Computing manuscript No.
(will be inserted by the editor)

Online Resource 1: Supplementary Information Module Counterfactuals for Attributing LLM Serving Scheduler Effects

Soroush Vahidi

Computing. Online Resource 1 provides extended results and reproducibility details for the accompanying article on large language model (LLM) serving. We use arrival-normalized weighted goodput (ANWG), service-level objective (SLO), key–value (KV) memory and graphics processing unit (GPU) as in the main paper. A confidence interval (CI) summarizes uncertainty; Benjamini–Hochberg (BH) adjustment uses the stated false discovery rate (FDR) threshold q. B denotes bootstrap draws, r Pearson correlation and n the stated sample count. Δ is treatment minus base ANWG; I is the unscaled four-arm interaction contrast in main Eq. 5.

A source window is a selected block of workload records; changing its replay rate does not create independent records. A workload instance fixes inputs and a seed namespace. A cell is the explicitly identified combination of workload and intervention; a fidelity unit additionally identifies a module candidate, window, rate and seed. Structural opportunity permits a choice; actionable disagreement means different recommendations at a shared state. Local erasure (also called override) means that disagreement yields the same admitted-request set. Paired trajectory divergence, matched-request consequences and aggregate outcomes are separate observations. The prospective opportunity panel uses top-choice disagreement as its actionable-opportunity diagnostic; it is not the local-erasure population. Prefill processes input tokens; fairness aging promotes waiting requests. A guard filters eligibility. Implementation names and commit identifiers below locate reproducibility evidence.

S1 Seed-level interaction strips

Figure S1 shows per-seed interaction estimates for the two prospective four-arm confirmatory cells.

Soroush Vahidi (corresponding author)
New Jersey Institute of Technology, Newark, NJ, USA
E-mail: sv96@njit.edu

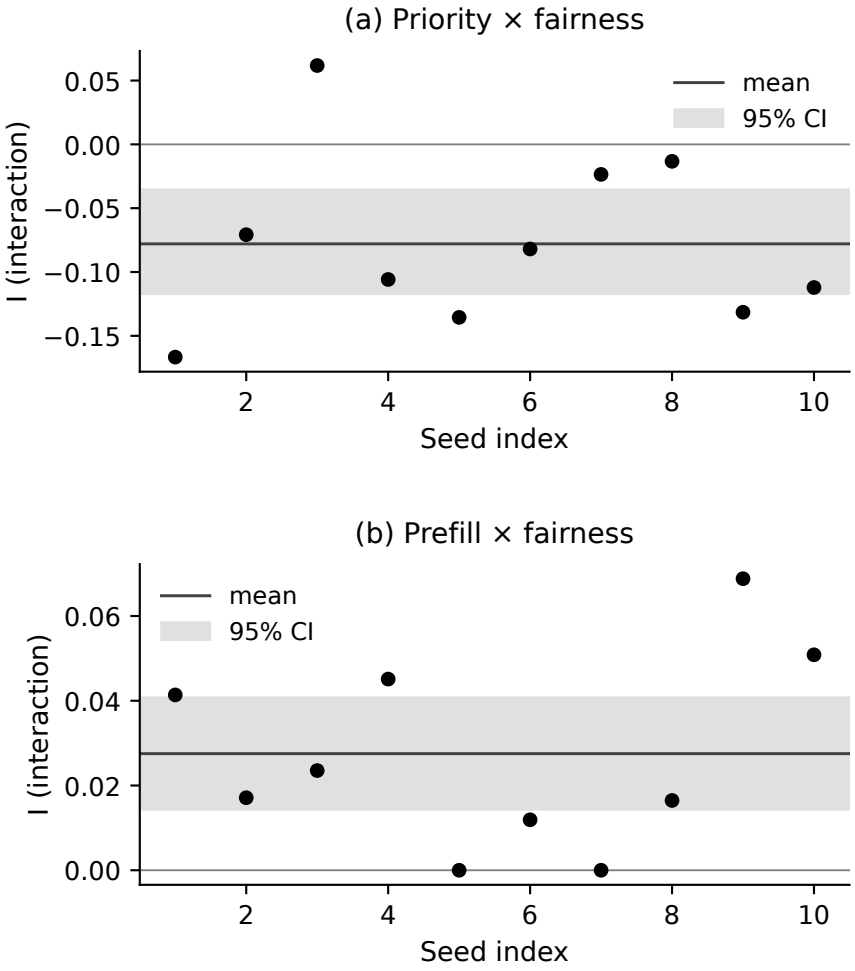


Fig. S1 Per-seed interaction estimates for (a) priority×fairness and (b) prefill×fairness (prospective confirmatory campaigns). Horizontal line: mean I; band: instance-clustered 95% CI

S2 Policy and module inventory

Table S1 summarizes manuscript-visible base policies and donor modules. It is not an exhaustive 72-intervention catalog. WSP denotes weighted shortest processing, EDF earliest deadline first, and ESTF estimated service time first; SLO denotes service-level objective and KV key–value memory.

Table S1 Representative policies/modules in the frozen inventory

Policy / module	Slot	Operational meaning	Fidelity
WSP, EDF, ESTF	Priority	Rank admit-filtered queue	Reconstructed
SLO slack, Scorpio priority	Priority	Urgency / SLO-aware rank	Reconstructed
Admission control, Scorpio budget	Admission	Filter waiting queue	Reconstructed
Adaptive chunked prefill	Prefill	Chunk/guard filter	Reconstructed
Reconstructed reserve alias	KV guard	KV-budget placement filter	Exploratory
Aging priority / tiebreak	Fairness	Wait-time promotion	Reconstructed
Round-robin / pressure-first-fit	Placement	GPU assignment mode	Reconstructed

S3 Seed lists

Fresh seed streams for the priority and prefill confirmatory campaigns are frozen per instance in the campaign workload-row files (30 instances each). Priority seeds span [2,127,662, 2,807,755]; prefill seeds span [3,272,858, 3,763,776]. Neither range is contiguous. These streams are disjoint from exploratory seeds used in earlier, non-confirmatory passes over the same mechanisms.

S4 Preregistration excerpts

The conditions are defined in main Table 2. The priority confirmatory campaign uses the frozen P0/P1/P2 family definitions; the prefill confirmatory campaign uses the F0/F1/F2 definitions; and the actionable-opportunity follow-up uses its own frozen protocol. Each protocol fixes the workload manifest, seed namespace, bootstrap procedure, FDR families at q=0.10, and interpretation labels before outcome analysis. Applying the same frozen p-values at q=0.05 changes few priority confirmatory calls (for example, 16/28 rather than 17/28 FDR-significant under P1); we report the preregistered q=0.10 threshold in the main text. The held-out and public-trace replay analyses similarly record frozen manifests and no-outcome-informed-selection checks with their analysis artifacts.

S5 Extended tables

Large machine-readable tables (full priority forest, secondary endpoints, intervention-isolation checks, corrected KV manifests) remain with the frozen analysis outputs rather than in the PDF.

Correlations by source key for structural and actionable opportunity support the pooled actionable opportunity follow-up (§6.4); each source has n=20 windows (Table S2).

Table S2 Per-source correlations between opportunity diagnostics and final divergence (exploratory; n=20 windows per source)

Source key	$r_{\text{structural}}$	$r_{\text{actionable}}$	n
Azure 2023 Code	0.149	0.289	20
Azure 2023 Conversation	0.983	0.967	20
Azure 2024 Code	0.568	0.622	20
Azure 2024 Conversation	0.896	0.878	20
BurstGPT v2	0.912	0.940	20
Mooncake Conversation	−0.444	−0.157	20
Mooncake ToolAgent	−0.235	0.141	20

Four source keys have structural correlations above 0.5; a fifth is weakly positive (0.149). Mooncake strata drive the pooled reversal: Mooncake Conversation is negative on both diagnostics ($r_{\text{structural}}$=−0.444, $r_{\text{actionable}}$=−0.157), while Mooncake ToolAgent is negative structurally ($r_{\text{structural}}$= − 0.235) but weakly positive on actionable opportunity ($r_{\text{actionable}}$= + 0.141), motivating the pooled follow-up in §6.4.

S6 Comparison of attribution questions

Table S3 compares questions, not superiority. Equivalent instrumentation can collect the same diagnostic observations. The serving systems listed are related work, not experimental baselines.

S7 Same-state downstream erasure

Figure S2 contrasts trajectory ambiguity with local classification and identifies the erasure stages. The main text reports both local erasure estimands and their uncertainty.

Table S3 Comparison of attribution questions. No superiority or inherited native validation is claimed

Approach	Prior question	Present question and relationship
Configuration ablation [5]	Which settings explain performance changes?	Where does a typed replacement act? Equivalent instrumentation can recover the same observations.
Sensitivity [13]	Which inputs explain output variance?	What is a fixed paired effect or conditional erasure rate? Different estimands.
Coz [4]	Which virtual code speedups improve performance?	Which recommendation differences survive? Different intervention; no inherited validation.
CausalSim [2]	How can replay address algorithm-dependent trace bias?	What follows from an internal model intervention? No production identification step here.
Factorial design [7, Section 12.2]	Do two factor effects add?	How does module context change an effect? Established contrast on paired compositions.
Priority, fairness and cache locality [6, 12, 14]	How should order, fairness and cache reuse guide scheduling?	How does a substituted module's recommendation propagate? Attribution, not a new ranking or locality rule.
Phase, placement and scheduling granularity [8–11, 15, 16]	Where and at what granularity should work execute to meet service objectives?	Which declared boundary changes outcomes? Reconstructed interventions; no native implementations.
Serving simulation [1, 3]	How do serving and infrastructure choices affect modeled performance?	What local and trajectory effects follow? A diagnostic contract; no inherited hardware validation.

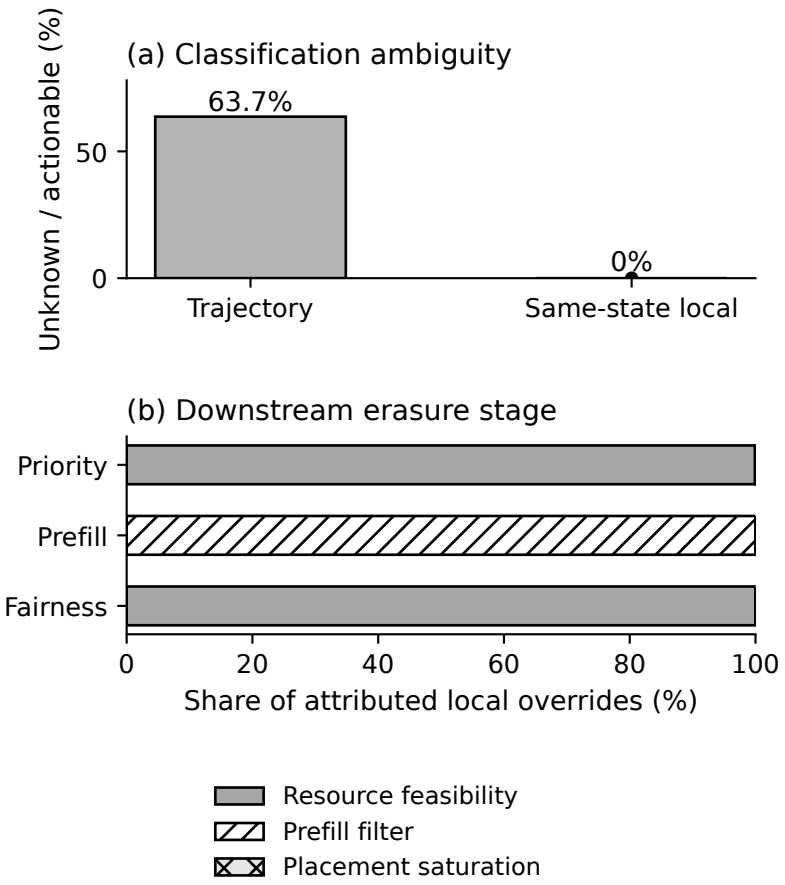


Fig. S2 Mechanism-specific downstream erasure under shared-state local counterfactuals (three selected pairs on base-visited replay states; final action is the admitted-request-ID set). (a) The trajectory comparison leaves 63.7% of actionable steps unclassified (65.1% among overrides); shared-state conditioning reduces ambiguity to zero within the implemented local taxonomy. (b) Attributed local overrides by dominant erasure stage: resource feasibility for priority and fairness; prefill filter for prefill. Stacked bars show shares among attributed overrides only. Pooled event counts (340,548 actionable disagreements) are not independent samples; per-window variation is summarized in the text

S8 Reproducibility

Public release v0.2.0 provides the outcome/count projections, provenance, 28 portable tests and reconstruction commands described in main Sect. 8. Its versioned README is at `https://github.com/SoroushVahidi/llm-scheduler-module-counterfactuals/blob/v0.2.0/README.md`. Sect. S14 maps the additional tables to public files; reconstruction of saved observations is distinct from historical execution.

S9 Post-hoc erasure and KV sensitivities

The historical experiment outputs and analyses are preserved. The following analyses use existing observations only and were not preregistered with the original campaigns. For erasure, seeds and pairs from one source window are never resampled separately. The primary interval is conditional on seven fixed sources. The hierarchical sensitivity draws seven source strata with replacement, then eight windows within each drawn stratum; with only seven sources it is exploratory and does not establish generalization to production systems.

Table S4 Erasure sensitivity intervals (95%; $B = 10{,}000$). Point estimates are 99.3226% event-weighted and 96.0806% positive-cell

Resampling	Event-weighted	Positive-cell
Within fixed sources	[98.80, 99.58]%	[93.23, 98.01]%
Unstratified windows	[98.60, 99.61]%	[92.68, 98.14]%
Sources, then windows	[96.52, 99.75]%	[89.17, 98.73]%

Table S4 reports three resampling schemes. All 168 pair–window cells remain in the resampling frame. The 96 cells with zero denominator supply no defined rate; undefined all-zero bootstrap draws would be recorded and omitted, but none occurred. Numerators and denominators were reconstructed from 56 raw files, each with three seeds, and agree exactly with the frozen window summaries. The three pairs were selected using prior mechanistic evidence and feasibility, before this follow-up; they are not a random sample of the 38 benchmark-eligible pairs.

Table S5 Separate exploratory KV populations: mean ΔANWG and pointwise 95% intervals, jointly resampling ten seed streams

Population	Pairs	Observations	Mean [interval]
Six clean slots	6	2460	-0.01384 [-0.01516, -0.01252]
Historical mixed	13	5330	-0.01376 [-0.01511, -0.01237]
One unflagged pair	1	410	-0.01373 [-0.01507, -0.01240]

Table S5 separates the three populations. The 410 workload instances comprise 80 regime instances and 330 joint-design instances. Each of ten seed streams appears across conditions. The six-slot population contains 2460 paired observations, not 2460 independent replicates. The mixed population includes seven contextual non-KV interventions; the one-pair population uses the aging base. Neither clean slot nor unflagged status establishes native Scorpio compatibility. The historical control rewards are reused across the imple-

mentation correction; complete execution-level version equivalence remains a limitation. The machine-readable evidence register records input hashes, analysis versions, populations, and limitations.

S10 Public-replay resampling and metric recovery

The public-replay frame contains seven source strata, three distinct windows per source, and three rates per window: 21 windows and 63 instances. Each instance has 72 requests, giving 1,512 distinct source records and 4,536 replay appearances. Windows are disjoint within each source; rate variants reuse their records. One selected priority pair and one selected prefill pair supply the diagnostic correlations, not all replay interventions. The stored diagnostic fractions do not retain historical decision counts, so event-weighted rates cannot be independently reconstructed here. Request changes are counted over the 72 input requests; we analyze stored window-level ratios without claiming execution attestation.

We resample three whole windows with replacement within each fixed source, retaining every rate and both diagnostic pairs. The estimand is Pearson correlation across the equal-weight window–rate design, not a correlation of window means. Source centering is recomputed within each draw. Table S6 gives post-hoc percentile ranges that use 10,000 draws and portable seeds derived with the 256-bit Secure Hash Algorithm (SHA-256); no draw had an undefined correlation. These are conditional sensitivity ranges under within-source window exchangeability. Deterministic window selection and only three windows per source preclude calibrated population coverage claims. Source- and rate-specific point estimates are retained in the analysis artifact; no new significance tests are claimed.

Table S6 Public-replay correlations and conditional 95% bootstrap sensitivity ranges. These supersede the incorrectly grouped historical intervals and do not establish population confidence coverage

Diagnostic relationship	r	Range
Priority opportunity–divergence, pooled	-0.241	$[-0.372, -0.112]$
Priority opportunity–divergence, source-centered	0.143	$[-0.015, 0.366]$
Prefill pressure–action divergence	0.898	$[0.628, 0.989]$
Prefill action–request change	0.639	$[0.319, 0.888]$

The metric panel is rebuilt from 30 controlled-priority, 30 controlled-prefill, 45 held-out, and 63 public-replay raw outputs. Five endpoints are joined on workload and intervention identifiers, with lower-is-better metrics sign-reversed before classification. All 3,940 cells have all five endpoints; none are

imputed. Raw-file hashes and recovered base/intervention/effect values are recorded in the metric evidence register and outcome-only comma-separated values (CSV) file. Numerical near-null tolerance is not a test of statistical or practical equivalence.

A later pinned dynamic-fidelity inventory explicitly marks all five flagged reserve-donor candidates as context-dependent and not dynamically executed. It supplies no missing validation for these candidates. They remain inconclusive; the five native-Scorpio interpretations remain invalid. KV results describe exploratory reconstruction behavior only.

S11 Equal-information diagnostic pilot

The equally instrumented ablation pilot is a finite diagnostic comparison, separate from the historical workload populations. Seven synthetic fixtures fix arrivals, unit-rate service, first-in, first-out (FIFO) and earliest-deadline-first (EDF) rankings, capacities and expected schedules. The identity case keeps FIFO in both arms. Protocol/fixtures were frozen at `0bc8e6124d30`; the executed runner/analyzers are recorded at `c3d6bf06b577`, using simulator snapshot `94f4621bb661`. The retained fixtures, complete observations, comparison table and validation records are indexed in the supplementary verification record.

S11.1 Comparison contract and oracle

Method A observes paired aggregate outcomes and can report benefit, harm or null while leaving the mechanism ambiguous. Method B is equally instrumented ablation: generic relational/set queries identify competition, ranking mismatches, equal admitted sets and matched request outcomes. Method C uses explicit diagnostic-stage predicates. B and C receive the identical records, local cloned branches, candidate definitions and request joins. Neither receives the expected schedules as analyzer input. They have the same acquisition cost and no asymmetric tuning or information allowance. A is not scored as incorrect for an appropriately ambiguous answer.

The separate verifier imports neither analyzer nor simulator. It checks expected admissions/completions, deadline arithmetic, waiting/active sets, capacity constraints and ordering rules. It is independent implementation of the elementary oracle, not external adjudication of realistic scheduler failures. The implementation author knew the fixture design; this is not a blinded challenge. A shared acquisition layer also limits the independence of B and C.

Table S7 gives the seven case diagnoses. Erasure is undefined in the three cases without disagreement, not zero. In “both fit,” admitted sets agree but ordered lists differ. The modeled parallel service makes that order difference outcome-irrelevant here; neither method may infer universal action equivalence from set equality. In the occupied-slot case, three disagreements are erased and

Table S7 Identical B/C diagnoses on all seven frozen cases. Opportunity, disagreement, erasure and local differences count base-trajectory decisions; request changes come from paired full trajectories. O: opportunity; D: disagreement; E: erased; L: local action difference. Δ is donor minus base ANWG. No confidence intervals or population-accuracy claim are attached.

Case	O	D	E	L	Requests	Δ
Singleton	0	0	0	0	0/1	0
Agreeing order	1	0	0	0	0/2	0
Both fit	1	1	1	0	0/2	0
Reordered, loose deadlines	1	1	0	1	2/2	0
Deadline crossing	1	1	0	1	2/2	1/2
Occupied slot, then changed order	4	4	3	1	2/3	1/3
Identity replacement	1	0	0	0	0/2	0

one survives. That 3/4 local rate is not a trajectory effect or a multi-module interaction.

S11.2 Execution accounting and evidential limits

Each batch has 28 full trajectories: two arms per case with logger-off/on neutrality controls. It also evaluates 112 discarded local branches, including observer-neutrality acquisition on donor trajectories. Repeating the batch once in a fresh process yields 56 trajectories and 224 branches. The scheduler seed is zero; the two processes use different Python hash seeds (0 and 1). All seven saved records match exactly between batches, all observer-neutrality checks pass, and B equals C in every case. The retained checker reports 112 primitive checks. All 30 entries in the preserved checksum list match the retained files.

The first batch took 0.0213 seconds excluding imports, a descriptive implementation measurement. Because B and C share acquisition and no controlled performance comparison was conducted, it establishes no cost or evaluation-count advantage. The repeat establishes determinism, not independent stochastic reliability. The pilot does not estimate representative diagnostic accuracy, invalid-intervention detection across realistic faults, human interpretation benefit or population-level equivalence. Its negative comparative finding is retained: no advantage over equally instrumented B was observed.

S12 Separate admission and KV-guard erasure extension

The extension protocol is recorded at `11286ef6ad25` and its execution pipeline at `c558bc151188`. It preserves the original local estimand but changes the pairs and base-visited state distributions. Admission replaces the EDF base's

admission slot with the Scorpio-labeled donor's filter. KV guard replaces the Scorpio-labeled base's guard with the KV-constrained policy's guard. Each pair's own base drives the reference trajectory. These are reconstructed mechanisms, not validated native Scorpio interventions.

The frozen frame uses the existing 21 public-replay windows (seven sources, three windows per source) at rate 1.0, not the original erasure study's 56 windows. Seeds 910101, 910102 and 910103 produce 126 unit files. All 42 pair–window cells have identical actionable and erased counts across those seeds. The original analyzer therefore counts one identical seed value per cell, not three independent copies. A fresh check of all raw files reproduces the retained cell counts, with no branch-reconstruction or leakage mismatch.

Table S8 Erasure extension, separately by fixed pair. Each family has 21 windows: 18 positive and three zero-denominator cells. Pooled counts collapse identical seed repeats. Intervals apply to the mean positive-window rate, *not* to the event-weighted fraction.

Family	Erased / actionable	Event rate	Cell mean	Cell-mean 95% interval
Admission	103159 / 103421	99.75%	95.75%	[92.38, 98.54]%
KV guard	6391 / 6996	91.35%	93.15%	[89.18, 96.59]%

Table S8 separates event and cell summaries. Positive-window medians are 99.66% for admission and 94.90% for KV guard; the respective ranges are 76.47–100% and 70.82–100%. This heterogeneity is retained rather than using the high pooled admission rate as a general family-level law. There was no preregistered universal $> 99\%$ erasure hypothesis.

The saved uncertainty calculation resamples the 18 positive-window rates within each fixed family, with $B = 10{,}000$ and random seed 424243. It does not resample individual events, seed repeats or independent module candidates. Inspection of the analyzer shows an unstratified positive-window bootstrap, not the original study's within-source bootstrap retaining the complete zero-denominator frame. The three zero-denominator windows remain explicitly reported and have no imputed rate. These intervals are conditional summaries for selected positive windows; source/time dependence and deterministic window selection limit population interpretation. We preserve the original intervals and do not attach them to the pooled fraction or combine the two campaigns into a common estimate.

Eligibility was checked against the available interface inventory, excluding an admission-control-base KV candidate. Such screening does not override the later native-KV limitations. This extension supports local erasure of the specified modeled outputs; it neither establishes long-horizon equality nor isolates a causal effect of choosing a module family.

S13 Dynamic recommendation fidelity and its limits

S13.1 Common-state comparison and coverage

Campaign `dynamic_transplant_fidelity_v1`, frozen at `a5d77e073190`, compares normalized module recommendations on identical cloned states. "Native donor" in the historical artifact means the donor implementation in the reconstructed repository, not a production engine. The state trajectory is generated by the base policy. Up to 64 early captured states per unit are inspected; rank-based reachability requires at least two entry candidates, while the other tested modules require at least one. Full-policy trajectories and final downstream action equality are not the endpoint.

The inventory contains 72 candidates: 57 dynamically tested and 15 context-dependent exclusions. The tested candidates span 63 configurations (21 windows at three rates) and three seed settings, giving 10,773 units. A fresh inspection reconciles the expected identity set and every stored normalized comparison with the unit counters. It finds no missing units, preliminary-test records, required-field failures, mutation-isolation failures, or recorded determinism failures. Of 689,472 evaluated states, 152,124 meet the module-specific reachability condition.

Table S9 Dynamic fidelity on reachable states. "Positive" counts units with at least one reachable state. Unit means average positive-unit fractions; pooled fractions weight by reachable-state counts. Neither aggregation creates independent workload replications.

Family	Units	Positive	Agree / reachable	Pooled	Unit mean
Admission	3969	3969	60957 / 60957	100%	100%
Priority	5292	3108	70239 / 70239	100%	100%
Fairness aging	1134	666	4329 / 14994	28.87%	50.93%
KV guard	378	378	5934 / 5934	100%	100%
All	10773	8121	141459 / 152124	92.99%	95.98%

Table S9 reports both weighting schemes. The 2,652 zero-opportunity units are reported, not assigned agreement rates. Fifty-one candidates agree exactly on the evaluated reachable states; six fairness candidates do not. The common windows, rate transformations, base-state histories and seed repeats preclude statistical independence. The equal positive-unit mean is descriptive, without a population confidence claim. All five previously flagged reserve-donor candidates are among the exclusions; perfect agreement for two other KV candidates does not close their missing state/boundary evidence. Least-laxity-first is absent as a donor in this frozen campaign and receives no fidelity claim.

S13.2 Interpreting the fairness mismatch

The stored output is a complete ranked-ID list, not an isolated scalar aging increment. The comparison helper evaluates the donor in its own priority context and the transplant with the retained base priority context. In the pinned implementation, the aging donor's priority key already combines request priority, waiting time, estimated service and laxity. The `aging_tiebreak` branch is applied only when the retained priority is not `aging_priority`; it sorts first by that priority key and then by waiting time. Copying the tiebreak slot therefore need not copy the donor's complete ranking. The six candidate means range from 16.36% to 81.55%.

This is observable context dependence of the chosen comparison boundary. It prevents a claim that these transplants universally recover their donor's ranked recommendations, even though the evaluated paths execute without mutation or determinism failures. It does not prove a bug in every fairness rule or invalidate the numerical effects of a separately specified rule. In particular, the controlled interaction runner identifies the distinct `fairness_aging_v1` time-consistent promotion rule. The transplant comparison, local admitted-set erasure and four-arm ANWG interaction each retain their own endpoint and implementation lineage.

S13.3 Prefill boundary and active-state views

Campaign `dynamic_transplant_fidelity_prefill_v1`, frozen at `91e33261cd02`, tests four prefill candidates on the same 63 configurations and three seed offsets: 756 units. Its input boundary is the identical post-admission candidate list. The normalized recommendation is the ordered list retained by the prefill filter. The donor and transplant use the same stateless component and parameters; the protocol explicitly expects agreement. It also checks the donor's modeled pre-ranking pipeline. Thus the study checks the specified boundary and state handling, not independent implementation accuracy.

Table S10 Prefill fidelity: all listed states agree. Views overlap and cannot be added as independent observations. Active census is bounded by the frozen cap; no unit reports truncation.

View / stratum	States	Positive units
Standard: all evaluable	48384	756
Standard: reachable	11697	756
Standard: guard active	4530	207
Active census: guard active	432645	348
First 64 active states: subview	18378	348

Table S10 lists the overlapping views. The standard view samples up to 64 initial states per unit. The active census follows guard-active states with a cap of 5,000 per unit; its first-64 subview gives a separate bounded descriptive view. All 756 raw hashes match the historical manifest. Completeness, integrity and determinism checks report no failures. The original analysis includes binomial upper bounds over unit-level disagreement indicators. We do not promote those to unseen-workload guarantees: units share windows, rates and deterministic components. The observations validate the tested modeled boundary, not native vLLM/Sarathi chunking, arbitrary donor policies or GPU-latency fidelity.

S14 Evidence integration and public-release boundary

The retained evidence includes a machine-readable verification receipt, input hashes, JavaScript Object Notation (JSON) result records, small source/result snapshots and the read-only verification code under `artifacts/computing_stage4a/` and `scripts/stage4a/`. The historical stores remain unchanged. Verification reads existing files and recomputes counts or contrasts; it is not another simulation or real-engine execution. All 312 file references in the frozen original evidence register (215 unique files, including three external dependencies) match their recorded hashes. That establishes preserved file identity, not complete historical execution attestation.

Table S11 Additional evidence lineage. Commit prefixes identify preserved private repository objects; full identifiers and file hashes are in the verification receipt. Eligible outcome/count projections are now in public v0.2.0; the raw files and execution environments are not thereby released.

Evidence	Source / artifact	Commit
Diagnostic pilot	Synthetic fixtures, primary/repeat JSON, checker	`c3d6bf06b577`
Erasure extension	126 raw files and 42 frozen cells	`c558bc151188`
Dynamic fidelity	10773 raw files and candidate crosswalk	`a5d77e073190`
Prefill fidelity	756 raw files and SHA-256 manifest	`91e33261cd02`
v3 execution	30 preserved bench-result files, job 1206924	`2549abe326e3`
v3 analysis repair	Same outputs, repaired seed recognition	`82be7b0a08b9`
Priority retry	20 arms, native priority comparison	`4305d41eb79e`

Table S11 identifies the original retained sources. Public v0.2.0 adds the original synthetic pilot fixtures and saved batches, 126 erasure seed-unit counts, 10,773 dynamic-fidelity unit counters, 756 prefill units represented in overlapping views, and the later-engine scalar summaries and pairing asser-

tions. The public paths are `data/stage4c/`, `data/recovered/`, `analysis/`, and `scripts/reproduce_stage4c.py`; the claim matrix gives exact commands. A historical private commit is a provenance reference, not a public download.

The recorded release coverage remains 36/48. In particular, the trajectory ambiguity family was withheld from full credit because the older manuscript called its denominator overrides. The corrected wording here uses the same 232,011 unknown labels: 364,302 actionable steps give 63.6864%, while 356,577 overrides give 65.0662%. Public counts support both calculations. The released quartile code also establishes the floor-rank convention used for the original local-erasure IQR; linear interpolation is a different summary, not a correction to the observations.

Fidelity counters permit aggregation, not independent replay of cloned states. Pairing flags permit counting recorded assertions, not comparison of unavailable served-input vectors. The per-source prospective table is reconstructible from saved seed counts; the predictive table remains frozen scores. The original prefill structural-pressure summary lacks its per-seed probe records and receives no full claim-family credit. Raw trace sequences and private histories are excluded, including material with unresolved Mooncake trace-license scope and missing historical filtered BurstGPT bytes. These limits persist despite matching hashes of eligible outcome projections. No public release was modified during this manuscript revision.

S15 Priority intervention heterogeneity

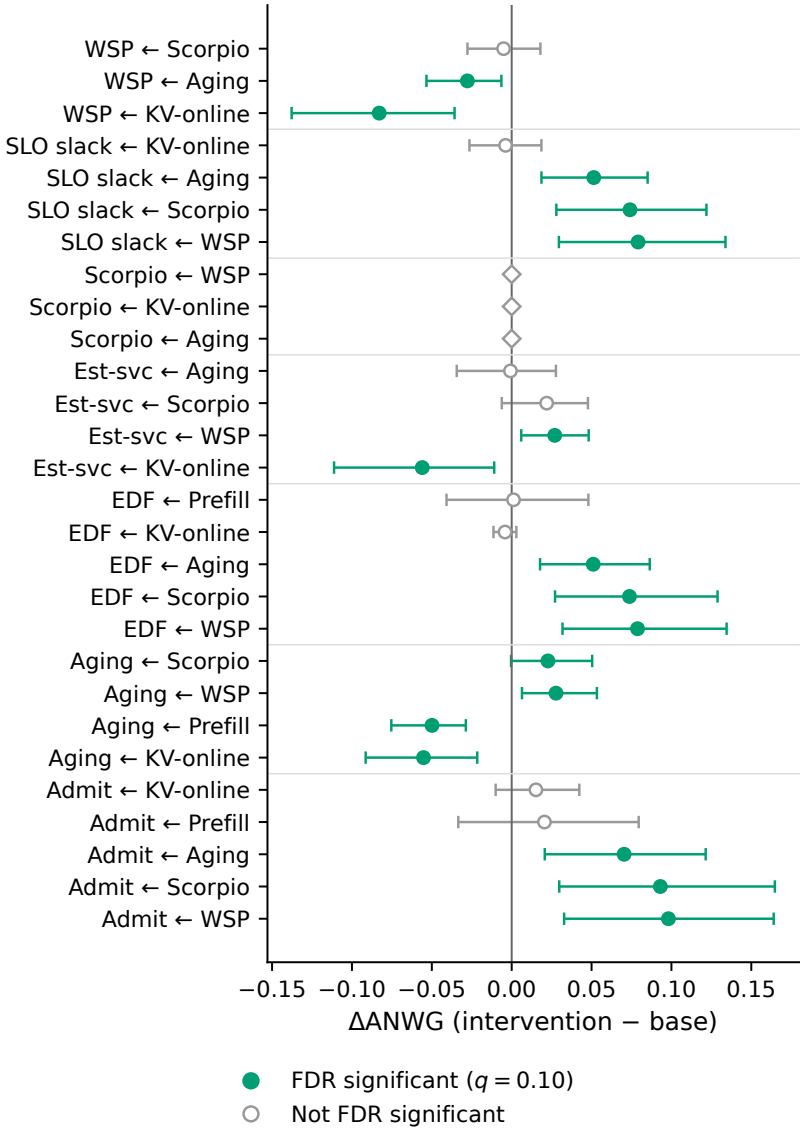


Fig. S3 All 28 priority single-module interventions under P1 (10 instances each). Points/intervals are instance-clustered mean ΔANWG with 95% bootstrap CI; filled markers denote Benjamini–Hochberg false discovery rate (BH-FDR) significance at q=0.10 (17/28 significant). Rows are grouped by base scheduler; labels show base←donor substitutions. Hollow diamonds mark contrasts that are zero on every instance

WSP: weighted shortest processing; Est-svc: estimated service time first; EDF: earliest deadline first; SLO slack: service-level-objective slack score; Admit: admission control; KV-online: key–value constrained online; Prefill: adaptive chunked prefill; Aging: aging priority. Scorpio denotes the reconstructed model, not validated native semantics.

S16 Exploratory KV reconstruction and compatibility

The corrected held-out campaign contains 13 affected single-transplant configurations evaluated on 410 workload instances. Six change only the declared KV slot; seven change admission, priority, or fairness on a Scorpio base whose KV pathway was also repaired. We reserve "clean slot" for the first six and do not treat it as a semantic compatibility certificate.

The repaired implementation maps both `kv_reserve` and `scorpio_decode_kv_filter` to the same reserve-placement component. Pinned-code contracts confirm reserve-threshold filtering, but this component does not implement the native Scorpio decode filter. Five of the six clean-slot pairs were interface-flagged; the remaining aging-base pair is merely unflagged. These results therefore describe an exploratory reconstructed reserve-guard intervention, not validated native-donor transplantation.

Post-hoc reanalysis averages the six pairs over the fixed 410-instance design: mean ΔANWG $= -0.01384$, with a post-hoc pointwise 95% interval $[-0.01516, -0.01252]$. Resampling keeps all pairs and workload conditions from one of the 10 held-out seed streams together ($B = 10{,}000$). In the regime arm, low load and moderate mixed load have observed zero effects; KV pressure and prefill-heavy workloads have means -0.01909 and -0.02362. In the joint arm the mean declines from zero to -0.05406 across zero to five elevated pressure axes (Figure S4). The historical mixed population and one-unflagged-pair sensitivity are retained in Sect. S9; their intervals are calculated separately. No new multiplicity-adjusted significance claim is made.

All 24 corrected pairwise definitions have positive mean interactions ($I \approx +0.014$) in the historical summary. Because repaired guard activation also affects these definitions, these are correction-structure interactions, not general evidence of module synergy. The original full-population results remain available for provenance.

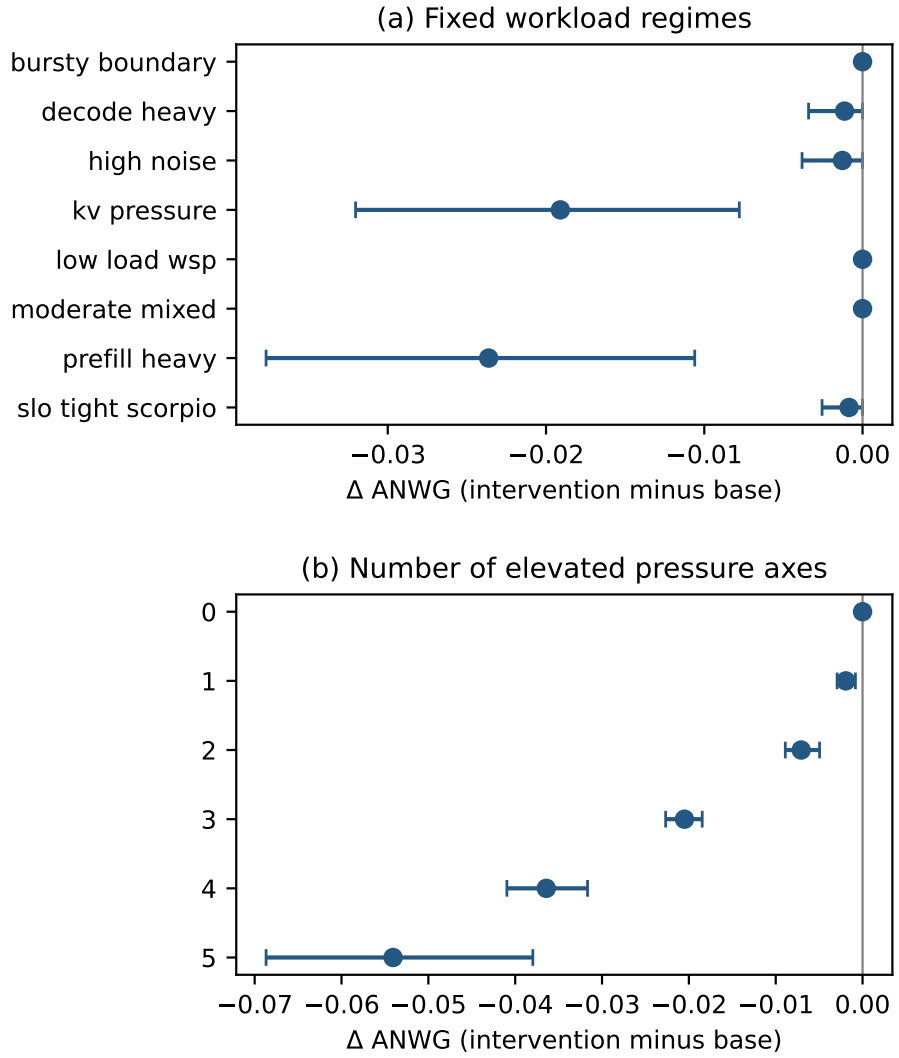


Fig. S4 Exploratory reconstructed reserve-guard effects for the same six clean-slot pairs in both panels. (a) Fixed workload regimes. (b) Joint design by number of elevated pressure axes. Points and pointwise 95% percentile intervals use this population only, jointly resampling 10 seed streams ($B = 10{,}000$), with fixed workload conditions and pairs. Clean slot does not imply validated native Scorpio semantics. Historical inactive rows and the seven contextual non-KV rows are excluded

S17 Extended metric-sensitivity results

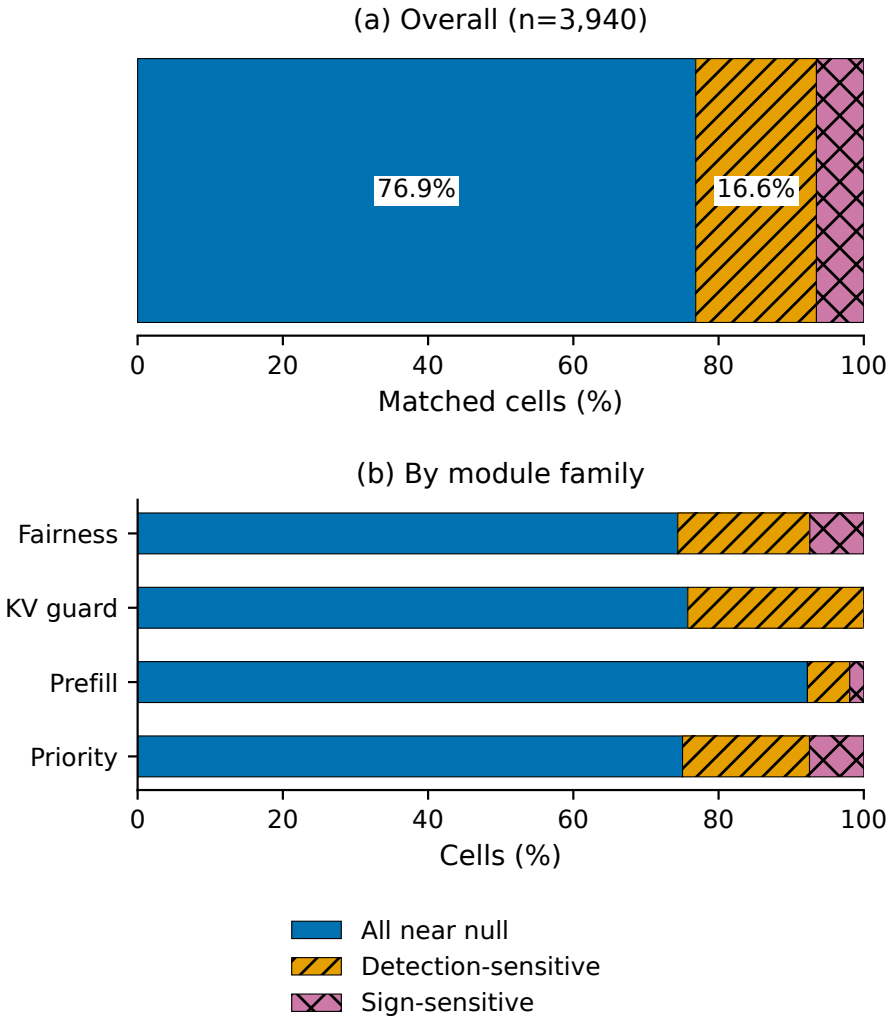


Fig. S5 Metric-sensitivity classification over 3,940 matched intervention cells, classified by numerical non-null status and benefit-oriented sign. (a) Overall composition: all near null (76.9%), detection-sensitive (16.6%), sign-sensitive (6.5%). (b) Same classification by module family; prefill is mostly all near null despite active request-level movement in §5.3, showing that an aggregate-null endpoint does not imply an inactive intervention

S18 Exploratory predictive baselines

Table S12 summarizes frozen baseline results on the primary grouped window holdout. Area under the receiver operating characteristic curve (AUROC) measures classification ranking; mean absolute error (MAE) measures prediction error. B0 predicts the training-set prevalence for classification and training-set mean for regression. The reported best values are descriptive minima or maxima over the prespecified candidates on the same test partition, not independently selected test performance. Opportunity AUROC is tied by module-only logistic regression, workload-plus-module logistic regression, and boosted trees; divergence AUROC is tied by workload-plus-module and richer diagnostic-feature logistic regression and boosted trees. Logistic regression models binary probabilities; boosted trees combine successive decision trees. The effect-size minimum comes from boosted trees; the interaction minimum comes from workload-only ridge regression (a penalized linear model). Model hyperparameters were fixed in the protocol. Feature sets intentionally differ, so these are predictive baselines, not the equal-information diagnostic comparison of Sect. S11. The limited test support prevents a general predictive-performance conclusion. AUROC 1.00 on opportunity/divergence tasks reflects only one independent test group and is not strong predictive evidence. The actionable-opportunity campaign is exported as window-level opportunity diagnostics (not counterfactual reward rows) in the benchmark. The proposed effect-with-divergence task has no overlapping workload identifiers between its input tables and was not successfully evaluated; no predictive gain is imputed for it.

Table S12 Prespecified predictive baselines (primary grouped window holdout). B0 is the constant baseline; MAE is mean absolute error

Task	Metric	B0	Best	Test groups	Status
Opportunity prediction	AUROC	0.50	1.00	1	exploratory
Decision divergence	AUROC	0.50	1.00	1	exploratory
Effect-size prediction	MAE	0.010	0.009	NR	exploratory
Pairwise interaction	MAE	0.016	0.015	NR	exploratory

NR: test-group count is not recorded in the compact regression summary; no count is inferred from the number of rows. The two AUROC tasks have one recorded independent test group each.

S19 Real-engine history and realized-input pairing

S19.1 Real-engine endpoint terminology

The attempted real-system probes use the following latency vocabulary. *Time to first token* (TTFT) is the interval from client request submission to receipt of the first output token in the retained benchmark, measured at the first returned token chunk. It includes queueing, prefill and communication time. *Time per output token* (TPOT) is the mean inter-token interval after the first token: completion latency minus TTFT, divided by output-token count minus one for requests with at least two output tokens. *End-to-end latency* is the full request completion interval from client request submission to receipt of the final token. *Throughput* is the rate of completed requests or generated tokens over a declared window. *Goodput* counts only requests (or tokens) that meet a declared service objective, such as an SLO or deadline. Related discussions are the preprints by Agrawal et al., *On Evaluating Performance of LLM Inference Serving Systems* (`https://arxiv.org/abs/2507.09019`), and Wang et al., *Revisiting Service Level Objectives and System Level Metrics in Large Language Model Serving* (`https://arxiv.org/abs/2410.14257`). No accepted or published status is asserted for these preprints. We use these terms only for real-system endpoints; simulator outcomes in this paper remain request-completion and ANWG unless explicitly labeled otherwise.

S19.2 Planned controls and initial attempts

Simulator interventions and public-trace replay describe mechanism relationships under a discrete-event serving model. A targeted real-vLLM protocol was intended to test selected diagnostics under real scheduling, model loading and token generation. Its initial attempts produced no valid treatment estimate; later completed runs require the separate pairing qualification below. Preliminary execution checks used vLLM 0.27.1, a pinned `Qwen2.5-7B-Instruct` snapshot, `DEFAULT_COMPILED` execution, CUDA GPU-programming-platform graph capture, and a frozen runtime manifest with out-of-band provenance controls. Logger-neutrality validation—paired control versus observational `LoggingScheduler` instrumentation—is recorded as passing in the retained summary of the tested control runs. Pressure calibration probes five replay-rate multipliers $(0.5, 1, 2, 4, 8)$ and records structural-opportunity rates of $0.0656, 0.1973, 0.2723, 0.4881, 0.6317$, which increase monotonically and satisfy the preregistered calibration criterion. The preregistered confirmatory LOW/MID/HIGH regimes are $0.5\times$, $1.0\times$, and $8.0\times$ replay rate, respectively. A targeted `PrefillGuardScheduler` treatment is implemented with deterministic no-op/divergence fixtures. In the confirmatory design, CONTROL denotes the native base vLLM scheduler under the frozen workload, not an external literature baseline.

S19.2.1 Confirmatory outcome

The retained summaries report passing preliminary execution, observer-neutrality and monotone pressure-calibration checks before the confirmatory executions. The complete original gate observations are unavailable for independent reconstruction; these historical reports are not new validation. Two preregistered confirmatory executions nonetheless failed objective infrastructure integrity before a valid LOW/MID/HIGH paired estimate could be computed. The original execution produced no valid scientific pairs. The permitted identical rerun reached terminal scheduler states but failed integrity because concurrent tasks collided on the shared benchmark service port, leaving only 5/15 paired task directories complete and zero valid HIGH-pressure paired seeds. The primary statistic is therefore undefined and the real-vLLM confirmatory result is infrastructure-inconclusive.

S19.3 Later completed campaigns and analysis repair

The v2 successor recovered 28/30 arms and failed its full integrity gate. v3 subsequently completed 15 paired tasks (30 arms, 96 requests per arm) on Wulver, job 1206924. The raw execution source is `2549abe326e3`, with frozen seed manifest at `1accfb7fb4cf`. The original analyzer incorrectly retained the v1 seed list and returned no valid v3 primary estimate. Repair protocol `b47adc6` and analyzer `82be7b0a08b9` recognize seeds 993001–993005 from the v3 manifest. They reuse the same bench-result files, not rerun trajectories.

For v3, x denotes load regime, s seed, and T and C treatment and control. The bar averages requests within an arm. Define

$$D_{x,s} = \left|\overline{\mathrm{TTFT}}_{T,x,s} - \overline{\mathrm{TTFT}}_{C,x,s}\right|, \tag{S1}$$

$$H_s = D_{\mathrm{high},s} - D_{\mathrm{low},s}. \tag{S2}$$

Here H_s is the contrast previously denoted C_s; H denotes its mean across seeds. This avoids reusing the main paper's cell-set symbol C. The repaired saved analysis reports mean $H = 13.5354$ ms, five positive seed contrasts, and a one-sided 95% bootstrap lower bound of 5.0335 ms. Fresh arithmetic on the preserved arm-level means reproduces the primary mean. The saved analysis classifies this contrast as positive under its frozen rule; the realized-input mismatch below prevents the intended causal interpretation.

A separate local native-priority campaign compares first-come, first-served (FCFS) and priority service on low/high arrival regimes. Its original execution had 18/20 valid instrumented arms; two low-regime arms lacked complete admission records. Retry source `4305d41eb79e` completes all 20 arms with 96 requests each, using five seeds 995001–995005, a cached Qwen 0.5B model and an RTX 5060 Ti. This differs from the v3 Qwen 7B/A100 setting and prefill treatment. Its primary H is the high-minus-low difference in high-priority TTFT benefit (control minus treatment), not v3's absolute all-request contrast. The saved mean is 732.8988 ms, with five positive contrasts and one-sided lower

bound 662.1473 ms. Low-priority requests incur a high-load TTFT cost; this is not universal improvement. All 83 entries of its preserved result checksum manifest match.

S19.4 Realized inputs contradict an identical-input pairing assumption

The preserved audit compared stored client `input_lens` vectors within paired arms and their multisets to distinguish reordering from different realized lengths. The preserved audit reports both comparisons failing in every v3 pair and every priority-retry pair (Table S13), with matching output-length vectors. It records hashes of existing JSON files. Public v0.2.0 exposes the scalar results and audit assertions, but not the exact served-input vectors; public reconstruction cannot independently repeat these vector comparisons. No model service or experiment was executed for this revision.

Table S13 Completed later vLLM runs versus realized-input equality. A pair is a regime–seed combination. Counts describe existing artifacts, not a hypothesis test

Campaign	Arms	Pairs	Equal input vectors	Equal input multisets
v3 prefill	30	15	0	0
Priority retry	20	10	0	0

The intended common trace path and command-line seed do not establish equal realized inputs. The inspected installed `TimedTrace` implementation uses `hash(key) & 0xFFFFFFFF` to seed prompt expansion. Python's process-dependent hash can differ between clients unless its process hash seed is controlled; a benchmark's random-seed flag alone does not establish that control. The retained validation checks trace hashes, counts and instrumentation, but does not check equal tokenized prompts across arms. This is a plausible mechanism for the observed mismatch. The historical process hash-seed environments and exact prompt expansion are not fully attested, so the audit does not establish it as the unique cause.

Different input-length multisets demonstrate failure of the strict identical-realized-input condition. They do not by themselves quantify the bias in either positive contrast, prove the contrast entirely spurious, or establish an alternative randomized-input estimand. The stored positive measurements and their bootstrap summaries remain part of the history; resampling cannot repair a mismatched pairing assumption. Successful execution, complete files, and repaired arithmetic therefore do not establish the intended paired causal interpretation. Strict real-engine paired confirmation of this protocol remains unestablished.

S20 Diagnostic overview

Main Fig. 1 provides the shared conceptual overview. Pressure alone need not create a nontrivial candidate set; structural opportunity need not produce a top-choice disagreement. These additional qualifications are tested in the controlled and prospective panels, not inferred from the diagram's arrows.

S21 Additional held-out and metric detail

Table S14 gives the per-module metric classifications. The exploratory held-out KV summary reports observed zeros at two lower pressure levels and $\Delta\text{ANWG} = -0.0054$ $[-0.0097, -0.0021]$ at high pressure. These historical reconstructed-guard results inherit the native fidelity and execution-version limitations in Sect. S16; they are not independent proof of native Scorpio validity.

Table S14 Metric categories from the recovered matched observations. Numerical tolerance is 10^{-12}; these are not significance classifications

Module	All near null	Detection-sensitive	Sign-sensitive	Total
Priority	2408	561	240	3209
Prefill	380	24	8	412
KV guard	150	48	0	198
Fairness	90	22	9	121
All	3028	655	257	3940

S22 Execution lineage and interpretation limits

Pinned source excerpts and complete file hashes are preserved in `artifacts/computing_stage2a/pinned_contract_sources.json`. The local-erasure records name source commit `b5bc1967`; the controlled priority deployment audit names `23d2c8d8`. Controlled-campaign snapshots and the local source share admission–prefill–priority–fairness–placement ordering. A manifest generation commit is not an execution attestation. Older KV controls are reused across the correction, and exact executable/state identity is incomplete. The correction routes both reserve and Scorpio-labeled strings to the same reserve component; the earlier Scorpio string was inactive. In the pinned source, the tiebreak branch requires a retained priority other than `aging_priority`; the aging donor already embeds aging in its priority key. The dynamic comparison makes the resulting context dependence explicit (Sect. S13). This corrects the

earlier inverse wording of the branch condition without changing experimental outputs. These implementation facts constrain semantic claims even when specification-level intervention-isolation checks pass.

The current register retains source hashes and distinguishes recorded, inspected and unverified execution identifiers. The later dynamic-fidelity inventory at `ed2321b3` marks all five reserve-donor candidates context-dependent and not executed by design. Successful tests on other candidates cannot validate these historical interventions. The five native-Scorpio interpretations remain invalid, and the five reserve-donor interventions remain inconclusive.

S23 Related-manuscript overlap detail

The related whole-policy/headroom manuscript and this module-attribution paper share infrastructure and reference-conditional counterfactual reasoning. The Azure source files are byte-identical across the audited projects, and record intersections occur with both its native and fresh confirmatory windows. Full source-specific intersection counts and hashes are in `docs/computing/stage2a_overlap_audit.md`; they must not be summed across campaigns as unique records.

For BurstGPT, the related original file has 1,429,737 rows; its positive-token filter retains 1,404,294. Mapping filtered ordinals to original row identifiers matches all 2,304 unique selected module-project token payloads and all 5,616 available relative-time comparisons. Public replay has 216 selected records, of which 144 intersect the related native windows; the prospective confirmatory subset has 1,440, with 17 intersections. The 576 local-erasure records and 72 prospective pilot records have none in that inspected native set. None of these inspected BurstGPT sets intersects the related fresh confirmatory windows. This is logical selected-record mapping, not full-file byte identity: the exact filtered release remains missing. Different filenames and replay annotations do not establish independent source datasets or independent research outcomes.

The related manuscript is posted as an SSRN preprint (8 October 2026), `https://doi.org/10.2139/ssrn.7584825`. Public reproducibility documentation is at `https://github.com/SoroushVahidi/llm-serving-heuristic-evolution`. Its v1.1.0 artifact archive and later repository additions are distinct versions and do not independently validate this paper's module interventions. The scientific relationship and shared methods are summarized in main Sect. 2.

S24 Frozen evidence catalog

The historical campaign catalog in Table S15 locates supporting artifacts. Inclusion in the catalog is not a new validation of every secondary analysis. In particular, seed replication does not create additional independent upstream

windows, and corrected KV evaluations remain exploratory. The main paper organizes scientific claims by estimand and diagnostic question.

Table S15 Evidence strata used in the manuscript. PRC = preregistered confirmatory, HO = held-out, EXP = exploratory/post-hoc diagnostic, HIST = historical or corrected provenance

Campaign (stratum)	Scope and recorded purpose
Priority P0/P1/P2 (PRC)	30 instances; 28 priority singles; P2 priority×fairness four-arm condition. Tests pressure vs. priority opportunity and the priority×fairness interaction.
Prefill F0/F1/F2 (PRC)	30 instances; 4 prefill singles; valid prefill×fairness four-arm condition. Tests inactive vs. sustained prefill opportunity, request consequences, and sub-additive degradation.
Corrected KV guard (HO/HIST/EXP)	410 instances; 10 seed streams; 25,010 corrected evaluations. Separates previously inactive implementation from exploratory reconstructed reserve-placement effects.
Held-out generalization (HO)	45 fresh workload instances and fresh high range seeds. Tests transfer of priority opportunity, KV pressure dependence, prefill stage separation, and priority×fairness interaction.
Public-trace replay (PRC replay)	63 dependent window–rate instances from BurstGPT, Azure 2023/2024, and Mooncake FAST'25. Tests public-trace arrivals and request shapes; outcomes remain simulated.
Public-trace seed stability (Secondary replay)	400 seed replicates per window–rate instance imply a design total of 25,200 units. Historical completion was reported; a complete raw archive and execution manifest were not recovered. The study targeted replay sensitivity to seed variation.
Actionable opportunity (PRC follow-up)	140 new trace windows, disjoint from the 21 public-replay source windows. Tests whether top-choice disagreement refines structural opportunity for the Mooncake anomaly.
Earlier corrected fairness study (PRC/EXP)	Fairness regimes and initial priority×fairness estimate. Supports fairness context.

S25 Worked diagnostic example from saved pilot observations

The following cases are original saved synthetic observations from Sect. S11, not hypothetical outcomes or new simulations. In the singleton case no competition occurs. In "both fit," FIFO and EDF recommend different orders but admit the same two-request set, so the local disagreement is erased and neither request outcome changes. In "reordered, loose deadlines," the admitted set changes, both request completion times change, and ANWG remains zero-difference. The deadline-crossing case changes the same number of request outcomes but improves ANWG by 1/2.

The aggregate null thus covers absent competition, local erasure and changed request timing without deadline crossing. Recommendations and local admitted sets distinguish the first two boundaries; matched full-trajectory outcomes resolve the third. Neither local set equality nor a zero aggregate contrast alone determines future timing. Both equally instrumented analyzers recover these distinctions on the saved cases (Table S7); no comparative superiority or population-level accuracy is inferred.

S26 Metric-panel accounting and numerical classification

The recovered metric panel contains 3,940 matched workload–intervention cells: controlled priority 850, controlled prefill 130, held-out 755 and public replay 2,205. Actionable-opportunity diagnostics supply no outcome cells to this panel. Each cell has ANWG and the four secondary metrics defined in Sect. 3. Effects are oriented toward benefit before classification: treatment minus base for goodput and soft credit, and base minus treatment for the two tardiness measures.

At numerical tolerance 10^{-12}, 3,028 cells (76.9%) are near null on all five endpoints. Another 655 (16.6%) are detection-sensitive, meaning some endpoints are outside the tolerance while others are not; 257 (6.5%) have both positive and negative benefit-oriented effects and are sign-sensitive. Sign sensitivity takes precedence over the detection category. These are numerical classifications, not significance tests or practical equivalence. Repeated cells on one workload are not independent replications.

Prefill is 92.2% all near null across this panel, consistent with the controlled example of request movement invisible to aggregate endpoints. The classification alone does not assert request movement in every null cell. The endpoint panel is finite and cannot establish invariance to all serving metrics. Online Resource 1, Fig. S5 and Table S14 retain the family breakdowns. All cells are reconstructed from 168 tracked raw outputs, with exact agreement on identifiers, hashes, categories and 59,100 base/treatment/effect values against the historical export. Numerical recovery does not repair missing execution attestations or native fidelity for exploratory KV rows.

A separate post-hoc completion-coverage check reclassifies all 3,940 public rows with the same tolerance and then inspects the 257 sign-sensitive rows.

Every one records weighted completion goodput exactly 1.0 in both arms; none has a change above the numerical tolerance. The split is 181 controlled priority, 11 controlled prefill and 65 held-out cells. The corresponding 36 raw source files independently reproduce these completion-goodput values. Thus the sign reversals do not coincide with an observed loss of weighted completion coverage. The native/single aggregate rows do not serialize exact completion-ID sets, so a request-membership or counterfactual decomposition cannot be recovered from them. Arrival normalization, completion conditioning, priority weighting and binary versus soft deadline penalties remain separate features of the endpoint definitions. This descriptive check adds no new significance test or claim of metric equivalence.

S27 Finite bootstrap resolution

Some historical bootstrap sign-tail probabilities have zero observed tail hits. This is a finite Monte Carlo result, not $p = 0$: the two-sided resolution at $B = 2000$ is 0.001. A post-hoc plus-one sensitivity $\min\{1, 2(h+1)/(B+1)\}$, where h is the tail-hit count, retains the reported priority 17/28 and prefill 0/4 classifications. This check does not turn a bootstrap tail diagnostic into an exact null-randomization test. The practical ANWG threshold 0.002 is distinct from statistical detectability.

References

1. Agrawal A, Kedia N, Mohan J, et al (2024) Vidur: A large-scale simulation framework for LLM inference. In: Proceedings of Machine Learning and Systems, pp 351–366, URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/b74a8de47d2b3c928360e0a011f48351-Abstract-Conference.html
2. Alomar A, Hamadanian P, Nasr-Esfahany A, et al (2023) CausalSim: A causal framework for unbiased trace-driven simulation. In: 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23). USENIX Association, pp 1115–1147, URL https://www.usenix.org/conference/nsdi23/presentation/alomar
3. Cho J, Choi H, Heo G, et al (2026) LLMServingSim 2.0: A unified simulator for heterogeneous and disaggregated LLM serving infrastructure. In: 2026 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, pp 1–14, https://doi.org/10.1109/ISPASS69572.2026.00012
4. Curtsinger C, Berger ED (2015) Coz: Finding code that counts with causal profiling. In: Proceedings of the 25th Symposium on Operating Systems Principles. ACM, SOSP '15, pp 184–197, https://doi.org/10.1145/2815400.2815409
5. Fawcett C, Hoos HH (2016) Analysing differences between algorithm configurations through ablation. Journal of Heuristics 22(4):431–458. https://doi.org/10.1007/s10732-014-9275-9
6. Fu Y, Zhu S, Su R, et al (2024) Efficient LLM scheduling by learning to rank. In: Advances in Neural Information Processing Systems, vol 37. Curran Associates, Inc., pp 59006–59029, https://doi.org/10.52202/079017-1882
7. Law AM (2015) Simulation Modeling and Analysis, 5th edn. McGraw-Hill Education
8. Li J, Li J (2026) CrossServe: Cross-layer scheduling for SLO optimization in Multi-Tenant LLM serving. In: Proceedings of the 55th International Conference on Parallel Processing. ACM, pp 187–196, https://doi.org/10.1145/3832810.3832859
9. Liao J, Dong Q, Liang Y, et al (2026) Laser: Unlocking layer-level scheduling for efficient Multi-SLO LLM serving. In: Proceedings of the 31st ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming. ACM, pp 509–521, https://doi.org/10.1145/3774934.3786413
10. Patel P, Choukse E, Zhang C, et al (2024) Splitwise: Efficient generative LLM inference using phase splitting. In: 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA). IEEE, pp 118–132, https://doi.org/10.1109/ISCA59077.2024.00019
11. Qin R, Li Z, He W, et al (2025) Mooncake: Trading more storage for less computation — a KVCache-centric architecture for serving LLM chatbot. In: 23rd USENIX Conference on File and Storage Technologies (FAST 25).

USENIX Association, Santa Clara, CA, pp 155–170, URL https://www.usenix.org/conference/fast25/presentation/qin

12. Sheng Y, Cao S, Li D, et al (2024) Fairness in serving large language models. In: 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). USENIX Association, Santa Clara, CA, pp 965–988, URL https://www.usenix.org/conference/osdi24/presentation/sheng
13. Sobol' IM (2001) Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Mathematics and Computers in Simulation 55(1–3):271–280. https://doi.org/10.1016/S0378-4754(00)00270-6
14. Srivatsa V, He Z, Abhyankar R, et al (2025) Preble: Efficient distributed prompt scheduling for LLM serving. In: International Conference on Learning Representations, pp 37057–37082, URL https://openreview.net/forum?id=meKEKDhdnx
15. Sun B, Huang Z, Zhao H, et al (2024) Llumnix: Dynamic scheduling for large language model serving. In: 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). USENIX Association, pp 173–191, URL https://www.usenix.org/conference/osdi24/presentation/sun-biao
16. Zhong Y, Liu S, Chen J, et al (2024) DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In: 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24). USENIX Association, pp 193–210, URL https://www.usenix.org/conference/osdi24/presentation/zhong-yinmin