reader should weigh. Its interval includes zero, so it is not evidence that the adaptation residual exceeds the between-run residual either; what it shows is that the sign of these matched differences is not recipe-invariant, and that the honest summary across all eight cells is no evidence that the adaptation residual exceeds the between-run residual, rather than a consistent negative in either direction.

So the one statement that survives in both families, under both null recipes, and on matched populations is a negative one: these comparisons give no evidence that the adaptation residual exceeds the residual between two independently adapted runs. Both arms contain adapted models, so this bounds how much of the residual can be attributed to a particular adaptation trajectory; it does not isolate adaptation, and it is not an equivalence test.

7 Two failed interventions

Two preregistered interventions were meant to strengthen the descriptive picture, and neither did. An eval-time test replaced a selected expert's output with a matched alternate and measured old-task perplexity; validation was non-vacuous but the result does not pass ($D = +0.0011$, 95% CI $[-0.00018, +0.00243]$), and its matching drops popularity in $\sim$98% of events. A matched-drift comparison against router noise gives median per-layer changes of about $+0.0059$ for noise and -0.0015 for adaptation, differing in direction in 25 of 32 layer–run cells. The preregistered ratio rule for that comparison returns FAIL, and we withdraw its interpretation rather than lean on it: numerator and denominator have opposite signs, so a negative ratio clears the threshold whatever the magnitudes are. Both tests, their defects and the finite-step rewiring kernel that produced Figure 2 are described in Appendix A.

8 Related work

The table makes our scope precise. The upper block is where an order-invariant summary is used to reason about a cost that depends on access order. Our construction bounds those inferences where the summary is built from the three quantities it preserves, and does not test co-activation tables or predicted activation sets. The middle block is not a target: when the cost really is a sum over tokens, a count is the right statistic and nothing we show disturbs it. The lower block is the opposite of our result and supports its premise, since work that takes locality seriously measures it temporally; Suram (2026) finds that LRU beats static pinning "because the exploitable signal is temporal", and

Table 5: Which routing statistic, used for which decision, and whether it records access order. Our construction holds per-prompt expert counts, per-expert rank counts and aggregate C_0 overlap fixed, so it bears on order-dependent rows keyed on those, and not on rows keyed on cross-layer co-activation or a predicted activation set, which the kernel does not preserve.

Work	Routing statistic	Decision it drives	Order
Cache and paging cost under order-dependent replacement, where a count-only summary is insufficient			
Fiddler (Kamahori et al., 2025)	per-expert popularity	pin hot experts, expected hit rate	inv.[†]
MoE-Infinity (Xue et al., 2024)	per-expert count matrix	prefetch, cache replacement	inv.
Klotski (Fang et al., 2025)	cross-layer co-activation	prefetch set, execution order	inv.
SiDA-MoE (Du et al., 2024)	predicted activation set	load/offload per token	inv.
Pre-gated MoE (Hwang et al., 2024)	layer-ahead predicted set	migrate next block	inv.
Mixtral-offload. (Eliseev & Mazur, 2023)	LRU recency, hit rates	cache size, prefetch	mixed
Communication and load-imbalance cost: order-invariant, so the inference is legitimate			
Lina (Li et al., 2023)	expert popularity	device allocation	inv.
C2R (Zhang et al., 2025)	expert co-activation	all-to-all volume	inv.
CPT routers (Thérien et al., 2025)	routing imbalance	worst-case layer latency	inv.
VELA (Hu et al., 2025)	access probability	expert placement	inv.
Locality measured or optimised directly rather than inferred			
Cacheable (Suram, 2026)	replayed traces, LRU/Belady	train for cacheability	sens.
ReMoE (Zhu et al., 2026)	short-horizon reuse	bias router to recent	sens.
Sticky Routing (Kayyam, 2026)	token-to-token switch rate	training-time objective	sens.
WiSP (Zhang et al., 2026)	working set, prediction	finds bandwidth binding	sens.

[†] Static popularity pinning alone is count-determined (Appendix F); the insufficiency applies to replacement and hybrid policies.

Zhang et al. (2026) reports that better routing prediction does not help single-stream decode because bandwidth rather than prediction quality binds.

Two works are close enough to state precisely. Engmann et al. (2026) audits observational routing-importance metrics against interventional expert ablation and finds the step from population summary to token-level claim unsupported; ours is the same move on the systems side. Chen et al. (2026) reports that sparse, well-balanced expert utilisation does not imply localised updates under continual training, the same shape of claim about a different outcome. On the continual-learning side, routing drift is measured and treated as damage to be suppressed (Hou et al., 2026; Xie et al., 2026; Dai et al., 2022). Wang et al. (2025) uses an order-invariant co-activation summary to choose expert substitutes but measures throughput directly rather than inferring it, and Tian et al. (2026) shows that overlap statistics do not determine functional redundancy. We are not aware of work that holds the reported routing summaries fixed and asks whether the simulated locality cost is thereby determined.

9 Limitations and conclusion

Two model families and two run-ID-labelled trajectories each do not establish architecture-wide or task-population generality, and the probe set was repeatedly inspected during analysis development. Qwen residuals differ from OLMoE in magnitude and conditioning sensitivity, and incomplete launch provenance prevents a verified match of training details. The cross-run comparison contains two adapted models, so it cannot identify the effect of removing adaptation; the rewiring chain is not a validated uniform null, and the failed causal test does not establish a null effect. The hardware replay measures expert transfer in isolation, so no end-to-end benefit is established, and routing comes from one teacher-forced pass per probe, so every access sequence is prefill-order. Appendix G has these in full.

Count how often each expert fires and you have learned something real about a model. You have not learned what it costs to serve. Two traces can agree on every count, rank profile and overlap with the starting checkpoint and still differ in simulated cache cost by more than adaptation produces, and by 2.25 seconds of measured transfer. We show it by construction, not by a failed test, which is why it survives our sampler's poor mixing: one pair suffices. Adaptation moves locality very little, while the order gap it cannot see is 0.135 and 14.2 seconds. The two are not independent and we do not say they are; what separates them is magnitude. The practical reading is narrow: to decide what to pin or prefetch, measure the access sequence; to call locality undisturbed, say what moving it would have looked like, since a flat line and a dead instrument look alike.

REPRODUCIBILITY STATEMENT

Every figure in this paper is computed from stored routing traces and regenerated from them rather than transcribed. The corpus is 32 parquet files, roughly 2.3 GB: OLMoE-1B-7B-0924-Instruct at four run-ID-labelled trajectories with a frozen router and two with a fully fine-tuned router, and Qwen1.5-MoE-A2.7B at two, each over checkpoints C_0 through C_3 on the same 441 probes and the same task order. One file is truncated in storage and is reported as unreadable rather than dropped.

The anonymised supplementary material contains the trace reader, the cache simulator, the two occupancy baselines, a worked one-command example, and a manifest giving a SHA-256, row count and readability flag for every trace. It also carries the limitations that bound reuse: the prefill-order caveat above, the hardcoded embedded seed that makes these run-ID replications rather than verified seeds, and the absence of a recorded prompt/answer boundary. The traces themselves exceed the supplementary size limit and will be released with a persistent identifier on publication.

AI USE STATEMENT

Generative AI tools were used substantially in this work, and we disclose their use at the task level as the ICLR 2027 policy requires.

They were used to implement methods and analysis code, to propose and refine hypotheses, to give feedback on experimental design and methodology, to clean and reformat data, to retrieve and analyse prior work, to help interpret results, and to draft and edit this manuscript. They were not used to generate any synthetic dataset; every number reported here comes from model runs and from analyses of the stored traces those runs produced. This paper contains no formal proofs, so AI tools contributed none. No translation was involved.

We reviewed all AI-assisted work. Analysis code was checked against independently computed values and, where a result is load-bearing, against the stored artifact it claims to summarise; the figures cited in the text are regenerated from those artifacts. Retrieved prior work was verified against the original sources, and several citations first produced with AI assistance were corrected or removed after that check. Every preregistration, gate and stopping decision recorded here was made by the authors. We take responsibility for the final content of this work, including its text, claims and artifacts.

REFERENCES

Anrui Chen, Ruijun Huang, Xin Zhang, Fang Dong, Hengjie Cao, Zhendong Huang, Yifeng Yang, Mengyi Chen, Jixian Zhou, Mingzhi Dong, Yujiang Wang, Jinlong Hou, Qin Lv, Robert P. Dick, Yuan Cheng, Tun Lu, Fan Yang, and Li Shang. Multi-head attention as a source of catastrophic forgetting in MoE transformers. *arXiv:2602.12587*, 2026. URL https://arxiv.org/abs/2602.12587.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. StableMoE: Stable routing strategy for mixture of experts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 7085–7095, 2022.

Zhixu Du, Shiyu Li, Yuhao Wu, Xiangyu Jiang, Jingwei Sun, Qilin Zheng, Yongkai Wu, Ang Li, Hai Li, and Yiran Chen. SiDA: Sparsity-inspired data-aware serving for efficient and scalable large Mixture-of-Experts models. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024. URL https://arxiv.org/abs/2310.18859.

Artyom Eliseev and Denis Mazur. Fast inference of Mixture-of-Experts language models with offloading. *arXiv:2312.17238*, 2023. URL https://arxiv.org/abs/2312.17238.

Leonard Engmann, Christian Medeiros Adriano, and Holger Giese. From observation to intervention: A causal audit of expert importance in Mixture-of-Experts models. *arXiv:2606.10703*, 2026. URL https://arxiv.org/abs/2606.10703. ICML 2026 Workshop on Philosophy of Science Meets Machine Learning, non-archival.

Zhiyuan Fang, Yuegui Huang, Zicong Hong, Yufeng Lyu, Wuhui Chen, Yue Yu, Fan Yu, and Zibin Zheng. Klotski: Efficient Mixture-of-Expert inference via expert-aware multi-batch pipeline. In *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2025. doi: 10.1145/3676641.3716261.

Zhiyan Hou, Haiyun Guo, Haokai Ma, Yandu Sun, Yonghui Yang, and Jinqiao Wang. PASs-MoE: Mitigating misaligned co-drift among router and experts via pathway activation subspaces for continual learning. *arXiv:2601.13020*, 2026. URL https://arxiv.org/abs/2601.13020.

Chenghao Hu, Yufei Kang, and Baochun Li. Communication-efficient MoE fine-tuning with locality-aware expert placement. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 166–176, 2025.

Ranggi Hwang, Jianyu Wei, Shijie Cao, Changho Hwang, Xiaohu Tang, Ting Cao, and Mao Yang. Pre-gated MoE: An algorithm-system co-design for fast and scalable Mixture-of-Expert inference. In *ACM/IEEE International Symposium on Computer Architecture (ISCA)*, 2024. doi: 10.1109/ISCA59077.2024.00078.

Keisuke Kamahori, Tian Tang, Yile Gu, Kan Zhu, and Baris Kasikci. Fiddler: CPU-GPU orchestration for fast inference of Mixture-of-Experts models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2402.07033.

Ali Kayyam. Sticky routing: Training MoE models for memory-efficient inference. *arXiv:2607.08780*, 2026. URL https://arxiv.org/abs/2607.08780.

Jiamin Li, Yimin Jiang, Yibo Zhu, Cong Wang, and Hong Xu. Accelerating distributed MoE training and inference with lina. In *USENIX Annual Technical Conference (ATC)*, 2023. URL https://arxiv.org/abs/2210.17223.

R. L. Mattson, J. Gecsei, D. R. Slutz, and I. L. Traiger. Evaluation techniques for storage hierarchies. *IBM Systems Journal*, 9(2):78–117, 1970. doi: 10.1147/sj.92.0078.

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:2608.18261*, 2026. URL https://arxiv.org/abs/2608.18261.

Benjamin Thérien, Charles-Étienne Joseph, Zain Sarwar, Ashwinee Panda, Anirban Das, Shi-Xiong Zhang, Stephen Rawls, Sambit Sahu, Eugene Belilovsky, and Irina Rish. Continual pre-training of MoEs: How robust is your router? *arXiv:2503.05029*, 2025. URL https://arxiv.org/abs/2503.05029.

Huiyuan Tian, Bonan Xu, and Shijian Li. Beyond geometric complementarity: Coherent overlap in sparse Mixture-of-Experts routing. *arXiv:2607.28308*, 2026. URL https://arxiv.org/abs/2607.28308.

Yun Wang, Lingyun Yang, Senhao Yu, Yixiao Wang, Ruixing Li, Zhixiang Wei, James Yen, and Zhengwei Qi. BuddyMoE: Exploiting expert redundancy to accelerate memory-constrained mixture-of-experts inference. *arXiv:2511.10054*, 2025. URL https://arxiv.org/abs/2511.10054.

Zhen-Hao Xie, Jun-Tao Tang, Yu-Cheng Shi, Han-Jia Ye, De-Chuan Zhan, and Da-Wei Zhou. SAME: Stabilized Mixture-of-Experts for multimodal continual instruction tuning. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2602.01990.

Leyang Xue, Yao Fu, Zhan Lu, Luo Mai, and Mahesh Marina. MoE-Infinity: Efficient MoE inference on personal machines with sparsity-aware expert cache. *arXiv:2401.14361*, 2024. URL https://arxiv.org/abs/2401.14361.

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:2606.21868*, 2026. URL https://arxiv.org/abs/2606.21868.

Mohan Zhang, Pingzhi Li, Jie Peng, Mufan Qiu, and Tianlong Chen. Advancing MoE efficiency: A collaboration-constrained routing (C2R) strategy for better expert parallelism design. In *Proceedings of NAACL*, 2025. URL https://arxiv.org/abs/2504.01337.

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. In *ICML*, 2026. URL https://arxiv.org/abs/2605.27081.

A THE TWO FAILED INTERVENTIONS IN FULL

Two preregistered interventions were meant to strengthen the descriptive picture. Neither did. We report both for what they rule out, and for the defects they exposed in our own instruments.

Eval-time causal substitutability. A preregistered test replaced a selected expert's output with a matched in- or out-of-neighbourhood alternate and measured old-task NLL. Validation was non-vacuous (628 patches; identity patch moves NLL by 0, a real swap by 0.061), but the test does not pass: $D = +0.0011$, 95% CI $[-0.00018, +0.00243]$, median event effect ≈ 0, and 10 of 400 prompts contribute 113% of the summed effect. Its matching also drops popularity in $\sim 98\%$ of events and its permuted control is a graph-label permutation rather than the preregistered degree-and-popularity-preserving null. We report it as a failed test with known defects, not as evidence of a null effect.

Matched identity drift and signed locality changes. A corrected positive control compares C0-referenced adaptation against router noise interpolated to each layer's adaptation drift, with signed deltas throughout. Median per-layer miss-rate changes are about $+0.0059$ for noise and -0.0015 for adaptation, differing in direction in 25 of 32 layer–run cells, which are not independent training replications. The preregistered ratio rule returns FAIL (median -2.47), and we withdraw its magnitude interpretation rather than lean on it: with opposite signs in numerator and denominator a negative ratio clears the threshold regardless of relative magnitude, so the outcome says nothing about which intervention moves locality more. We report the signed quantities instead.

A new descriptive reanalysis avoids the ratio and averages signed contrasts equally over all 16 layers. At capacity 16, noise minus adaptation is $+0.007643$ and $+0.007153$ in the two adapted runs. It is positive in 15/16 layers in each run at capacities 8, 16 and 32. The runs share the same noise curves; these are not independent noise replications. Only aggregate noise metrics are retained locally, so a paired prompt interval and alternative within-token access ordering cannot yet be evaluated. The prospective confirmation requires raw noise traces, concentration controls and a model-aware noise comparison on the second architecture; the Qwen null analysis alone does not provide that comparison.

A finite-step concentration-preserving trace intervention. We implement a separate CPU pilot on the original OLMoE run-0 traces: the first probe identifier per task (nine prompts), up to 256 tokens each, and layers 1, 8 and 15. We then rerun the identical kernel, invariants, chain count and seeds over 120 prompts and all 15 eligible layers, reported at the end of this section. Starting from C3, each proposal exchanges two experts at the same rank between two positions of one prompt/layer. We reject exchanges that create a duplicate within a top-k set or alter the prompt/layer's total set overlap with C0. The kernel preserves per-prompt expert counts, per-expert rank counts and aggregate C0 overlap exactly; it can redistribute identity changes among token positions and does not fix per-token drift. We use $K_p = 20\,T_p$ proposals for a prompt of length T_p, with five separately seeded RNG runs.

For fixed traces R, a cache metric M, the specified kernel and proposal counts K, the target is

$$\theta_K(R) = \mathbb{E}_U\big[M(T_K(R;U)) - M(R)\big]. \tag{1}$$

This finite-step expectation does not require stationary or uniform sampling over constrained traces. It is distinct from a uniform permutation-null target. The five RNG runs estimate this intervention on the selected traces; they do not establish population-wide or model-level effects.

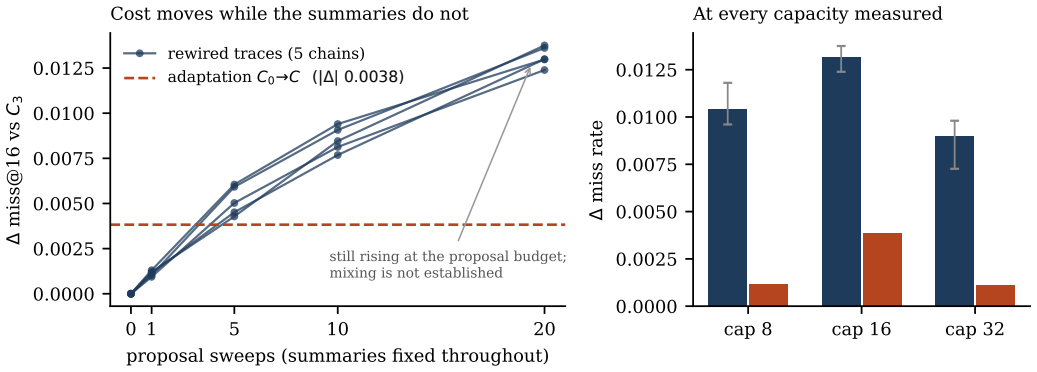


Figure 2: The reported summaries do not determine the simulated cost. Every point holds per-prompt expert counts, per-expert rank counts and aggregate C_0 overlap exactly fixed; only the distribution of identity changes over token positions differs. **Left:** five independently seeded chains at capacity 16. The rewired traces pass the $C_0 \to C_3$ adaptation difference itself between sweeps 3 and 5 and keep rising at the proposal budget, so the sampler has not mixed and no uniform-null reading is warranted; the non-determination conclusion does not require mixing. **Right:** at the final sweep the rewiring moves miss rate further than the adaptation does at every capacity measured, by factors of 9.0, 3.4 and 8.1 at capacities 8, 16 and 32. Error bars are the five-chain range, not a confidence interval. Nine prompts and three layers, the preregistered subset; the full-trace run over 120 prompts and 15 layers is reported in §7 and moves the metric further in the same direction. Expected recency-tie policy.

The cache resets at each prompt, unlike the earlier concatenated-stream metric. The simulation is *token-batched*: every selected expert is checked against the cache state before that token, then recency is updated for the whole selected set. It is not literal serial-access LRU. We compare rank and reverse-rank recency updates, paired random expert priorities, and an expected recency-tie policy. Under the latter, all experts co-selected by a token share its timestamp; if a tie crosses the capacity boundary, occupancy is averaged uniformly over the tied experts. Exhaustive small-stream enumeration agrees with averaging all within-token ordering histories. At capacity 16, the mean finite-step change under this policy is $+0.013145$, with observed five-run range $[+0.012390, +0.013753]$. This range is not a confidence interval.

All invariants pass, but only 5,921 of 420,300 proposals are accepted (1.41%), and displacement continues to increase over the pilot. Mixing is not established, so no uniform-null p-value is warranted. This does not invalidate the finite-step target. The reached traces demonstrate that the preserved summaries do not uniquely determine the token-batched cache cost on this subset under the specified kernel. They do not establish a learned temporal mechanism, a valid perturbed-model forward pass, or hardware benefit. Confirmation of θ_K requires a predeclared population and prompt/RNG uncertainty with the kernel and K fixed in advance.

The same kernel on the full traces. The subset above is the one we preregistered, and it is small. We therefore rerun the identical kernel, invariants, capacities, policies, chain count and chain seeds over 120 prompts and all 15 eligible layers, changing only which prompts and layers are loaded. That is 1,800 prompt/layer cells against 27, and 28,521,000 proposals against 420,300. Every invariant check passes on every cell after every sweep. At the final sweep, under the expected-tie policy, the mean change is $+0.013313$ at capacity 8, $+0.015465$ at 16 and $+0.011194$ at 32, with five-chain ranges of ± 0.00013, ± 0.00016 and ± 0.00007. The adaptation differences on the same traces are -0.000050, -0.000789 and -0.000590. Two properties of this run matter more than the point estimates: the effect is *larger* on 66 times more data rather than regressing toward zero, and the chain-to-chain spread is roughly four times tighter. As an unplanned consistency check, this run's $C_3 - C_0$ miss@16 of -0.000789 matches the value computed by `order_decomposition.py`, a separately written script, to six decimal places. Acceptance remains low at 531,892 of 28,521,000 proposals (1.86%) and displacement is still rising at the budget, so mixing is no more established

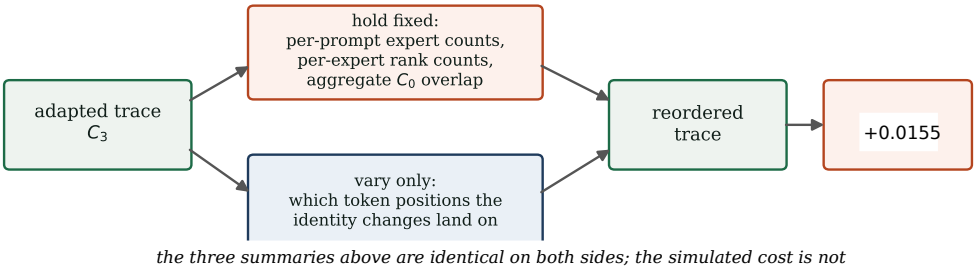


Figure 3: The constrained construction in one line, on the expanded population of 120 prompts and all 15 eligible layers. Both traces have identical per-prompt expert counts, per-expert rank counts and aggregate C_0 overlap; they differ only in which token positions the identity changes occupy. Those three summaries are what is held fixed, and the $+0.0155$ shown is the constrained kernel's movement on that population, not the larger within-prompt reordering arm of Table 3, whose reference is co-permuted. Simulated cost differs by more than adaptation itself produces.

here than in the pilot, and the existential reading is the same one: a larger reached set of pairs, not a population estimate.

Measurement checks that did pass. Applying a global bijection to expert ids in a saved trace leaves the checked locality metrics unchanged, establishing label invariance of those proxies. The original modelled cache policy uses token-batched accesses with rank-ordered recency updates. That choice moves absolute miss rates by up to 0.003 and should be declared. Finally, the common claim that these traces show "no temporal locality" because shuffling gives the same miss rate is an artifact of shuffling *prompt order* only. Shuffling tokens *within* prompts raises miss@16 by 0.08 to 0.16.

What the construction needs, and what it does not. The claim is existential, and Figure 3 is one instance of it: two trace configurations agreeing on per-prompt expert counts, per-expert rank counts and aggregate C_0 overlap can disagree on simulated cost, and establishing that needs one pair, not a representative sample. So the sampler's poor mixing ($531{,}892$ of $28{,}521{,}000$ proposals accepted on the full traces, displacement still rising at the budget) does not undermine it; mixing would matter only if we were estimating a population quantity, which we are not. The preserved invariants are chosen to match what the literature reports rather than what is convenient, and what the kernel deliberately does not preserve is the assignment of identity changes to token positions. It changes token-level expert assignments subject to those three invariants, and does not preserve all co-activation statistics. Consequently the result extends to a class: any cost proxy that is a function of expert-use histograms, per-expert rank profiles and aggregate overlap alone is constant across the pairs we construct while the simulated cost is not. It does not extend to summaries encoding temporal structure, such as inter-arrival statistics or windowed residency; our own residency statistic is in that second group. The contribution is to say where not to look.

B SIMULATOR AND ITS VALIDATION

Because every quantity above is measured through a simulator, its behaviour is part of the evidence.

Access model. Simulation is token-batched: all experts selected by a token are checked against the cache state as it stood before that token, then recency is updated for the whole selected set. This is not literal serial-access LRU, and we say so rather than implying hardware fidelity. Within-token update order is a free choice, so we compare four policies: rank order, reverse-rank order, paired random priorities, and an expected recency-tie policy under which co-selected experts share a timestamp and boundary-crossing ties are averaged uniformly over the tied experts. Exhaustive enumeration on small streams confirms that the tie-expected policy agrees with averaging over all within-token ordering histories, which is why we report it as the primary policy. The choice moves absolute miss rates by up to 0.003, and any paper quoting simulated miss rates should declare it.

Three checks that pass, and one common claim that does not survive. Applying a global bijection to expert identifiers in a saved trace leaves the checked metrics unchanged, so those proxies are label-invariant and cannot be reflecting identifier artifacts. The cache resets at prompt boundaries in the pilot, rather than running as one concatenated stream, and we report which metric each number belongs to. The third check is the one that matters most for interpretation: the common claim that these traces exhibit "no temporal locality" because shuffling leaves miss rates unchanged is an artifact of shuffling *prompt order* only. Shuffling tokens *within* prompts raises miss@16 by 0.08 to 0.16. There is substantial within-prompt temporal structure; the shuffle that failed to find it was shuffling at the wrong granularity.

Reproduction. Trace files are verified by hash before analysis, with dimensional checks that each model's expert-pool size and top-k match its recorded routes, and row alignment checked per checkpoint. The OLMoE probes produce $2{,}352{,}848$ routing events per checkpoint across 16 layers; the Qwen traces align $3{,}310{,}368$ rows per checkpoint across 24 layers with $E{=}60$, $k{=}4$. All analyses in this paper run on CPU from stored traces. The rewiring pilot uses nine deterministically selected prompts, three layers, at most 256 tokens each, $K_p = 20\,T_p$ proposals for a prompt of length T_p, and five separately seeded runs. The full-trace rerun uses the same code path and seeds with 120 prompts drawn by a fixed seed and all 15 eligible layers, and the nine-prompt configuration reproduces byte-identically after the change that made the subset selectable.

C THE ORDER AND COUNT DECOMPOSITION, AND WHY WE WITHDRAW IT

For each checkpoint we measure simulated miss rate under the trace's actual token order and under a randomized order preserving every per-prompt and per-expert rank count exactly. The count component is the difference between the two randomized rates, the order component is the change in how much arrangement helps each trace, and the two sum to the total by construction, so the residual is an arithmetic check rather than evidence.

The result depends on coverage in a way that forbids interpretation. At 60 prompts and 5 layers both components are negative and arrangement accounts for most of the movement: total -0.002659, count -0.000202, order -0.002457. The independently adapted second run agrees in sign: total -0.001935, count -0.000470, order -0.001465. But at 120 prompts and all 15 eligible layers the count component reverses sign to $+0.001498$ against an order component of -0.002287, so the two partly cancel and the total, -0.000789, is smaller in magnitude than either component.

A share computed from absolute magnitudes, which is how we first reported this, conceals exactly that cancellation, and we no longer report one. The honest reading is that adaptation's own effect at the coverage where we have the most data is too small relative to the simulator's conventions to attribute to a channel at all. This says nothing about the magnitude of the arrangement degree of freedom itself, which is measured directly in §5 and does not depend on this decomposition.

D HOW FAR REAL ADAPTATION MOVES THE ORDER GAP

For every available OLMoE checkpoint we compute each layer's order gap: simulated miss@16 under the trace's actual token order against a randomized order preserving that cell's own per-prompt and per-expert rank counts exactly, averaged over five shuffle seeds, with the count and rank-count invariants asserted. Because the randomization is per-cell, each checkpoint is compared against its own count-matched baseline. That removes the level difference counts would produce, but it does not license reading the remaining variation as a pure arrangement effect: each checkpoint carries its own count-conditioned baseline, and interactions between counts and arrangement are not excluded by the construction. Throughout this appendix the gap is actual minus randomized, the opposite sign to the main text's randomized minus actual, so the values here are negative where §5 is positive.

The sample is 60 prompts and all 15 eligible layers over 15 checkpoints: four runs spanning the frozen and full-router arms at C_0 through C_3, less one full-router endpoint whose trace file is truncated in our mirror at 52.2 MB against the 85.7 MB the other fifteen carry. That file is excluded explicitly rather than silently, and no quantity reported anywhere in this paper depends on it: the full-router figures come from per-checkpoint metric summaries computed on the machine that produced the traces, not from these files.

Across the resulting 225 cells the order gap has mean -0.1434 and standard deviation 0.0345, ranging from -0.2012 to -0.0872. Most of that variation is between layers, which differ systematically, so the comparison that answers the reachability question holds the layer fixed. Within a layer, the spread across the four runs and four checkpoints has median 0.007265, minimum 0.002382 and maximum 0.012254, of which the within-run part across a single run's own checkpoints is 0.003391. These 60-prompt cells are not the population the constrained kernel's $+0.015465$ and adaptation's -0.000789 come from, which is the 120-prompt expanded run, so we compare them descriptively and form no ratio: the denominator would be the adaptation effect, which §5 declines to divide by because it lies inside the spread the simulator's own access conventions produce. What the numbers support is that order gaps vary across observed checkpoints and trajectories by an amount comparable to the constrained kernel's movement and larger than adaptation's.

The spread divides about evenly between two sources. Within a single run, across its own four checkpoints, the median within-layer spread is 0.003391 with a maximum of 0.009426; across trajectories at C_3 alone, where three of the four have usable traces because the excluded endpoint above is one of them, it is 0.002864 with a maximum of 0.010220. Both training progression and the trajectory label move the order gap, at comparable magnitude.

One further arm answers a question the main text raises but cannot otherwise settle. An extended sweep adds a fifth trajectory given twice the optimizer steps per task, which makes it a different operating point rather than a replicate, so it is carried separately and never pooled with the 225 cells above. At C_3 its mean order gap is -0.144871 over its 15 cells, one per layer, against -0.145103 over the 30 cells of the two frozen trajectories run at the standard budget, a difference of 0.000232. That is a small observed difference rather than an equivalence result: doubling the optimizer budget moved these means very little, which is suggestive but is not a test that the order gap is independent of how much a model was adapted.

Two limits. Four runs across two arms is a small sample of trajectories, so this is a range over what we have and not an estimate of a population range; and C_0 is the unadapted base model, shared across runs by construction, so C_0 cells are not independent between runs. The claim this supports is narrow: order gaps observed across real checkpoints and trajectories vary by an amount comparable to what the kernel achieves, so the quantity the kernel manipulates is not one the training runs we have hold still. Because each checkpoint carries its own count-matched baseline, this does not establish that training moves the kernel's particular degree of freedom.

E THE HARDWARE REPLAY IN FULL

The replay is a memory-system benchmark, not a model forward pass. Its access sequence is the recorded routing of a real checkpoint, which is the quantity the simulator abstracts; the tensors carry the true expert shape ($3 \times 2048 \times 1024$, bf16, 12.58 MB) but not trained weights, which is irrelevant to transfer cost and avoids any claim about outputs. Sixty-four such tensors sit in pinned host memory, 16 slots are resident on the device, and a miss copies one expert host-to-device and evicts the least recently used. Timing brackets each whole sequence with device synchronisation, so it measures transfer and nothing else. The three sequences are replayed interleaved within each repeat, so drift and thermal effects apply equally to all of them. Subset: 20 prompts and layers 1, 4, 8, 11, 15, drawn as a prefix of the rewiring run's own selection so that every replayed cell has a matching rewired state.

Table 6: Measured on an RTX A6000, median of three repeats. Simulated miss rates are the tie-expected policy on the same cells. Bytes are miss count times expert size, so their agreement with the simulator is a correctness check rather than a finding; the measured quantity of interest is time.

Sequence	Time (s)	vs actual	Miss rate	Simulated	GB/s
Actual token order	38.85	—	0.4159	0.3919	19.66
Randomized, counts preserved	53.05	+14.20	0.5676	0.5287	19.65
Constrained rewiring	41.10	+2.25	0.4399	0.4141	19.66

Three readings from Table 6. The order gap is 36.6% of the actual-order transfer time, so the degree of freedom the summaries omit is worth a third of this workload's expert traffic. The constrained

construction, which preserves the three summaries above, costs 5.8% of measured time, so the insufficiency is not an artifact of the simulator. And measured miss rates run 0.024 to 0.039 above the tie-expected simulator on the same cells, consistent in sign and rough size with the $+0.0385$ serial-versus-token-batched gap reported in §5, which is a check on the simulator rather than a new result.

What this does not establish: a deployed system overlaps transfer with compute, batches requests and prefetches, none of which happens here, so these are upper bounds on cost and the comparison between sequences is the measurement. One device, one capacity, one subset.

F THE PINNED-POLICY SWEEP IN FULL

Pinning the top N experts per layer by aggregate count and running the tie-expected policy over the remaining $C - N$ slots. Because per-layer expert counts are invariant under the kernel, the frequency-derived pinned set is identical between the adapted and rewired traces; this is asserted per layer in code rather than assumed. Pinned experts are treated as preloaded and never missing, which is what pinning means and is also the choice most favourable to the objection, since it maximises the share of cost that counts alone determine. Two correctness conditions follow and both are checked: with nothing pinned the simulator reduces exactly to the tie-expected one used throughout the paper, reproducing $+0.015465$ and -0.000789; and with the budget fully pinned the cost is a function of counts alone, so the rewiring effect must be exactly zero. Table 7 reports the sweep.

Table 7: The insufficiency under a hybrid pinned-plus-replacement policy, 120 prompts and 15 layers, five chains at the final sweep. The last row is a correctness check that preserved counts force to zero, not a finding. An oracle pinned set drawn from the traces themselves is an advantage a deployed system would not have.

	capacity 16		capacity 32	
Pinned share	Δ miss	retained	Δ miss	retained
0% (pure replacement)	$+0.015465$	100%	$+0.011194$	100%
12%	$+0.014615$	94%	$+0.010733$	96%
25%	$+0.013785$	89%	$+0.010066$	90%
50%	$+0.011667$	75%	$+0.008531$	76%
75%	$+0.007107$	46%	$+0.007238$	65%
100% (pure pinning)	$+0.000000$	0%	$+0.000000$	0%

The adaptation effect over the same sweep stays between -0.000789 and $+0.001934$ at capacity 16 and between -0.001551 and -0.000018 at capacity 32, that is, within the band we already describe as indistinguishable from the simulator's own conventions, at every pinned share.

G THREATS TO VALIDITY

The construction is small. The preregistered construction was nine prompts and three layers, which is a demonstration rather than a survey. The full-trace rerun covers 120 prompts and all 15 eligible layers, so the non-determination claim no longer rests on a small subset, but it is still one model family, one training run and one pair of checkpoints. It remains sufficient for non-determination and insufficient for any statement about how much cost is typically undetermined across models, and we report magnitudes as measured rather than as general quantities.

A simulator is not a machine. We measure expert-loading replay time on one device, at one capacity, over one subset of prompts and layers (Appendix E). That does not establish end-to-end serving latency, throughput, energy, or any benefit under the compute-transfer overlap, batching and prefetching a deployed system performs. The insufficiency result says that a class of summaries fails to determine a simulated quantity; how far that quantity tracks real serving cost beyond the replay we report is a separate question, and one that a systems evaluation with a real offloading regime would have to answer.

Provenance is incomplete. The replay writer hardcodes the embedded trace seed, so we retain run-ID labels rather than claiming independent training seeds, and we avoid any conclusion that depends on seed independence. The residency estimates are reported per run for this reason.

The descriptive estimates depend on the null recipe. Binned and exact size/fill conditioning give different magnitudes and, in Qwen, reorder the comparison arms (§6). We report both rather than nominating one, and we treat any conclusion that survives only under a single recipe as unestablished.

What would change our mind. A kernel that preserved the same summaries and left simulated cost unchanged, on a substantially larger and better-mixed sample, would suggest our reached traces are unrepresentative even if the existential claim survives. A demonstration that the temporal degree of freedom we exploit cannot arise from actual adaptation, only from artificial rewiring, would restrict the practical relevance of the result without touching its logic; Appendix D is our attempt to pre-empt that, and it would be undercut by a larger set of trajectories showing real checkpoints clustered far more tightly in order gap than ours are. And a residency estimate that remained stable across donor-matching recipes, in both families, would make the descriptive half of this paper considerably stronger than we currently claim it is.